



Johnny Johansen og Jorunn Lajord

FD-trygd

Dokumentasjonsrapport
Utdanning. 1992-1997

Notater

Forord

Denne rapporten inneholder en dokumentasjon av en del av dataene i FD-Trygd (Forløpsdatabasen-Trygd). Dokumentasjonen omfatter høyeste fullførte utdanning og igangværende/avsluttet utdanning, og omfatter opplysninger for årene 1992 til 1997. For å klargjøre data for 1997 må vi legge inn data for 1998 i tillegg i basen. Dermed er kontrollene for 1998 tatt med i kapitlene 3 og 6. Dette notatet erstatter Notater 98/50 for årene 1992 og 1993.

I kapittel 1 er det gitt en generell orientering om FD-Trygd. I dette kapitlet er det også gjort rede for de typer arbeidsprosesser og aktiviteter som prosjektgruppa utfører på materialet fra arbeidsstart med rådatafiler og til avslutning av arbeidet med ilegg av data i databasen. Til disse arbeidsprosessene hører blant annet spesifisering og kontroll av data, datamodellering og programmering.

Kapitlene 2-7 beskriver grunnlagsmaterialet for utdanning, og hvordan dette tilrettelegges for forløpsstrukturen i databasen. Opplysninger om høyeste fullførte utdanning kan nyttes som bakgrunnsvariabel. Disse dataene er beskrevet i kapitlene 2-4. For hver person har vi i tillegg opplysninger om perioder (start-/stopdatoer) for utdanning som gjennomføres/er gjennomført. Dette for å se disse i sammenheng med opplysninger om andre aktiviteter, som f.eks. "trygde-aktiviteter" og deltakelse i inntektsgivende arbeid. Disse dataene er beskrevet i kapitlene 5-7.

I arbeidet med denne datatilretteleggingen er det mange personer som har bistått prosjektgruppa med faglig råd og veiledning. Vi nyter her anledningen til spesielt å takke medlemmene i referansegruppa for FD-Trygd. Disse er Anita M. Sivertsen, Sosial- og helsedepartementet, Thomas Øigarden, Arbeidsdirektoratet, Carl Gjersem, Finansdepartementet, Oddbjørn Haga og Ola Thune, Rikstrykdeverket, Dag Kiberg, Norsk Samfunnsvitenskapelig Datatjeneste, Kjell Vaage, Universitetet i Bergen, samt Anders Akselsen, Eva Birkeland, Grete Dahl, Bjørn Roar Joneid, Tom Kornstad, Berit Otnes, Øyvind Sivertstøl, Anne Kristine Skappel og Inger Texmon, alle Statistisk sentralbyrå.

En spesiell takk går også til kolleger ved seksjon for befolknings- og utdanningsstatistikk, Statistisk sentralbyrå, for verdifull bistand i forbindelse med spesifikasjonen av data.

Kapitlene 1, 3 og 6 er skrevet av Jorunn Lajord, som også har hatt redigeringsansvar for rapporten. Kapittel 2, 4, 5 og 7 er utarbeidet av Johnny Johansen og Jorunn Lajord i fellesskap. Torger Eidem har vært med på å utarbeide forrige notat, Notater 98/50.

Hele dokumentasjonsrapporten er lagt på en felles disk, (q:\dok\fdtrygd\.....), og er således gjort tilgjengelig på elektronisk form for alle ansatte i SSB. I tillegg er informasjon om prosjektet lagt ut på Statistisk sentralbyrås web-sider, og er således gjort tilgjengelig for alle. Adressen er: <http://www.ssb.no/emner/03/fd-trygd/>.

Innhold

	Side
1 Innledning	3
2 Spesifikasjonsnotat – Høyeste fullførte utdanning	4
2.1 Beskrivelse av grunnlagsregistre	4
2.2 Korreksjoner og kontroller	5
2.3 Miniregistre	6
2.4 Avvik fra offisiell statistikk.....	7
3 Kontroller - Høyeste fullførte utdanning	8
3.1 Kontroller på filer for befolkningens høyeste utdanning, pr. 1/10 hvert år	8
4 Diagrammer – Høyeste fullførte utdanning.....	10
4.1 Dataflytdiagram nivå 1 - miniregister (SAS)	11
4.2 Fysisk dataflytdiagram - miniregister (SAS).....	12
4.3 Programbeskrivelse av miniregister-rutiner (SAS)	13
4.4 Dataflytdiagram nivå 1 - database (Oracle).....	14
4.5 Fysisk dataflytdiagram - database (Oracle).....	15
4.6 Databaserutiner for høyeste fullførte utdanning (Oracle)	17
4.7 Programbeskrivelse av db-rutiner for høyeste fullførte utdanning (Oracle)	18
4.8 Flytdiagram for PL/SQL-kode til høyeste fullførte utdanning.....	22
4.9 Datamodell for høyeste fullførte utdanning	23
4.10 Modellspesifikasjoner for høyeste fullførte utdanning	24
5 Spesifikasjonsnotat – Igangværende utdanning.....	31
5.1 Beskrivelse av grunnlagsregistrene	31
5.2 Korreksjoner og kontroller	35
5.3 Miniregistre	39
5.4 Sammenkobling ved ilegg i basen	41
5.5 Forløpstabell-Kurs.....	42
5.6 Forløpstabell-Viktigste kurs	49
5.7 Avvik fra offisiell statistikk.....	61
6 Kontroller.....	62
6.1 Kontroller på filene for igangværende utdanning, pr. 1/10 hvert år	62
6.2 Kontroller på filene for avslutta utdanning i “skole”-året.....	64
7 Diagrammer – Igangværende utdanning.....	66
7.1 Dataflytdiagram nivå 1 - miniregister (SAS)	67
7.2 Fysisk dataflytdiagram - miniregister (SAS).....	69
7.3 Programbeskrivelse av miniregister-rutiner (SAS)	71
7.4 Dataflytdiagram nivå 1 - database (Oracle).....	73
7.5 Fysisk dataflytdiagram - database (Oracle).....	74
7.6 Databaserutiner for igangværende utdanning (Oracle)	77
7.7 Programbeskrivelse av databaserutiner på kursnivå (Oracle)	79
7.8 Programbeskrivelse av databaserutiner på viktigste-kursnivå (Oracle)	86
7.9 Flytdiagram for PL/SQL-kode til igangværende utdanning	93
7.10 Datamodell for igangværende utdanning	97
7.11 Modellspesifikasjoner for igangværende utdanning	98
De sist utgitte publikasjonene i serien Notater	119

1 Innledning

FD-Trygd er en forløpsdatabase som er under oppbygging i Statistisk sentralbyrå. De viktigste datatypene som inngår i FD-Trygd er trygdedata, demografidata, utdanningsdata, sosialhjelpsdata, sysselsettingsdata, arbeidssøkerdata og inntekts- og formuesdata. Dataene er hentet fra administrative registre i Statistisk sentralbyrå, Rikstrygdeverket og Arbeidsdirektoratet.

FD-Trygd skal inneholde opplysninger for hele befolkningen for perioden 1992-2000. Person er statistisk analyseenhet for alle opplysningene i databasen. Til hver personrecord i databasen er det opplysninger om familienummer, slik at det er mulig å knytte sammen informasjon for alle personer som hører til samme familie. Familie er definert i overensstemmelse med de registreringer og definisjoner som finnes ved personregistreringen. Dataene i FD-Trygd vil være velegnet til analyse av mange aktuelle velferdspolitiske problemstillinger. Dataene vil blant annet gi grunnlag for studier av trygdeforløp og av overgangen mellom forskjellige sosiale ordninger og arbeidsmarkedet.

Det er opprettet en styringsgruppe og en referansegruppe for FD-Trygd. Både i styringsgruppa og referansegruppa er det representanter fra Sosial- og helsedepartementet, Finansdepartementet, Rikstrygdeverket, Norsk Samfunnsvitenskapelig Datatjeneste, Arbeidsdirektoratet og SSB. I referansegruppa er også en representant fra Institutt for økonomi, UiB. Det daglige arbeidet med prosjektet utføres av prosjektgruppa ved Avdeling for personstatistikk, SSB.

Prosjektgruppas arbeid kan inndeles i flere hovedaktiviteter. Gruppa spesifiserer og henter ut de dataene som skal inngå i databasen. For de ulike typer data eller statistikkområder finnes disse opplysningene i forskjellige administrative registre. De administrative registrene inneholder som oftest opplysninger pr. tidspunkt. I enkelte registre er også dataene satt sammen slik at de gir opplysninger fortløpende over tidsperioder, for eksempel over et år. Det kreeres egne filer (miniregiste) for de dataene som hentes ut fra de administrative registrene. Miniregistrene er input til databasen, og det blir foretatt en del kontroller og annen type kvalitetssikringsarbeid på opplysningene i miniregistrene. (Det vil også bli foretatt konsistenskontroller av dataene på tvers av statistikkområdene etter at dataene er lagt inn i databasen). Etter at dataene i miniregistrene for hvert enkelt statistikkområde er spesifisert og kontrollert består prosjektgruppas arbeid i å modellere dataene. Dette betyr å foreta en omorganisering av dataene, f. eks. fra data pr. tidspunkt, til forløpsdata. Omorganiseringen til forløpsdata krever en nøye gjennomgang av mulig og relevant datering av opplysningene i miniregistrene og i de opprinnelige administrative registrene, slik at «endringer i tilstander» blir ivaretatt og registrert best mulig i databasen. I FD-Trygd er slike tilstandsendringer registrert ved en datovariabel. For en del tilstandsendringer er dateringen angitt ved år, måned, dag. For andre opplysninger er dateringene år, måned eller år, kvartal. For noen få data finnes bare årsopplysninger. Dette gjelder for eksempel for inntekts- og formuesdataene. Etter at datamodelleringen er avsluttet består prosjektgruppas arbeid i å programmere og utvikle rutiner for ilegg av data i databasen.

I prosjektgruppas arbeid blir dokumentasjon tillagt stor vekt. Dette gjelder både dokumentasjon av dataene i FD-Trygd så vel som dokumentasjon av alle aktiviteter som er forbundet med å bygge opp og videreutvikle denne forløpsdatabasen. I dette notatet er dokumentert det arbeid som er utført for å spesifisere opplysningene om befolkningens høyeste utdanning og utdanning som befolkningen er igang med eller har avsluttet i perioden.

2 Spesifikasjonsnotat – Høyeste fullførte utdanning

Vi har to hovedtyper av filer med utdanningsopplysninger som vi bruker i FD-Trygd. Det er registre over Høyeste fullførte utdanning på den ene siden, og registre over Igangværende utdanning og registre over Avslutta utdanning som den andre hovedtypen. Registeret over høyeste fullførte utdanning brukes til å finne hva slags utdanning befolkningen har. Dette registeret behandles for seg i kapitlene 2-4. For å finne hva slags utdanning befolkningen er i gang med (forløpsdata), bruker vi igangværende- og avsluttafiler. Disse registrene behandles i kapitlene 5-7.

Prosjektet FD-Trygd inneholder opplysninger fra og med 1/1-1992. Utdanningsopplysningene starter med statistikkåret 1991 og siste årgangen er 1998.

2.1 Beskrivelse av grunnlagsregistre

2.1.1 Filidenter

Ved oppstart av prosjektet lå utdanningsregistrene på stormaskin, og måtte derfor overføres til UNIX. Vi tar med filidentene både på stormaskin og på UNIX. Filene har samme innhold og omfang på begge plattformene.

På stormaskinen hadde filene disse identene:

PL213.S4368.I654A1A1.G9100.V00

PL213.S4368.I654A1A1.G9200.V00

PL213.S4368.I654A1A1.G9300.V00

I 1999 er utdanningsfilene lagt over til unix, og filidentene er:

\$UTD/utdnivaa/arkiv/nus89/g1994.dat

\$UTD/utdnivaa/arkiv/nus89/g1995.dat

\$UTD/utdnivaa/arkiv/nus89/g1996.dat

\$UTD/utdnivaa/arkiv/nus89/g1997.dat

\$UTD/utdnivaa/arkiv/nus89/g1998.dat

De filene vi danner oss som grunnlagsregistre har fått følgende filidenter:

\$TRYGDFOB/data/utdann/g91/grunnlag/hoeyfull/g91.ssd04

\$TRYGDFOB/data/utdann/g92/grunnlag/hoeyfull/g92.ssd04

\$TRYGDFOB/data/utdann/g93/grunnlag/hoeyfull/g93.ssd04

\$TRYGDFOB/data/utdann/g94/grunnlag/hoeyfull/g94.ssd04

\$TRYGDFOB/data/utdann/g95/grunnlag/hoeyfull/g95.ssd04

\$TRYGDFOB/data/utdann/g96/grunnlag/hoeyfull/g96.ssd04

\$TRYGDFOB/data/utdann/g97/grunnlag/hoeyfull/g97.ssd04

\$TRYGDFOB/data/utdann/g98/grunnlag/hoeyfull/g98.ssd04

2.1.2 Filbeskrivelser

Denne filen omfatter alle bosatte personer 16 år og over, samt 15-åringer som har avsluttet grunnskolen. I tillegg er noen personer med ukjent bosted registrert. 15-åringer og de med ukjent bosted er ikke med i offisiell statistikk, og tas ikke med i basen.

Vi starter med fila pr. 1/10-1991, fordi vi ønsker registrert høyeste fullførte utdanning ved starttidspunkt for FD-Trygd, 1/1-1992.

Nedenfor har vi listet opp hvilke variable vi tar ut fra de ulike grunnlagsfilene, og forklaring på noen av variablene.

Posisjon	Tekst
1-11	Fødselsnummer. Også ugyldige fødselsnummer skal tas med i basen
43-48	Utdanningens art (omkodet). Følger standard for utdanningsgruppering
49-50	Klassetrinn (omkodet). Viser hvilket klassetrinn utdanningen vanligvis avsluttes på.
59-62	Fullføringstidspunkt (YYMM, må kodes om til YYYYMM)

Opplysningene om høyeste fullførte utdanning blir oppdatert hvert år med opplysninger fra avsluttafila. Fra og med filen pr. 1/10 1993 blir filen også oppdatert med opplysninger om utdanning fullført i utlandet (opplysninger fra Lånekassen). Opplysningene om utenlands-utdanning dekker utdanning fullført i utlandet (av Lånekassens kunder) fra og med skoleåret 86/87. For personer som innvandret til Norge i 1991 eller senere, samt for 30-40 prosent av de som innvandret mellom 1980 og 1990, har man ikke opplysning om utdanning tatt i utlandet før de innvandret til Norge.

Denne filen er en nettofil i det personer er enhet. Personer som har fullført mer enn en utdanning, er bare registrert med den høyeste av utdanningene. Det vil si at den utdanningen som har det høyeste nivået, er registrert. Hvis en person har fullført flere utdanninger på samme nivå, er det den utdanningen med høyeste fullføringstidspunkt som er registrert.

Vi bruker de omkodede variablene for utdanning, fordi det er disse som «gjelder» i dag. De uomkodede variablene beskriver utdanning før overgangen fra 7-årig folkeskole til 9-årig grunnskolen. En del (yrkes)utdanning som før var kodet til 8. og 9. klassetrinn ble kodet om til utdanning på videregående nivå (10. klassetrinn).

2.2 Korreksjoner og kontroller

2.2.1 Fødselsnummerkontroll

Vi kjører de samme fødselsnummerkontrollene som kjøres for resten av FD-Trygd (se Q:\dok\fdtrygd\info\plandok>manual\kbs19961210n1.doc. Merking av fødselsnumre ligger i variabelen mrk_fnr, og resultatene finnes i kapittel 3.

2.2.2 Dublettkontroll

På disse filene skal det ikke være dubletter i fødselsnummer. Dublettkontrollen utføres dermed med fødselsnummer som eneste variabel. Resultat av dublettkontrollen finnes i kapittel 3.

2.2.3 Datokontroll

Datoene som ligger på utdanningsregistre er månedsopplysninger, dermed får vi ingen «ugyldige» datoer, men vi sjekker gyldige kombinasjoner av år og måned.

Kontrollen som utføres på variabelen **fullføringstidspunkt (FULLF)**:

Hvis manglende fullføringstidspunkt (NULL) Så

Legger ikke inn noen dato, mrk_dl settes lik 1

Hvis månedsopplysningen er '00' Og år er gyldig Så

Opplysning settes til '01' og mrk_dl settes lik 2

Vi sjekker i tillegg om noen har fullføringstidspunkt etter at fila er skrevet ut. Hvis dette er tilfelle, slettes disse recordene.

2.2.4 Omarbeidelser

I basen legger vi inn en ny record for hver gang det har skjedd en endring i høyeste fullførte utdanning (oppdatert av fagseksjonen), og vi tar vare på gamle records for i ettertid å kunne rekonstruere befolkningens høyeste utdanning på de tidspunktene vi måtte ønske. Vi legger inn en dato som viser hvilket årsregister de nye opplysninger kommer fra. Denne årgangsopplysningen oppdateres også for løpende tilfeller uten noen endringer i internvariablene.

I grunnlagsregisteret er det registrert noen 15-åringer og noen med uoppgitt bosted. Disse blir ikke tatt med i den offisielle statistikken. Vi fjerner derfor disse to gruppene fra miniregisteret, for å få likhet med den offisielle statistikken.

For å lette uttaket av data fra databasen, kodes det ut fra **utdanningens art** (NUS) fire nye variable som viser de ulike trinnene i utdanningskoden:

NUS1 er første siffer i utdanningskoden

NUS2 er de to første sifrene i utdanningskoden

NUS3 er de tre første sifrene i utdanningskoden

NUS4 er de fire første sifrene i utdanningskoden

Denne omkodingen blir gjort ved ilegg i basen.

2.3 Miniregistre

Miniregisteret for høyeste fullførte utdanning er en fil med de variablene vi har tatt ut fra grunnlagsfila, samt variable som er dannet i løpet av prosessen. Miniregistrene for høyeste fullførte utdanning ligger på:

```
$TRYGDFOB/data/utdann/g91/minireg/hoeyfull/g91.ssd04  
$TRYGDFOB/data/utdann/g92/minireg/hoeyfull/g92.ssd04  
$TRYGDFOB/data/utdann/g93/minireg/hoeyfull/g93.ssd04  
$TRYGDFOB/data/utdann/g94/minireg/hoeyfull/g94.ssd04  
$TRYGDFOB/data/utdann/g95/minireg/hoeyfull/g95.ssd04  
$TRYGDFOB/data/utdann/g96/minireg/hoeyfull/g96.ssd04  
$TRYGDFOB/data/utdann/g97/minireg/hoeyfull/g97.ssd04  
$TRYGDFOB/data/utdann/g98/minireg/hoeyfull/g98.ssd04
```

Det blir i tillegg dannet tekstfil av miniregistrene som skal brukes til ilegg i basen. Disse har filident:

```
$TRYGDFOB/data/utdann/g91/minireg/hoeyfull/g91.txt  
$TRYGDFOB/data/utdann/g92/minireg/hoeyfull/g92.txt  
$TRYGDFOB/data/utdann/g93/minireg/hoeyfull/g93.txt  
$TRYGDFOB/data/utdann/g94/minireg/hoeyfull/g94.txt  
$TRYGDFOB/data/utdann/g95/minireg/hoeyfull/g95.txt  
$TRYGDFOB/data/utdann/g96/minireg/hoeyfull/g96.txt  
$TRYGDFOB/data/utdann/g97/minireg/hoeyfull/g97.txt  
$TRYGDFOB/data/utdann/g98/minireg/hoeyfull/g98.txt
```

Filbeskrivelse for miniregisteret er:

Antall tegn	Type	Kortnavn	Variabel
11	char	fnr	Fødselsnummer
6	char	nus	Utdanningens art
2	num	kltrinn	Klassetrinn
6	num	fullf	Fullføringstidspunkt (YYYYMM)
1	char	mrk_fnr	Fødselsnummer-merking
1	char	mrk_dl	Datomerking
4	num	aargang	Årgang (kodes ut fra filnavnet)

2.4 Avvik fra offisiell statistikk

Opplysningene for høyeste fullførte utdanning stemmer med den statistikken som er publisert. Dette fordi vi ikke gjør noe med recordene fra disse filene.

3 Kontroller - Høyeste fullførte utdanning

I dette kapittelet har vi sett på noen resultater fra kjøring på filer for utdanning, og sammenlignet med offisiell statistikk.

3.1 Kontroller på filer for befolkningens høyeste utdanning, pr. 1/10 hvert år

	1991	1992	1993	1994
Antall records på grunnlagsregisteret	3 416 459	3 433 126	3 448 098	3 461 479
Antall dubletter (fjernet)	-	-	-	-
Antall personer under 16 år i registeret (slettes)	825	750	724	792
Antall personer med uoppgitt bosted i registeret (slettes)	25	19	20	18
Antall records i miniregisteret	3 415 609	3 432 357	3 447 354	3 460 669
Antall i offisiell statistikk (Statistisk årbok)	3 415 609	3 432 357	3 447 354	3 460 669
<u>Datokontroll i miniregisteret</u>				
<u>(fullføringstidspunkt):</u>				
Antall records med fullføringstidspunkt etter utløpet av registerperioden (skal slettes)	-	-	-	-
Antall records som mangler fullføringstidspunkt (mrk_dl=1)	93 033	98 391	86 786	91 327
Antall records hvor månedsopplysningen er rettet fra '00' til '01' (mrk_dl=2)	29 973	28 162	26 557	25 250
<u>Fødselsnummerkontroll i miniregisteret:</u>				
Antall med gyldig fødselsnummer (mrk_fnr=0)	3 415 454	3 432 304	3 447 179	3 460 465
Antall med ugyldig fødselsnummer (mrk_fnr=3)	155	53	175	204
<u>Utdanningsnivå (miniregisteret):</u>				
Grunnskole	1 061 452	1 029 666	994 257	963 265
Videregående skole	1 688 370	1 706 706	1 729 249	1 743 550
Universitets- og høyskole	572 049	596 903	629 558	655 266
Uoppgitt eller ingen fullført utdanning	93 738	99 082	94 290	98 588
<u>Offisiell statistikk:</u>				
Grunnskole	1 061 452	1 029 666	994 257	963 265
Videregående skole	1 688 370	1 706 706	1 729 249	1 743 550
Universitets- og høyskole	572 049	596 903	629 558	655 266
Uoppgitt eller ingen fullført utdanning	93 738	99 082	94 290	98 588










3.1 (forts.) Kontroller på filene for befolkningens høyeste utdanning, pr. 1/10 hvert år

	1995	1996	1997	1998
Antall records på grunnlagsregisteret	3 475 324	3 487 606	3 501 728	3 522 058
Antall dubletter (fjernet)	-	-	-	-
Antall personer under 16 år i registeret (slettes)	793	811	805	788
Antall personer med uoppgitt bosted i registeret (slettes)	17	17	14	13
Antall records i miniregisteret	3 474 514	3 486 778	3 500 909	3 521 257
Antall i offisiell statistikk (Statistisk årbok)	3 474 513	3 486 778	3 500 909	3 521 257
<u>Datokontroll i miniregisteret</u> (fullføringstidspunkt):				
Antall records med fullføringstidspunkt etter utløpet av registerperioden (skal slettes)	-	-	-	-
Antall records som mangler fullføringstidspunkt (mrk_dl=1)	94 411	96 086	101 195	109 953
Antall records hvor månedsopplysningen er rettet fra '00' til '01' (mrk_dl=2)	24 147	23 198	22 425	21 859
<u>Fødselsnummerkontroll i miniregisteret:</u>				
Antall med gyldig fødselsnummer (mrk_fnr=0)	3 474 268	3 486 778	3 500 909	3 521 257
Antall med ugyldig fødselsnummer (mrk_fnr=3)	246	-	-	-
<u>Utdanningsnivå (miniregisteret):</u>				
Grunnskole	908 087	855 733	823 963	794 835
Videregående skole	1 784 193	1 820 076	1 835 169	1 853 569
Universitets- og høyskole	680 839	708 164	734 142	756 710
Uoppgitt eller ingen fullført utdanning	101 395	102 805	107 635	116 143
<u>Offisiell statistikk:</u>				
Grunnskole	908 087	855 733	823 963	794 835
Videregående skole	1 784 193	1 820 076	1 835 169	1 853 569
Universitets- og høyskole	680 838	708 164	734 142	756 710
Uoppgitt eller ingen fullført utdanning	101 395	102 805	107 635	116 143


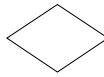
4 Diagrammer – Høyeste fullførte utdanning

Det er utarbeidet flere typer diagrammer over IT-rutinene. Diagrammene er også laget på flere nivåer. Dataflytdiagram nivå 1 gir en grov oversikt over systemet. Fysisk dataflytdiagram viser en detaljert oversikt over alle program, og sammenhengen mellom data og program. Datamodellen viser det endelige resultat, dvs. oversikt/spesifikasjon av data (forløpsdata) lagret i databasen (Oracle). Det er utarbeidet diagrammer både for rutiner i forbindelse med danning av miniregister, og rutiner for danning av forløpsdata i databasen. IT-dokumentasjonen er utarbeidet i hht. SSB's egne metode for systemutvikling «Håndbok i utvikling av statistikkssystemer - Med vekt på IT-metode».

Følgende standardsymboler fra IT-metoden er benyttet:

Symbol	Symbolnavn	Tilhørende teknikk
	Database/datasett (ORACLE og SAS)	Fysisk dataflyt
	Sekvensielle filer (flate filer)	Fysisk dataflyt
	Dokument (papirtabeller)	Fysisk dataflyt
	Program	Fysisk dataflyt, dialogsystem, DFD0
	Dataflytretning	Fysisk dataflyt/ DFD0, DFD1, dialogsystem
	Peker til fortsettelse	Fysisk dataflyt
	Prosess	DFD1
	Datalager (entitet)	(DFD0), DFD1
	Fysisk datatabell i databasen	Datamodell

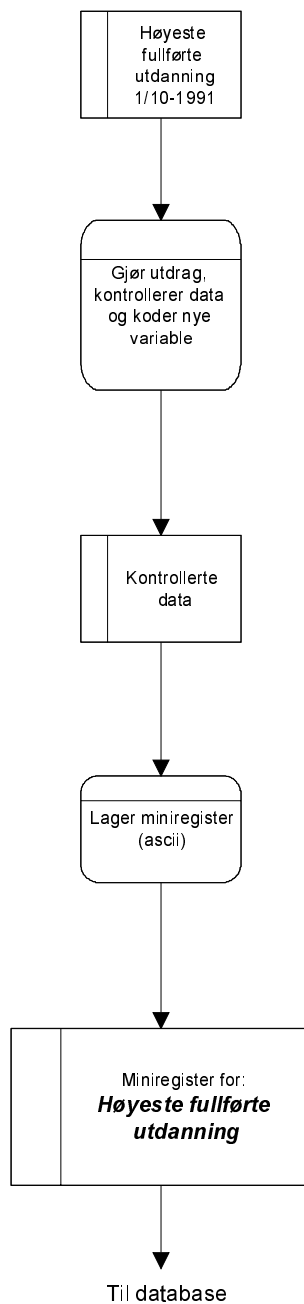
I tillegg benyttes følgende standardsymboler:

Symbol	Symbolnavn	Tilhørende teknikk
	Arbeidsoperasjon	Flytdiagram for program-rutiner
	Test	Flytdiagram for program-rutiner

4.1 Dataflyttdiagram nivå 1 - miniregister (SAS)

<i>FD-TRYGD</i>	X:350\TRYGD\FOB\EDBDOK\DIAGRAM\UTDANN\HOYFULL\MR_FLYT.AF3
	Diagramansvarlig: JLa 5/4-2000
DFD1 - Danning av miniregister for høyeste fullførte utdanning	

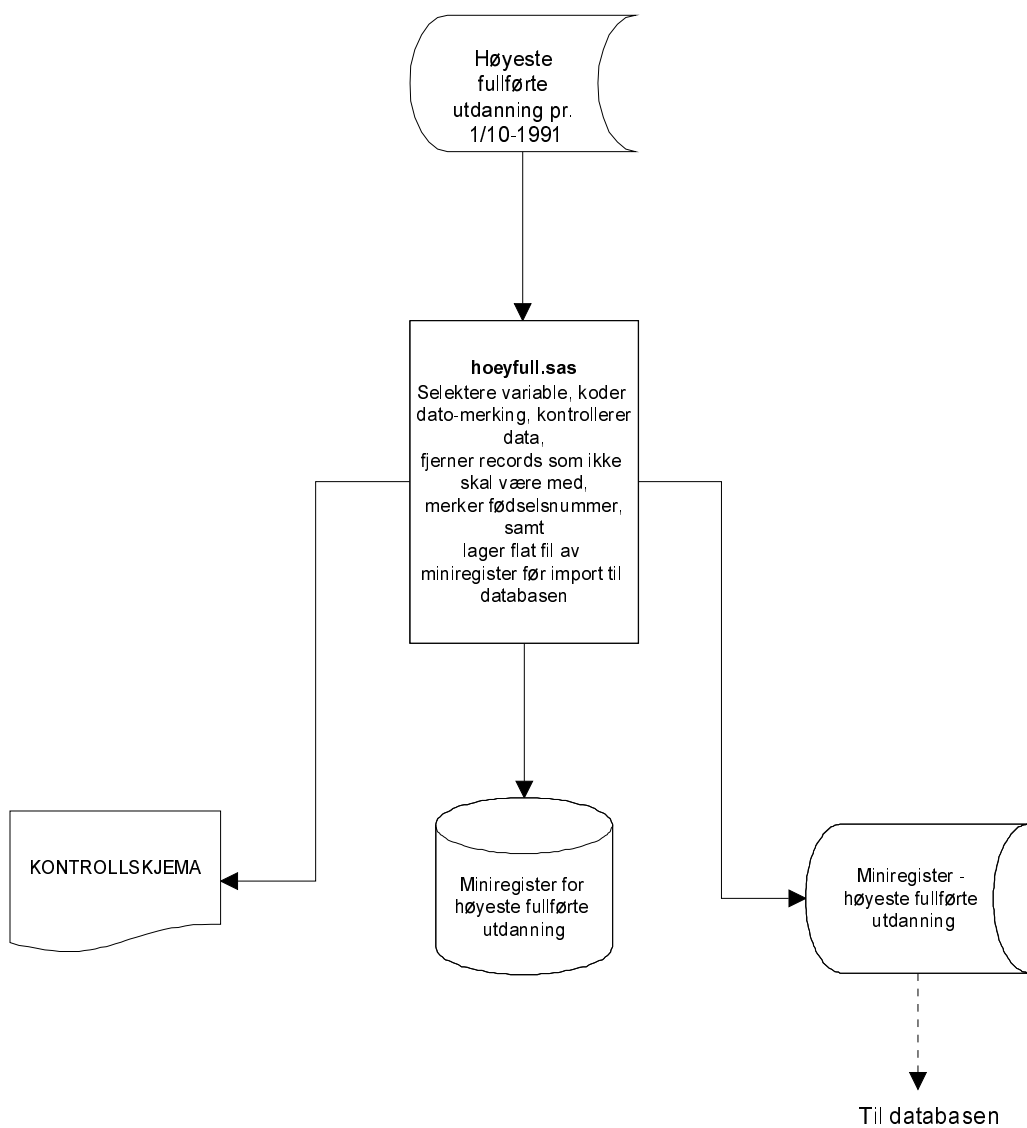
DFD1 gjelder tilsvarende for de senere årgangene.



4.2 Fysisk dataflytdiagram - miniregister (SAS)

<i>FD-TRYGD</i>	X:350\TRYGD\FOB\EDBDOK\DIAGRAM\UTDANN\HOYFULL\MR_FLYT.AF3
	Diagramansvarlig: JLa 5/4-2000
Fysisk dataflytdiagram - Danning av miniregister for høyeste fullførte utdanning	

Diagrammet gjelder tilsvarende for de senere årgangene.



4.3 Programbeskrivelse av miniregister-rutiner (SAS)

Nedenfor følger programheadinger til alle programmer som er benyttet til danning og kontroll av miniregistre på utdanning. Programmer som er benyttet, ligger på Unix.

4.3.1 Høyeste fullførte utdanning

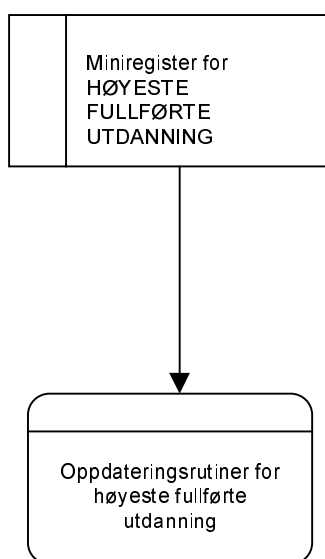
Programmet ligger under \$TRYGDFOB/prog/SAS/utdann/gYY/hoeyfull/...

```

/*****
Prosjekt .....: FD-Trygd
Program navn .....: hoeyfull.sas
Skrevet av .....: JLa
Dato .....: 24.09.1997
Versjon .....: 1.00
Programmets funksjon ..: Lage sasfil av høyeste fullførte utdanning,
                        kontrollerer fullføringsdatoer,
                        sjekker fødselsnummer og dubletter og
                        skriver miniregisteret til tekstfil.
Programmet kaller ....: Makro for fødselsnummerkontroll
Filer inn.....: $TRYGDFOB/data/utdann/g91/grunnlag/hoeyfull/g91.txt
Filer ut.....: $TRYGDFOB/data/utdann/g91/minireg/hoeyfull/g91.txt
                  $TRYGDFOB/data/utdann/g91/minireg/hoeyfull/g91.ssd01
Endret når .....: 07.02.1998
Endret av .....: JLa
Grunn til endring ....: Lage rutinen fulstendig
Merknader .....: Husk å rette fødselsår for 15-åringene som skal
                  slettes
*****/
```

4.4 Dataflytdiagram nivå 1 - database (Oracle)

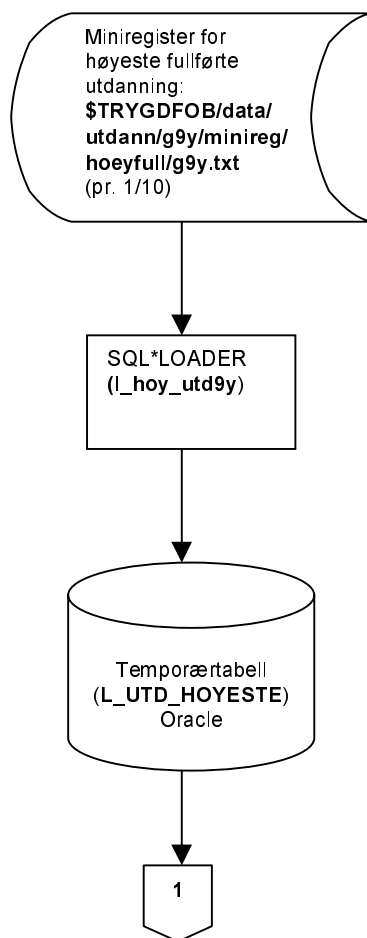
<i>FD-TRYGD</i>	Diagramansvarlig: JoJ	Dato: 28.02.2000
DFD1 - Databaseprosesser som danner forløp for HØYESTE FULLFØRTE UTDANNING		

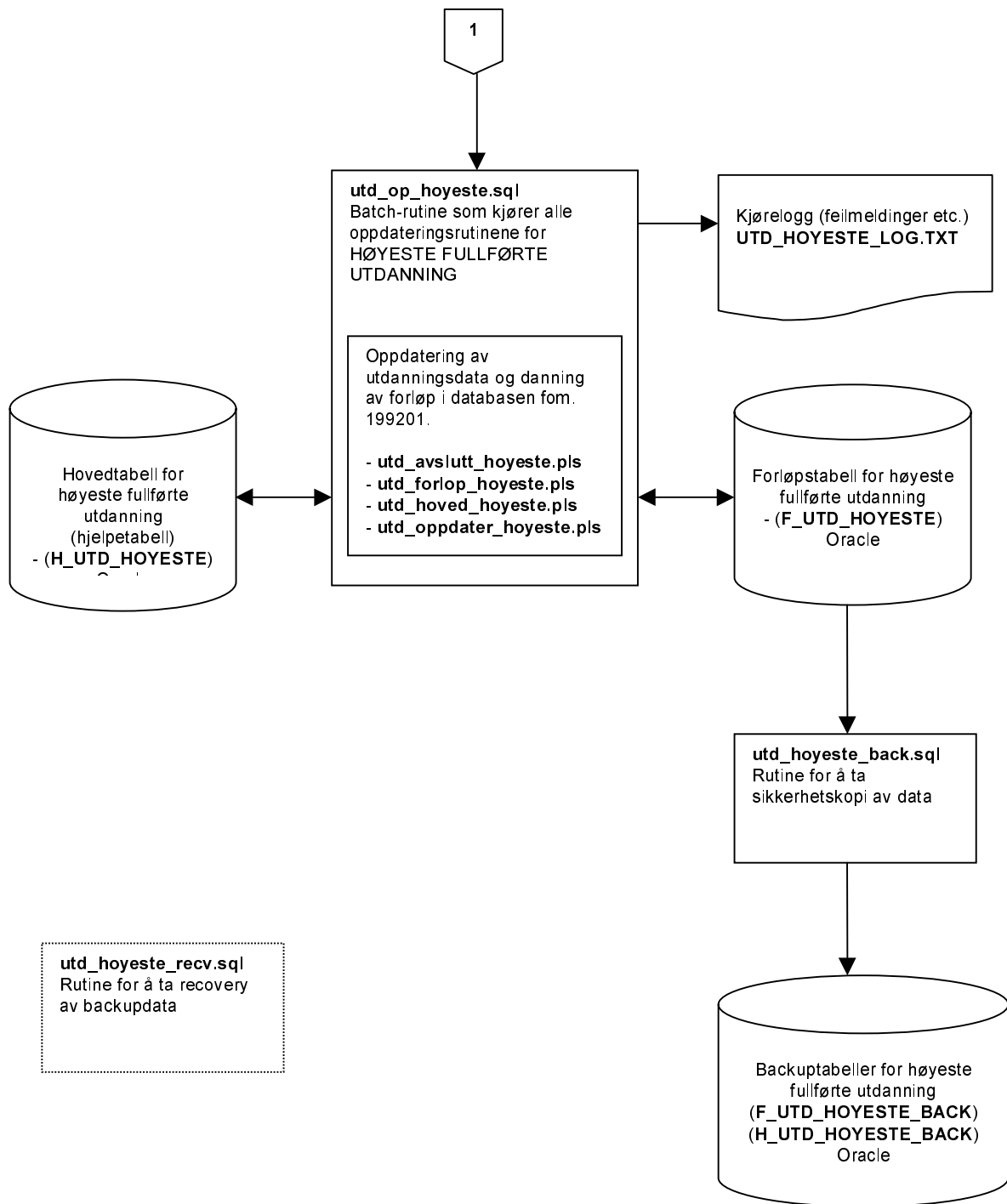


4.5 Fysisk dataflytdiagram - database (Oracle)

<i>FD-TRYGD</i>	Diagramansvarlig: JoJ	Dato: 25.02.2000
Fysisk dataflytdiagram - databaserutiner for HØYESTE FULLFØRTE UTDANNING		

y = siste siffer i årstall





4.6 Databaserutiner for høyeste fullførte utdanning (Oracle)

Datagrunnlaget for å danne forløp for høyeste fullførte utdanning består av årsfiler med befolkningens høyeste fullførte utdanning pr. 1/10 hvert år.

4.6.1 Lagringssted for databaserutiner

Programmene som er benyttet for å danne forløp for iganværende utdanning i Oracle-databasen er lagret på UNIX under følgende kataloger:

```
$STRYGDFOB/prog/ORACLE/utdann/hoyeste//BACKUP/.. .  
$STRYGDFOB/prog/ORACLE/utdann/hoyeste/DATATEST/..  
$STRYGDFOB/prog/ORACLE/utdann/hoyeste/OPPDATER/..  
$STRYGDFOB/prog/ORACLE/utdann/hoyeste/SQLLOAD/.. .
```

4.6.2 Innlastingshastighet under danning av forløp

Rutinene som danner forløp for utdanning er store og meget krevende for serveren. Disse bør ikke parallellkjøres med andre innleggingsrutiner eller andre krevende spørringer. Erfaring har vist at innleggingshastigheten synker dramatisk i perioder hvor serveren har mange prosesser å arbeide med samtidig (innlastingstiden kan variere med flere timer for en og samme årgang alt etter som hvor mye serveren har å gjøre).

TIPS:

Følg med på *top* på UNIX før danning av forløp startes for å se om andre krevende og/eller lange kjøringene foregår.

1. Prøv å finne ut hvilke tidspunkter/tidsluker hvor det er generelt liten trafikk på serveren. DBA og/eller systemoperatørene på serveren kan gi svar på dette.
2. Prøv å avtale med andre som bruker serveren til tunge oppgaver om å få en «ledig» tidsluke for deg selv på serveren.
3. Det er laget en liten prosedyre som baserer seg på UTL_FILE (logg_status) som gjør det mulig å følge med på innlastingshastigheten under danning av forløp.

På UNIX: se på filen:

```
/u01/app/oracle/admin/TFH3/ora_out/utd_hoyeste_log.txt
```

Filen vil inneholde all informasjon om innlastingen (hastighet, tidsforbruk samt evt. feil som har oppstått). En innlastingshastighet på over 180 records pr. sekund er bra, under 100-120 records pr. sekund er mindre bra. "Samplingshastigheten" kan endres i rutinen

```
UTD_OPPDATER_HOYESTE.PLS.
```

4.6.3 Kontroll av batch-kjøring i databasen

Kontroller at resultatet av batch-kjøringen er som forventet. Loggfilene som er beskrevet i punkt 4.6.2 skal **alltid** sjekkes for ev. feilmeldinger etter hver kjøring!

4.7 Programbeskrivelse av db-rutiner for høyeste fullførte utdanning (Oracle)

4.7.1 SQL*Loader-rutine for import til Oracle

Før nye data fra et årssett legges inn i den temporære tabellen **L_UTD_HOYESTE** må en først slette evt. gammelt innhold i disse. Dette gjøres enklest med kommandoen «**TRUNCATE TABLE <tabellnavn>**».

Vi starter med å laste data fra flate filer inn i de temporære tabellene vha. egne innlastingskript for SQL*Loader. Under UNIX-katalogen \$STRYGDFOB/prog/ORACLE/utdann/hoyeste/SQLLOAD/ kjøres innlastingskriptene slik: **SQLLOAD trygdfob/<passord>@tfh3 l_hoy<YY>.ctl DIRECT = TRUE** hvor YY = år. Data er nå lagt inn i tabellen **L_UTD_HOYESTE**, og er klar for oppdateringsrutinene for å danne forløp på høyeste fullførte utdanning i databasen.

4.7.2 UTD_OP_HOYESTE.SQL (Hovedprog. for oppdatering av høyeste fullførte utdanning)

Rutinen fungerer som hovedprogram for oppdatering av høyeste fullførte utdanning **fom. 199201** i databasen og kaller ulike sub-program for danning av forløpsdata. Data lastes inn i loadtabellen **L_UTD_HOYESTE** for ett og ett år som beskrevet i punktet over. Hovedprogrammet prosesserer hvert årssett av data fortløpende. Dette gjøres med rutinen **STRYGDFOB/prog/ORACLE/utdann/hoyeste/OPPDATER/utd_op_hoyeste.sql**.

Alle personer i nyeste datagrunnlag, dvs. personer i den temporære tabellen **L_UTD_HOYESTE**, sjekkes mot forrige års datagrunnlag (innholdet i hovedtabellen **H_UTD_HOYESTE**), record for record. Ut i fra bestemte regler, som angitt i 4.8, avgjøres det om det skal kodes en *tilgang*, *avgang* eller *endring* for hver enkelt record som prosesseres. *Endring* betyr endret verdi i en eller flere av internvariablene siden forrige registrering.

Kildekoden til dette programmet er lagt inn i dokumentasjonen for at det skal være lettere å se rekkefølgen rutinene kompiles i:

```

/*****
Prosjekt .....: FD-Trygd (HØYESTE FULLFØRTE UTDANNING)
Program navn .....: utd_op_hoyeste.sql
Skrevet av .....: JoJ
Dato .....: 11.11.1999
Versjon .....: 1.0
Programmets funksjon ..: Batch-jobb som kaller oppdateringsrutinen for
                        oppdatering av høyeste fullførte utdanning i
                        forløpstabellen.
Programmet kaller ....: utd_oppdater_hoyeste.pls
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/

/* Sender meldinger til egen logg-fil */
SPOOL utd_op_hoyeste
```

```

* Kompilerer alle PL/SQL-prosedyrer som skal benyttes */

@utd_hoved_hoyeste.pls /
@utd_forlop_hoyeste.pls /
@utd_avslutt_hoyeste.pls /
@utd_oppdater_hoyeste.pls /
COMMIT;

/* Setter timer på (bare for å se hvor lang tid jobben tar) */
SET TIMING ON;
SET TIME ON;

/* Starter oppdatering av høyeste fullførte utdanning */
EXECUTE utd_oppdater_hoyeste;
COMMIT;

/* Mailer loggen når rutinen er ferdig */
!mail joj@ssb.no < /u01/app/oracle/admin/TFH3/ora_out/utd_hoyeste_log.txt

/* Sletter prosedyrene etter bruk */
DROP PROCEDURE utd_hoved_hoyeste;
DROP PROCEDURE utd_forlop_hoyeste;
DROP PROCEDURE utd_avslutt_hoyeste;
DROP PROCEDURE utd_oppdater_hoyeste;
COMMIT;

/* Lukker spool-meldings-filen */
SPOOL OFF

EXIT;

```

4.7.3 UTD_HOVED_HOYESTE.PLS

```

/*****
Prosjekt .....: FD-Trygd (HØYESTE FULLFØRTE UTDANNING)
Program navn .....: utd_hoved_hoyeste.pls
Skrevet av .....: JoJ
Dato .....: 11.11.1999
Versjon .....: 1.0
Programmets funksjon ..: Legger behandlede records ut i hovedtabellen for
                        høyeste fullførte utdanning: H_UTD_HOYESTE.

Programmet kaller ....:
Programmet kalles fra : utd_oppdater_hoyeste.pls
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/

```

4.7.4 UTD_FORLOP_HOYESTE.PLS

```

/*****
Prosjekt .....: FD-Trygd (HØYESTE FULLFØRTE UTDANNING)
Program navn .....: utd_forlop_hoyeste.pls
Skrevet av .....: JoJ
Dato .....: 11.11.1999
Versjon .....: 1.0
Programmets funksjon .: Legger behandlede records ut i forløpstabellen for
                        høyeste fullførte utdanning: F_UTD_HOYESTE.

Programmet kaller ....:
Programmet kalles fra : utd_oppdater_hoyeste.pls
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

4.7.5 UTD_AVSLUTT_HOYESTE.PLS

```

/*****
Prosjekt .....: FD-Trygd (HØYESTE FULLFØRTE UTDANNING)
Program navn .....: utd_avslutt_hoyeste.pls
Skrevet av .....: JoJ
Dato .....: 11.11.1999
Versjon .....: 1.0
Programmets funksjon .: Avslutter tilfellet ett år etter siste
                        "livstegn" fra lastetabellen for høyeste fullførte
                        utdanningkurs (L_UTD_HOYESTE).

Programmet kaller ....:
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

4.7.6 UTD_OPPDATER_HOYESTE.PLS

```

/*****
Prosjekt .....: FD-Trygd (HØYESTE FULLFØRTE UTDANNING)
Program navn .....: utd_oppdater_hoyeste.pls
Skrevet av .....: JoJ
Dato .....: 11.11.1999
Versjon .....: 1.0
Programmets funksjon .: Oppdatere høyeste fullførte utdanning i
                        forløpsdatabasen med data fra ny årgangsfil.
                        For å oppdatere tabellene brukes DELETE og INSERT i
                        kombinasjon da dette har vist seg å være mer
                        effektivt enn UPDATE med det RAID-systemet (RAID5)
                        som SSB har valgt.

Programmet kalles fra : utd_op_hoyeste.sql
Filer inn .....:
Filer ut .....:
Endret når .....:
Endret av .....:
Grunn til endring ....:
*****/
```

4.7.7 UTD_HOYESTE_BACK.SQL (backup-rutine)

Det er spesielt viktig å merke seg at batch-rutinene som legger ett og ett årssett med utdanningsdata i databasen (danner forløp), bygger på / jobber mot, forrige års data (resultatet av forrige års kjøring).

Det vil igjen si at dersom en har lagt inn n antall årssett med data og en batch-rutine bryter/krasjer, må ALLE data legges inn på nytt. For å unngå dette er det utarbeidet en egen backup-rutine.

Denne kjøres med skriptet

\$STRYGDFOB/prog/ORACLE/utdann/hoyeste/BACKUP/utd_hoyeste_back.sql.

Kjør rutinen etter innlegging av hvert årssett når du er sikker på at forrige kjøring gikk bra.

4.7.8 UTD_HOYESTE_RECV.SQL (recovery-rutine)

Dersom innlegging av nye data forårsaker feil som stopper rutinen under kjøring, eller på annen måte ødelegger tidligere innlagte data, kan gamle data hentes tilbake ved hjelp av denne rutinen. Det

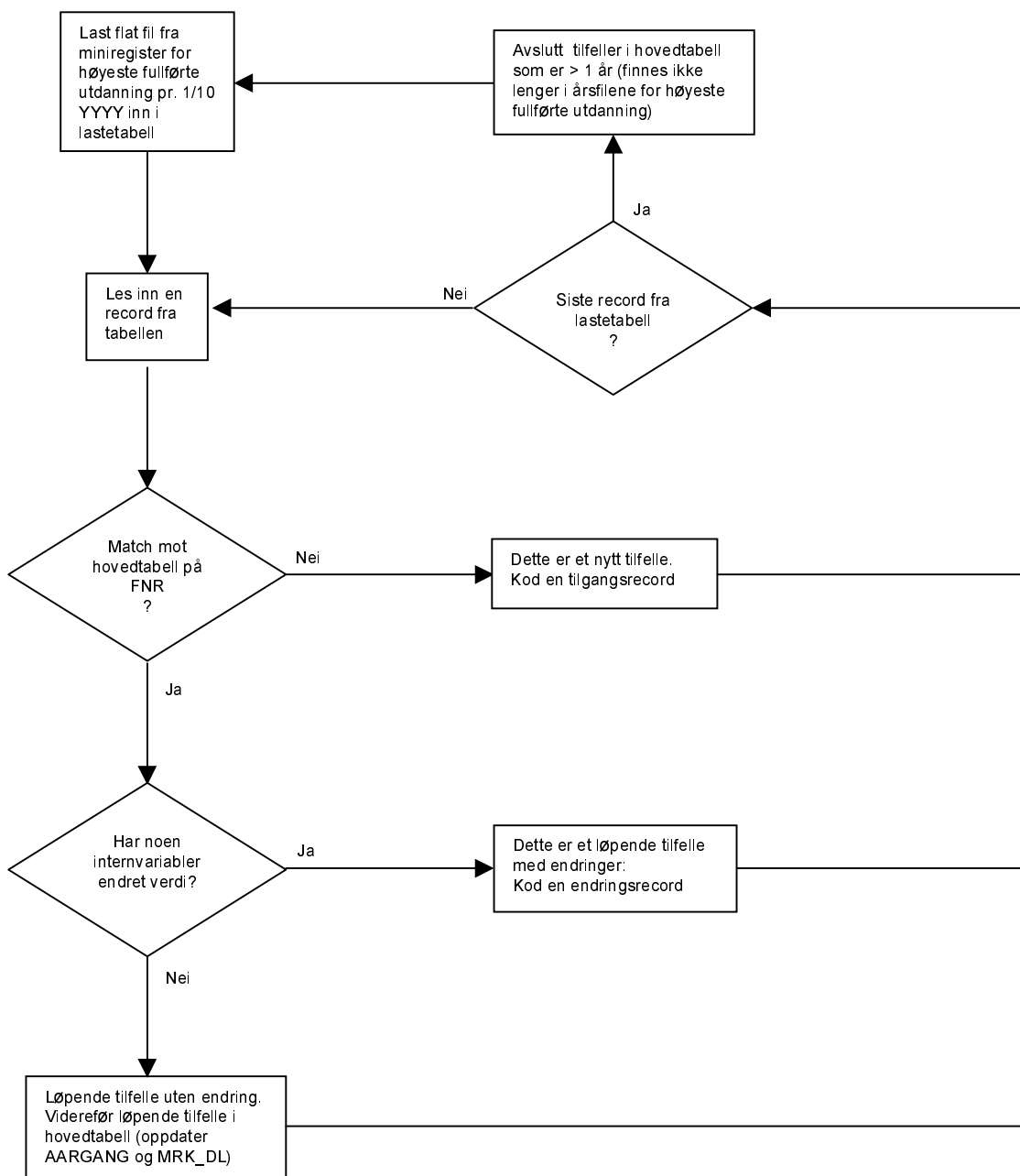
forutsettes at rutinen UTD_HOYESTE_BACK.SQL er kjørt FØR siste innleggingsforsøk. Rutinen henter da tilbake alle data tom. forrige årssett. Rutinen kjøres med skriptet

\$STRYGDFOB/prog/ORACLE/utdann/hoyeste/BACKUP/utd_hoyeste_recv.sql.

4.8 Flytdiagram for PL/SQL-kode til høyeste fullførte utdanning

FD-TRYGD	Diagramansvarlig: JoJ	Dato: 06.04.2000
Forenklet flytdiagram for PL/SQL-kode til HØYESTE FULLFØRTE UTDANNING		

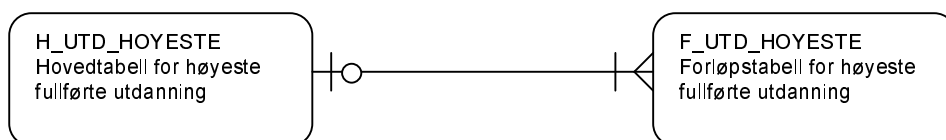
Rutiner for å danne forløp for høyeste fullførte utdanning



4.9 Datamodell for høyeste fullførte utdanning

<i>FD-TRYGD</i>	Diagramansvarlig: JoJ	Dato: 25.02.2000
Datamodell - HØYESTE FULLFØRTE UTDANNING		

Miniregisterene for HØYESTE FULLFØRTE UTDANNING kjøres mot sine respektive hoved- og avgangstabeller (som inneholder status for nyeste datagrunnlag) for oppdatering og danning av forløp.



4.10 Modellspesifikasjoner for høyeste fullførte utdanning

Alle definisjoner av Oracle databasetabeller, med tilhørende kommentarer og kodelister, ligger lagret i Oracle Repository. Oracle Designer 2000 er benyttet som modelleringsverktøy.

I databasen er det også etablert backup-tabeller som ikke er beskrevet nærmere i denne rapporten da disse har samme struktur som sine respektive originaltabeller:

F_UTD_HOYESTE_BACK
H_UTD_HOYESTE_BACK

4.10.1 Loadtabell for innlasting av høyeste fullførte utdanning

Tabellnavn: L_UTD_HOYESTE (loadtabell)

Generelle kommentarer:

Årsfiler for høyeste fullførte utdanning pr 01.10.YYYY lastes inn i en loadtabell (mottakstabell) før bearbeiding i databasen. Årsfilene inneholder ca. 3 400 000 - 3 500 000 records.

Variabelbeskrivelse for tabellen: L_UTD_HOYESTE

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Fødselsnummer FNR VARCHAR2 11	Også records med ugyldige fødselsnumre tas med i FD-Trygd.
Utdannings art NUS VARCHAR2 6	Disse kodene følger "Standard for utdanningsgruppering" . NUS-koden på 6 siffer.
Klassetrinn KLTRINN NUMBER 2	Viser det klassetrinn utdanningen vanligvis avsluttes på.
Fullføringstidspunkt FULLF NUMBER 6	Viser når utdanningen er fullført. Format: YYYYMM
Miniregisterkontroll - fødselsnummer MRK_FNR VARCHAR2 1	Merking av fødselsnummer. Gyldige koder: 0 = Gyldig fødselsnummer 1 = Ugyldig fødselsnummer, men gyldig d-nummer 2 = Ugyldig fødselsnummer bestående av blankt personnummer 3 = Ugyldig fødselsnummer som ikke omfattes av kode 1 eller 2
Miniregisterkontroll - datokontroll MRK_DL VARCHAR2 1	Datokontroll av fullføringstidspunktet. Gyldige koder: 1 = Fullføringstidspunktet mangler, settes til NULL 2 = Månedsopplysning er rettet fra 00 til 01 0 = Ingen endring
Årgang AARGANG NUMBER 4	Viser hvilken årsfil record kommer fra. Format: YYYY

4.10.2 Forløps- og hovedtabell for høyeste fullførte utdanning

Tabellnavn: F_UTD_HOYESTE (forløpstabell)
H_UTD_HOYESTE (hovedtabell)

Kommentarer til forløpstabellen: F_UTD_HOYESTE:

Inneholder forløp for høyeste fullførte utdanning og er generert ved hjelp av årsfiler pr. 01.10.YYYY. For hver årsfil som "kjøres" mot forløpsdatabasen, genereres det ca. 330 000 - 390 000 nye records i forløpstabellen (tilganger, endringer og avganger).

Kommentarer til hovedtabellen: H_UTD_HOYESTE:

Dette er en hjelpetabell som inneholder siste datagrunnlag som er benyttet for å danne forløp i databasen. Inneholder kun en record pr. utdanningstilfelle (pr. FNR) og blir liggende i tabellen til vi koder en avgang for tilfellet (helt ut av systemet). Inneholder til enhver tid ca. 3 400 000 - 3 500 000 records.

Generelle kommentarer:

Strukturen (alle kolonner) i tabellene F_UTD_HOYESTE og H_UTD_HOYESTE er identisk.

Alle utdanningsdata «knyttes til» en variabel kalt KODE som sier noe om hendinger/endringer som er tilknyttet høyeste fullførte utdanning.

For mer utfyllende informasjon om hvordan de ulike variabler kodes, se flytdiagram for PL/SQL-kode kapittel 4.5.

Variabelbeskrivelse for tabellene: F_UTD_HOYESTE og H_UTD_HOYESTE

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Fødselsnummer FNR VARCHAR2 11	Også records med ugyldige fødselsnumre tas med i FD-Trygd.
Hendelseskode KODE VARCHAR2 1	Settes i databasen og sier noe om hendelser/endringer tilknyttet høyeste fullførte utdanning. Gyldige koder: 0 = avgang 1 = tilgang 2 = endring (i internvariabel) 3 = venstresensur (startet før 199201)
Registreringsdato REGDATO NUMBER 4	Regdato vil fungere som en siste oppdaterings-/endringsdato for hele recorden sett som en helhet, uavhengig om det er tilgang, avgang eller endring. Format: YYYY
Tilfelleets tilgangsdato TILGDATO NUMBER 4	Tilgangsdato for utdanningstilfellet. Settes ved tilgang til nytt utdanningstilfellet og endres ikke. Format: YYYY
Tilfelleets avgangsdato AVGDATO NUMBER 4	Avgangsdato for utdanningstilfellet. Settes når utdanningstilfellet avsluttes, ellers blank (NULL). Format: YYYY
Utdannings art NUS VARCHAR2 6	Disse kodene følger "Standard for utdanningsgruppering" . Dette er hele NUS-koden (6 siffer)
Endringsdato for NUS NUSDATO NUMBER 4	AARGANG brukes for å datere endringer. Format: YYYY
Utdannings art NUS1 VARCHAR2 1	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder første siffer for NUS-koden.
Endringsdato for NUS1 NUS1DATO NUMBER 4	AARGANG brukes for å datere endringer. Format: YYYY
Utdannings art NUS2 VARCHAR2 2	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de to første siffer for NUS-koden.
Endringsdato for NUS2 NUS2DATO NUMBER 4	AARGANG brukes for å datere endringer. Format: YYYY
Utdannings art NUS3 VARCHAR2 3	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de tre første siffer for NUS-koden.

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Endringsdato for NUS3 NUS3DATO NUMBER 4	AARGANG brukes for å datere endringer. Format: YYYY
Utdannings art NUS4 VARCHAR2 4	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de fire første siffer for NUS-koden.
Endringsdato for NUS4 NUS4DATO NUMBER 4	AARGANG brukes for å datere endringer. Format: YYYY
Klasstrinn KLTRINN NUMBER 2	Viser det klasstrinn utdanningen vanligvis avsluttes på.
Endringsdato for klasstrinn KLTRDATO NUMBER 4	AARGANG brukes for å datere endringer. Format: YYYY
Fullføringstidspunkt FULLF NUMBER 6	Viser når utdanningen er fullført. Format: YYYYMM
Endringsdato for fullføringstidspunkt FULFDATO NUMBER 4	AARGANG brukes for å datere endringer. Format: YYYY
Miniregisterkontroll - datokontroll MRK_DL VARCHAR2 1	Datokontroll av fullføringstidspunktet. Gyldige koder: 1 = Fullføringstidspunktet mangler, settes til NULL 2 = Månedsopplysning er rettet fra 00 til 01 0 = Ingen endring
Miniregisterkontroll - fødselsnummer MRK_FNR VARCHAR2 1	Merking av fødselsnummer. Gyldige koder: 0 = Gyldig fødselsnummer 1 = Ugyldig fødselsnummer, men gyldig d-nummer 2 = Ugyldig fødselsnummer bestående av blankt personnummer 3 = Ugyldig fødselsnummer som ikke omfattes av kode 1 eller 2

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Endringskode for internvariablene ENDRKODE NUMBER 6	<p>Settes ved danning av forløp i databasen og viser hvilke internvariable som evt. har endret verdi siden forrige record i forløpstabellen.</p> <p>Hver internvariabel får en unik endrekode (verdi) og disse verdiene summeres dersom endringer har funnet sted på flere internvariable samtidig. Summen angir en unik kombinasjon av endringer. De enkelte endringene har følgende verdier:</p> <p>0 = Ingen endring i noen internvariabler 1 = NUS er endret 2 = NUS1 er endret 4 = NUS2 er endret 8 = NUS3 er endret 16 = NUS4 er endret 32 = KLTRINN er endret 64 = FULLF er endret</p> <p>Hvis en record har f.eks. endrkode = 33 betyr det at både NUS og KLTRINN er endret.</p>
Årgang AARGANG NUMBER 4	<p>Viser hvilken årsfil record kommer fra. Oppdateres for løpende for tilfeller uten endringer.</p> <p>Format: YYYY</p>

5 Spesifikasjonsnotat – Igangværende utdanning

For å finne hva slags utdanning befolkningen er igang med, og for å finne forløpet på disse utdanningene bruker vi flere typer grunnlagsregistre.

5.1 Beskrivelse av grunnlagsregistrene

5.1.1 Filidenter

Igangværende utdanning pr. 1/10

På stormaskinen hadde filene disse identene:

PL213.S4353.I630E1A1.G9100.V00

PL213.S4353.I630E1A1.G9200.V00

PL213.S4353.I630E1A1.G9300.V00

I 1999 er utdanningsfilene lagt over til unix, og filidentene er:

\$UTD/utdbase/arkiv/igang/g1994.dat

\$UTD/utdbase/arkiv/igang/g1995.dat

\$UTD/utdbase/arkiv/igang/g1996.dat

\$UTD/utdbase/arkiv/igang/g1997.dat

\$UTD/utdbase/arkiv/igang/g1998.dat

De filene vi danner oss som grunnlagsregistre har fått følgende filidenter:

\$TRYGDFOB/data/utdann/g91/grunnlag/igangv/g91.ssd04

\$TRYGDFOB/data/utdann/g92/grunnlag/igangv/g92.ssd04

\$TRYGDFOB/data/utdann/g93/grunnlag/igangv/g93.ssd04

\$TRYGDFOB/data/utdann/g94/grunnlag/igangv/g94.ssd04

\$TRYGDFOB/data/utdann/g95/grunnlag/igangv/g95.ssd04

\$TRYGDFOB/data/utdann/g96/grunnlag/igangv/g96.ssd04

\$TRYGDFOB/data/utdann/g97/grunnlag/igangv/g97.ssd04

\$TRYGDFOB/data/utdann/g98/grunnlag/igangv/g98.ssd04

Avslutta utdanning i “skole”-året

På stormaskin hadde disse filene følgende identer:

PL213.S4353.I630C2A1.G9192.V00

PL213.S4353.I630C2A1.G9293.V00

PL213.S4353.I630C2A1.G9394.V00

I 1999 er utdanningsfilene lagt over til unix, og filidentene er:

\$UTD/utdbase/arkiv/avslutt/g1994_1995.dat

\$UTD/utdbase/arkiv/avslutt/g1995_1996.dat

\$UTD/utdbase/arkiv/avslutt/g1996_1997.dat

\$UTD/utdbase/arkiv/avslutt/g1997_1998.dat

De filene vi danner oss som grunnlagsregistre har fått følgende filidenter:

\$TRYGDFOB/data/utdann/g92/grunnlag/avslutt /g92.ssd04

\$TRYGDFOB/data/utdann/g93/grunnlag/avslutt /g93.ssd04

\$TRYGDFOB/data/utdann/g94/grunnlag/avslutt /g94.ssd04

\$TRYGDFOB/data/utdann/g95/grunnlag/avslutt /g95.ssd04

\$TRYGDFOB/data/utdann/g96/grunnlag/avslutt /g96.ssd04

\$TRYGDFOB/data/utdann/g97/grunnlag/avslutt /g97.ssd04

\$TRYGDFOB/data/utdann/g98/grunnlag/avslutt /g98.ssd04

Semesterfiler

For studenter på universiteter, erstatter vi **studiemål** med **fag i semesteret** for filene for 1991-1996. Begge disse variablene er kodet ut fra utdanningens art (NUS). Til dette bruker vi høstsemesterfiler. Høstsemesterfila hadde følgende filident på stormaskin:

PL213.S4354.I630C5A1.G9100.V00

PL213.S4354.I630C5A1.G9200.V00

PL213.S4354.I630C5A1.G9300.V00

I 1999 er utdanningsfilene lagt over til unix, og filidentene er:

\$UTD/utdbase/arkiv/igang_fag_i_semester/g1994.dat

\$UTD/utdbase/arkiv/igang_fag_i_semester/g1995.dat

\$UTD/utdbase/arkiv/igang_fag_i_semester_netto/g1996.dat

De filene vi danner oss som grunnlagsregistre har fått følgende filidenter:

\$TRYGDFOB/data/utdann/g91/grunnlag/fagisem/g91h2.ssd04

\$TRYGDFOB/data/utdann/g92/grunnlag/fagisem/g92h2.ssd04

\$TRYGDFOB/data/utdann/g93/grunnlag/fagisem/g93h2.ssd04

\$TRYGDFOB/data/utdann/g94/grunnlag/fagisem/g94h2.ssd04

\$TRYGDFOB/data/utdann/g95/grunnlag/fagisem/g95h2.ssd04

\$TRYGDFOB/data/utdann/g96/grunnlag/fagisem/g96h2.ssd04

Vårsemesterfilene bruker vi til å legge inn startdato for universitetsstudenter på avsluttafilene for årgangene 1992-1997. Vårsemesterfilene hadde følgende filident på stormaskin:

PL213.S4354.I340D0B2.G92H1.V00

PL213.S4354.I340D0B2.G93H1.V00

PL213.S4354.I340D0B2.G94H1.V00

I 1999 er utdanningsfilene lagt over til unix, og filidentene er:

\$UTD/utdbase/arkiv/studigangvaar/g1995.dat

\$UTD/utdbase/arkiv/studigangvaar/g1996.dat

\$UTD/utdbase/arkiv/studigangvaar/g1997.dat

De filene vi danner oss som grunnlagsregistre har fått følgende filidenter:

\$TRYGDFOB/data/utdann/g92/grunnlag/fagisem/g92h1.ssd04

\$TRYGDFOB/data/utdann/g93/grunnlag/fagisem/g93h1.ssd04

\$TRYGDFOB/data/utdann/g94/grunnlag/fagisem/g94h1.ssd04

\$TRYGDFOB/data/utdann/g95/grunnlag/fagisem/g95h1.ssd04

\$TRYGDFOB/data/utdann/g96/grunnlag/fagisem/g96h1.ssd04

\$TRYGDFOB/data/utdann/g97/grunnlag/fagisem/g97h1.ssd04

5.1.2 Filbeskrivelse

Igangværende utdanning pr. 1/10 og avslutta utdanning i løpet av året

Igangværendefilene omfatter alle personer som er i gang med en utdanning ut over ordinær grunnskole pr. 1/10. (SSB har ikke individ-data for igangværende utdanning på ordinær grunnskole.)

På videregående nivå omfatter filene utdanning med varighet på minst 300 timer, på høyskoler minst 5 vekttall i semesteret. Personer med påbegynningsstidspunkt f.eks. i januar 1992 og som er igangværende pr. 1/10 samme år, ligger på fila.

Igangværende filene er nettofiler. Det vil si at selv om en person er registrert å være i gang med flere kurs/enkeltutdanninger pr. 1/10, så er bare en av disse utdanningene tatt med i dette registeret.

Utplukket blir gjort ved at man først velger den utdanningen som er registrert som heltid. Hvis flere utdanninger er registrert som heltid, velger man den utdanningen som har det nyeste starttidspunktet.

Hvis man fortsatt har flere utdanninger med heltid, velger man den utdanningen som har høyest klassetrinn.

Avsluttafilene omfatter alle personer som avslutter grunnskolen, eller avslutter/avbryter videregående skole eller høyere utdanning. Avsluttafila for år t inneholder opplysninger om avsluttet utdanning i perioden 1/10 år t-1 til 30/9 år t. Vi starter med opplysningene fra avsluttafil 1992.

Filen er antakelig ufullstendig med hensyn til de som avbryter. Filen regnes ellers å være av brukbar kvalitet, siden det er disse filene som brukes til å oppdatere registeret over høyeste fullførte utdanning.

Avsluttafilene er bruttofiler. Det vil si at alle kurs/enkeltutdanninger som en person avslutter (fullfører eller avbryter) i løpet av et år blir registrert i fila.

Filbeskrivelsene for igangværende- og avsluttafilene er like, og vi behandler derfor disse under ett. Imidlertid markerer vi i tabellen nedenfor hvilke variable vi tar med fra hvilke registre.

Posisjon	Igang	Avsl	Tekst
1- 6	x	x	Utdanningens art. Denne variabelen tilsvare variabelen utdanningens art (omkodet) i filen for høyeste fullførte utdanning. Kodene følger «Standard for utdanningsgruppering».
7- 8	x	x	Klassetrinn. For studenter ved universitetene har fagseksjonen et eget program for oppjusteringen av klassetrinn. For videregående skoler og høyskoler settes klassetrinn av skolen med utgangspunkt i kurset eleven går på. Hvis studenten skifter mellom høyskole og universitet, eller endrer studiemål, kan klassetrinn derimot være mindre representativ for den faktiske progresjonen studenten har.
9	x	x	Heltid/deltid. Dette er en variabel som i utgangspunktet blir satt av skolen ut ifra om kurset er ment å utgjøre et fulltidsstudium. På avslutta filene blir alle studenter ved universitetet kodet som heltidsstudenter. På igangværende filene blir imidlertid universitetsstudentene registrert på heltid/deltid etter gitte kriterier - f.eks. forventet studieprogresjon. Kodene er: Heltid=1 Deltid=2.
10-13	x	x	Kursets påbegynningsstidspunkt (MMYY)
14-17		x	Kursets avslutningsstidspunkt (MMYY)
18-24	x	x	Skolenummer. Variabelen brukes (sammen med klassebetegnelse) til å kode arbeidsmarkeds-kurs (AMO-kurs) for 1992 og 1993, men denne variabelen skal ikke inn i basen
25-27	x	x	Klassebetegnelse. Variabelen brukes (sammen med skolenummer) til å kode AMO-kurs for 1992 og 1993, men denne variabelen skal ikke inn i basen
28	x	x	Voksenopplæring. Variabelen viser om kurset er en utdanning som går inn under lov om voksenopplæring. Variabelen har kodene: Voksenopplæring=1 Ikke voksenopplæring=2 Uoppgitt=0 (denne koden legges inn av oss)
31-41	x	x	Fødselsnummer. Også ugyldige fødselsnummer skal med i basen.

Posisjon	Igang	Avsl	Tekst
42		x	Utfall. Variabelen viser om personen har avbrutt eller fullført kurset. Hovedregelen er at utdanning regnes som fullført når vitnemål utstedes. For noen kurs blir det imidlertid ikke utstedt vitnemål. Disse blir også registrert som fullført. Kodene for variabelen er: Avbrutt=2 Fullført=8 Fra og med 1995-årgangen blir det innført flere koder for denne variabelen. Vi bruker imidlertid ikke disse kodene, og koder om disse nye kodene til de kodene som er gjengitt over
43-46	x	x	Elevers påbegynningsstidspunkt (MMYY)
47-50		x	Elevers avslutningsstidspunkt (MMYY)
55-56	x	x	Skoleslag. Brukes til å kode universitetsstudenter, og skal ikke i basen
70-71	x	x	Studieretningskode. Variabelen viser om kurset som eleven går på er et kurs under lov om videregående opplæring. Her ligger det opplysninger om hvilken studieretning (linje) kurset tilhører
72-73	x	x	Fagprøve og Tors-kode. I første siffer ligger det en F for de som er igang med eller har avsluttet en fagprøve (lærlinger). Denne opplysningen skal inn i databasen. For alle studenter på nivå 5 og over (høyskoler og universitet), er «Tors kode» lagt inn i første og andre siffer. Denne gir en gruppering av disse utdanningene.
74	x	x	Kurskode. I femte siffer i studieretningskoden ligger opplysning om «kursstype», og er registrert for de som er under lov om videregående opplæring.
94-99	x	x	NACE-kode (f.o.m. 1993-fila). Etter 1993 brukes variabelen til å kode AMO-kurs og universitetsstudenter, men denne variabelen skal ikke inn i basen

Vi er interessert i å finne den perioden en person er under utdanning. Vi tar med både elevens og kursets start- og stoppdato, men vil bare benytte kursets påbegynnings-/avslutningsstidspunkt dersom elevens datoer ligger med blank. Det vil si at det er bare én start- og én stoppdato for hver utdanning som blir tatt med videre.

Variabelen skoleslag angir «skoletype», og tar utgangspunkt i skolen, ikke hvilke kurs studentene leser. Variabelen har to sifre, og dette tilsvarer de to siste sifrene i den fem-sifrede koden i Standard for næringsgruppering (for undervisningssektoren) som gjaldt til og med 1992 (ISIC). Variabelen beholder verdiene også etter 1992, selv om næringsgrupperingen erstattes av NACE fra og med 1994. Fra og med 1996 har vi gått over til å benytte NACE-kode=80.301 for å finne studenter ved universitet. Inndelingen i NACE er grovere, og litt annerledes enn i ISIC. Denne variabelen skal brukes til å finne hvem som studerer ved universitet. Det har betydning både for å kode hovedgruppevariabelen, og for å finne de som skal bytte ut utdanningens art med opplysninger fra høstsemesterfila.

Høstsemesterfila

For 1991-1996 viser **utdanningens art** på igangværendefilene studiemål. Vi er blitt anbefalt av fagseksjonen å benytte fag i semesteret i stedet for studiemål for universitetsstudentene. For studenter ved høyskoler er imidlertid studiemål og fag i semesteret det samme. **Fag i semesteret** finnes på semesterfilene. Fra 1997 er det **fag i semesteret** som er registrert som **utdanningens art** på igangværendefilene.

Vi tar med følgende variable fra høstsemesterfilene:

Posisjon	Tekst
1-6	Utdanningens art. Se igangværende.
31-41	Fødselsnummer. Også ugyldige fødselsnumre
55-56	Skolegruppe. Denne variabelen tilsvareer skoleslag i registrene over
65-70	Fag i semesteret.

På høstsemesterfila er studenter både ved høyskoler og universitet registrert. For 1991-1995 er høstsemesterfila en «bruttofil». Det vil si at en person kan være registrert igangværende både med en universitetsutdanning og en høyskoleutdanning. Det betyr også at en person kan være registrert som igangværende med en universitetsutdanning på denne fila, men ikke på den tilsvarende igangværendefila. Det er imidlertid ingen som er registrert med to (eller flere) universitetsutdanninger samtidig.

Vårsemesterfila

Vi trenger vårsemesterfila til å kontrollere hvilke universitetsstudenter som er registrert i vårsemesteret. Vi benytter følgende variable:

Posisjon	Tekst
1-11	Fødselsnummer
12-14	Årgang - semester YYK
30-35	Fag i semesteret

Variabelen **årgang** må vi ha med for å selektere ut de personene som studerer i vårsemesteret (YY1). Det ligger noen personer på filene som er registrert i høstsemesteret før.

5.2 Korreksjoner og kontroller

5.2.1 Fødselsnummerkontroll

Vi skal kjøre de samme fødselsnummerkontrollene som skal kjøres for resten av FD-Trygd (se Q:\dok\fdtrygd\info\plandok>manual\kbs19961210n1.doc).

5.2.2 Dublettkontroll

På igangværendefilene skal det ikke være dubletter i fødselsnummer. Dubletter her defineres derfor ved fødselsnummer alene.

Registrene for avslutta utdanning er brutto-filer, det vil si at det ligger dubletter i fødselsnummer. Det skal det gjøre, siden en person kan avslutte flere utdanninger i løpet av ett år. Dubletter på avslutta filene defineres ved variablene Fødselsnummer+utdanningens art+startdato+stopppdato.

Vi skal ikke ha to like kurs (samme NUS) løpende samtidig, og slår sammen like kurs som er overlappende i tid. Det vil si at vi tar startdato fra det kurset som har startet først, og stopppdato fra det kurset som er avsluttet sist.

Sorteringen som er brukt for å kontrollere records mot hverandre er: fødselsnummer x utdanningens art x startdato x stopppdato (FNR x NUS x STARTD x STOPPD). Dette kan f.eks. se slik ut, med tre records hvor start- og endepunktene på linjene angir startdato og stopppdato:

```
record 1 |-----|
record 2   |-----|
record 3     |-----|
```

Vi tar startdatoen fra record 1, stoppdatoen fra record 3 og internvariablene fra den recorden med utfall = 8 (fullført). Det betyr at hvis kun record 2 har utfall = 8, så hentes alle internvariablene fra denne recorden. Hvis både record 2 og record 3 har utfall = 8, så hentes alle internvariablene fra record 3.

Vi merker hvilke record vi henter opplysninger fra ved slik sammenkobling i **mrk_dl3** etter følgende regler:

- Hvis** Start- og stoppdato (dvs alle variable) er hentet fra samme record **Så**
MRK_DL3 = 0
- Hvis** Startdato, men ikke stoppdato, hentes fra den recorden hvor de andre opplysningene hentes fra **Så**
MRK_DL3 = 1
- Hvis** Stoppdato, men ikke startdato, hentes fra den recorden hvor de andre opplysningene hentes fra **Så**
MRK_DL3 = 2
- Hvis** Start- og stoppdato hentes fra en record mens de andre opplysningene hentes fra en annen record **Så**
MRK_DL3 = 3
- Hvis** Startdato, stoppdato og de andre opplysningene hentes fra tre ulike records **Så**
MRK_DL3 = 4

Ved sammenslåing av kurs til en periode, blir mrk_dl1 og mrk_dl2 oppdatert med riktige koder i forhold til den nye perioden.

5.2.3 Datokontroll

Datoene som ligger på registrene er månedsopplysninger, dermed får vi ingen «ugyldige» datoer, men vi sjekker gyldige kombinasjoner av år og måned.

Datoer i utdanningsfilene er på formen MMY. Disse kodes om til formen YYYYMM.

Vi skal sitte igjen med bare én start- og én stoppdato pr. record for igangværende og avslutta filene. Vi velger på følgende måte:

- Hvis** elevens påbegynnings- (avslutnings-) tidspunkt er utfyllt **Så**
start- (slutt-) tidspunkt settes lik elevens påbegynnings- (avslutnings-) tidspunkt
- Ellers**
start- (slutt-) tidspunkt settes lik kursets påbegynnings- (avslutnings-) tidspunkt

Vi legger så inn en kode som viser hvilke datoer vi har brukt. På igangværendefilene får vi da følgende koder for variabelen **mrk_dl**:

- 1 = Elevens påbegynnings- og avslutnings- tidspunkt er brukt
- 3 = Kursets påbegynnings- og avslutnings- tidspunkt er brukt
- 5 = Uoppgitt påbegynnings- og avslutnings- tidspunkt er satt til 'YY08'

På avsluttafilene får vi følgende koder for variabelen **mrk_dl1**:

- 1 = Elevens påbegynnings- og avslutnings- tidspunkt er brukt
- 2 = Elevens påbegynnings- og kursets avslutnings- tidspunkt er brukt
- 3 = Kursets påbegynnings- og elevens avslutnings- tidspunkt er brukt
- 4 = Kursets påbegynnings- og avslutnings- tidspunkt er brukt
- 5 = Påbegynnings- tidspunkt mangler (helt), mens elevens avslutnings- tidspunkt er brukt
- 6 = Påbegynnings- tidspunkt mangler (helt), mens kursets avslutnings- tidspunkt er brukt (universitetsstudenter)
- 7 = Påbegynnings- tidspunkt er uoppgitt, mens elevens avslutnings- tidspunkt er brukt
- 8 = Påbegynnings- tidspunkt er uoppgitt, mens kursets avslutnings- tidspunkt er brukt
- 9 = Påbegynnings- tidspunkt er større eller lik stoppdato, mens elevens avslutnings- tidspunkt er brukt
- A = Påbegynnings- tidspunkt er større eller lik stoppdato, mens kursets avslutnings- tidspunkt er brukt
- B = Eventuelle andre kombinasjoner (restgruppe)

Alle som studerer ved universitet, er registrert med elevens påbegynnings- og avslutningstidspunkt. Dette er imidlertid datoer som er satt ved fagseksjonen, og sier ikke noe om faktisk dato for når studenten startet eller avsluttet, men viser semesterstart og -stopp. Derfor koder vi datoene for disse studentene til kursets påbegynnings- og avslutningstidspunkt.

På igangværendefilene skal vi i tillegg til å sjekke manglende datoer og gyldigheten av år og måned, også sjekke inkonsistente datoer. Her skal det ikke være noen med påbegynningstidspunkt etter 1/10 i det året fila gjelder for. På igangværendefilene er det ingen som mangler både elevens og kursets påbegynningstidspunkt.

Mangler påbegynningstidspunkt:

Hvis påbegynningstidspunkt er '0101' eller '0000' **Så**

Påbegynningstidspunkt settes til 'YY08' med:

MRK_DL = 5

Logisk inkonsistente datoer:

Hvis påbegynningstidspunkt er etter det fila gjelder for (påbegynningstidspunkt > YY10) **Så**

Recorden slettes

Det er ingen på avslutta filene som mangler både elevens og kursets avslutningstidspunkt. Vi må imidlertid sjekke gyldigheten av år og måneder. I tillegg sjekkes at avslutningstidspunkt ikke er utenom filas "gyldighetsområde" (hvis noen har slike ugyldige avslutningstidspunkter, slettes disse recordene).

Hvis påbegynningstidspunktet mangler, må vi sette ett påbegynningstidspunkt ved danning av miniregistrene. På avsluttafilene mangler alle ved universitet påbegynningstidspunkt. Disse datoene skal settes ved kontroll mot igangværende- og vårsemesterfiler. Ved uoppgitte datoer eller hvis påbegynningstidspunkt er større eller lik avslutningstidspunkt, settes påbegynningstidspunkt ved sjekk mot igangværendefila . Dette gjøres etter følgende regler og merkes i variabelen **mrk_dl2**:

Hvis påbegynningstidspunktet er satt (ikke NULL) **Så**

MRK_DL2 = 0

Hvis HOVED = 4 **Så**

For universitetsstudenter må vi legge inn startdato ved å koble mot igangværende fil året før, og vårsemesterfila samme året:

Hvis match mot igangværende med fødselnummer + utdanningens art som nøkkel **Så**

påbegynningstidspunktet hentes fra igangværendefila

MRK_DL2 = 1

Hvis match mot vårsemesterfila med fødselnummer + utdanningens art som nøkkel **Så**

Påbegynningstidspunktet settes til start i vårsemesteret (YY01) med:

MRK_DL2 = 2

Hvis match mot igangværende med FNR som nøkkel **Så**

(samme person er registrert igang med et annet kurs på igangværendefila)

påbegynningstidspunktet settes til YY.09 med:

MRK_DL2 = 3

Ellers

påbegynningstidspunktet settes til start i det semesteret kurset er avsluttet med:

MRK_DL2 = 4

Hvis HOVED < 4 Så

Hvis HOVED < 1 Og personen er over 16 år Så

For andre som har ugyldige påbegynningsstidspunkt, settes påbegynningsstidspunktet ut fra match mot igangværendefila året før:

Hvis match mot igangværende med FNR x NUS som nøkkel Så

påbegynningsstidspunktet hentes fra igangværendefila med:

MRK_DL2 = 7

Hvis match mot igangværende med FNR som nøkkel Så

(samme person er registrert igang med et annet kurs på igangværendefila)

påbegynningsstidspunktet settes til påbegynningsstidspunktet for dette kurset med:

MRK_DL2 = 8

Ellers

påbegynningsstidspunktet settes til YY_108 med:

MRK_DL2 = 9

Ellers *(alle som avslutter ordinær grunnskole)*

påbegynningsstidspunktet settes lik avslutningsstidspunktet med:

MRK_DL2 = A

Avslutningsstidspunktene på avsluttafilene trenger vi ikke gjøre noe med.

5.2.4 Omarbeidelser

Vi gjør en del omkodinger og bearbeidelser av de opprinnelige filene før vi får de miniregistrene som skal brukes til ilegg i basen.

På igangværende- og avsluttafilene må de som studerer ved universitetene behandles for seg. Vi koder derfor en egen **hovedgruppevariabel** (HOVED) som i tillegg til å vise elevens/studentens utdanningsnivå, også skiller mellom de som studerer ved høyskoler og de som studerer ved universitet. Vi får da følgende koder (før 1996 brukte vi skoleslag til å skille mellom universitet og høyskole):

0 = Ingen utdanning (første siffer i utdanningens art = 0)

1 = Grunnskole (første siffer i utdanningens art = 1,2)

2 = Videregående skole (første siffer i utdanningart = 3,4)

3 = Høyskoler (første siffer utdanningens art = 5,6,7,8, og NACE-kode < 80.301)

4 = Universitet (første siffer fag i semesteret = 5,6,7,8, og NACE-kode = 80.301)

5 = Andre (sekkepost)

9 = Uoppgitt (første siffer i utdanningens art = 9)

Noen av kursene på utdanningsnivå 5 tar egentlig en utdanning på videregående nivå. Disse er vi blitt anbefalt av fagseksjonen å legge over i hovedgruppe 2. De kursene det gjelder har utdanningens art=5 og NACE-kode=80.302 eller 80.309.

For å lette uttaket av data fra databasen, kodes det ut fra **utdanningens art** (NUS) fire nye variable som viser de ulike trinnene i utdanningskoden:

NUS1 er første siffer i utdanningskoden

NUS2 er de to første sifrene i utdanningskoden

NUS3 er de tre første sifrene i utdanningskoden

NUS4 er de fire første sifrene i utdanningskoden

Vi lager en egen kode for de som er registrert med **AMO-kurs** (AMO).

For 1992 og 1993:

Hvis de tre første posisjonene i skolenummeret er lik '999' og klassebetegnelsen er '000' **Så**

AMO = 1

Ellers

AMO = 0

Fra 1994:

Hvis NACE-kode=80.422 **Så**

AMO = 1

Ellers

AMO = 0

Ved kjøring på data ser vi at alle kurs som er kodet som AMO også er registrert som voksenopplærings kurs. Dette utgjør mellom 80 og 90 prosent av alle voksenopplæringskurs.

Vi lager tilsvarende en egen kode for **lærlinger** (LAERLING):

Hvis Fagprøve = F **Så**

LAERLING = 1

Ellers

LAERLING = 0

For studenter ved universitetene (hovedgruppe=4) som ligger på igangværende filene, må vi bytte ut **utdanningens art** (som er studiemål), med «fag i semesteret» fra høstsemesterfilene. Hvis fødselsnummer + utdanningens art er likt i en høstsemesterfil og i tilsvarende igangværendefil, legges fag i semesteret (fra høstsemesterfila) inn i stedet for utdanningens art (på igangværendefila).

På avsluttafilene har ikke studenter ved universitetene fått kodeverdi i **voksenopplæring** (VOKSEN). For records med NULL i voksenopplæring setter vi voksenopplæring = 0.

Fra 1995 har vi nye verdier i **utfall** for noen records. Disse kodes om til gamle verdier på følgende måte:

Hvis UTFALL = B, I eller A **Så**

UTFALL = 8

Hvis utfall=M eller S **Så**

UTFALL = 2

5.3 Miniregistre

Miniregistrene er en fil for avsluttet utdanning og en fil for igangværende utdanning, for hver årgang. Disse filene vil ha lik struktur for å lette innlastingen i basen. Det betyr at variable som kommer bare fra den ene type fil vil være blanke på den andre filen.

Igangværende fil

Miniregistre for igangværendefilene blir lagt på:

\$TRYGDFOB/data/utdann/g91/minireg/igangv/g91.ssd04

\$TRYGDFOB/data/utdann/g92/minireg/igangv/g92.ssd04

\$TRYGDFOB/data/utdann/g93/minireg/igangv/g93.ssd04

\$TRYGDFOB/data/utdann/g94/minireg/igangv/g94.ssd04

\$TRYGDFOB/data/utdann/g95/minireg/igangv/g95.ssd04

\$TRYGDFOB/data/utdann/g96/minireg/igangv/g96.ssd04

\$TRYGDFOB/data/utdann/g97/minireg/igangv/g97.ssd04

\$TRYGDFOB/data/utdann/g98/minireg/igangv/g98.ssd04

Miniregistrene blir også lagret som tekstfiler for ilegg i basen. Disse har filidentene:

\$TRYGDFOB/data/utdann/g91/minireg/igangv/g91.txt
\$TRYGDFOB/data/utdann/g92/minireg/igangv/g92.txt
\$TRYGDFOB/data/utdann/g93/minireg/igangv/g93.txt
\$TRYGDFOB/data/utdann/g94/minireg/igangv/g94.txt
\$TRYGDFOB/data/utdann/g95/minireg/igangv/g95.txt
\$TRYGDFOB/data/utdann/g96/minireg/igangv/g96.txt
\$TRYGDFOB/data/utdann/g97/minireg/igangv/g97.txt
\$TRYGDFOB/data/utdann/g98/minireg/igangv/g98.txt

Filbeskrivelse for tekstfilene er:

Antall tegn	Type	Kortnavn	Variabel
11	char	fnr	Fødselsnummer
1	char	hoved	Hovedgruppevariabel
6	char	nus	Utdanningens art
1	char	nus1	Utdanningens art, 1. siffer
2	char	nus2	Utdanningens art, 2 første siffer
3	char	nus3	Utdanningens art, 3 første siffer
4	char	nus4	Utdanningens art, 4 første siffer
6	num	startd	Påbegynningsstidspunkt (YYYYMM)
6			Blank
2	num	kltrinn	Klasstrinn
1	char	heldel	Heltid/deltid
1	char	voksen	Voksenopplæring
2	char	studret	Studieretningskode
1	char	laerling	Lærling
1	char	amo	AMO-kurs
1			Blank
1	char	mrk_dl	Datomerking
1			Blank
1			Blank
1			Blank
1	char	mrk_fnr	Fødselsnummerkontroll-merking
4	char	"YYYY"	Årgangsopplysning

Internvariable i dette registeret er: Utdanningens art (NUS, NUS1, NUS2, NUS3 og NUS4) Klasstrinn (KLTRINN), Heltid/deltid (HELDEL), Voksenopplæring (VOKSEN), Studieretningskode (STUDRET), Lærling (LAERLING) og AMO-kurs (AMO).

Avslutta fil

Miniregistrene for avslutta utdanning legges på filene:

\$TRYGDFOB/data/utdann/g92/minireg/avslutt/g92.ssd04
\$TRYGDFOB/data/utdann/g93/minireg/avslutt/g93.ssd04
\$TRYGDFOB/data/utdann/g94/minireg/avslutt/g94.ssd04
\$TRYGDFOB/data/utdann/g95/minireg/avslutt/g95.ssd04
\$TRYGDFOB/data/utdann/g96/minireg/avslutt/g96.ssd04
\$TRYGDFOB/data/utdann/g97/minireg/avslutt/g97.ssd04
\$TRYGDFOB/data/utdann/g98/minireg/avslutt/g98.ssd04

Det blir laget tekstfiler av avsluttafilene som skal brukes til ilegg i basen. Disse blir lagt på:

```
$TRYGDFOB/data/utdann/g92/minireg/avslutt/g92.txt  
$TRYGDFOB/data/utdann/g93/minireg/avslutt/g93.txt  
$TRYGDFOB/data/utdann/g94/minireg/avslutt/g94.txt  
$TRYGDFOB/data/utdann/g95/minireg/avslutt/g95.txt  
$TRYGDFOB/data/utdann/g96/minireg/avslutt/g96.txt  
$TRYGDFOB/data/utdann/g97/minireg/avslutt/g97.txt  
$TRYGDFOB/data/utdann/g98/minireg/avslutt/g98.txt
```

Filbeskrivelse for tekstfilene er:

Antall tegn	Type	Kortnavn	Variabel
11	char	fnr	Fødselsnummer
1	char	hoved	Hovedgruppevariabel
6	char	nus	Utdanningens art
1	char	nus1	Utdanningens art, 1. siffer
2	char	nus2	Utdanningens art, 2 første siffer
3	char	nus3	Utdanningens art, 3 første siffer
4	char	nus4	Utdanningens art, 4 første siffer
6	num	startd	Påbegynningsstidspunkt (YYYYMM)
6	num	stoppd	Avslutningstidspunkt (YYYYMM)
2	num	kltrinn	Klasstrinn
1	char	heldel	Heltid/deltid
1	char	voksen	Voksenopplæring
2	char	studret	Studieretningskode
1	char	laerling	Lærling
1	char	amo	AMO-kurs
1	char	utfall	Utfall
1			Blank
1	char	mrk_dl1	Hvilken dato som er brukt fra grunnlagsregisteret
1	char	mrk_dl2	Datomerking ved endring av dato
1	char	mrk_dl3	Hvilken record opplysningene kommer fra ved dublettkontroll
1	char	mrk_fnr	Fødselsnummerkontroll-merking
4	char	"YYYY"	Årgangsopplysning

Internvariable i dette registeret er: Utdanningens art (NUS, NUS1, NUS2, NUS3 og NUS4) Klasstrinn (KLTRINN), Heltid/deltid (HELDEL), Voksenopplæring (VOKSEN), Studieretningskode (STUDRET), Lærling (LAERLING) og AMO-kurs (AMO).

5.4 Sammenkobling ved ilegg i basen

Det er personers utdanningsaktivitet vi er opptatt av å få registrert i FD-Trygd. Ut fra miniregistrene for igangværende og avslutta filene, skal vi konstruere de utdanningsforløpene en person har.

5.4.1 Definisjoner:

I databasen registrerer vi utdanningsaktivitet på tre nivåer (mot to nivåer for andre områder i FD-Trygd). Disse blir:

- *Kurs* er enkeltutdanning for en person som vi får fra utdanningens art. Det vil si at så lenge en person er registrert med samme NUS (utdanningens art), vil vi si at personen holder på med samme kurs. (Dette nivået har vi ikke for andre områder)
- *Utdanning* som er ett eller flere sammenhengende kurs som til sammen utgjør en hovedgruppe slik vi har kodet hovedgruppe i avsnitt 5.2.4. (Dette tilsvarer hovedgrupper i andre områder)

- *Utdanningstilfelle* er en sammenhengende periode en person er under utdanning, uavhengig av hvilke kurs som måtte være registrert over tid (tilsvarer tilfelle i andre områder). Med en sammenhengende periode mener vi at det ikke skal være opphold som er lengre enn én måned, bortsett fra sommermånedene. Altså hvis en person avslutter et kurs en måned og starter opp med et nytt neste måned eller avslutter et kurs i juni og starter opp igjen i august eller september, regner vi det som en sammenhengende periode.

For å få til denne oppbyggingen i databasen, har vi vært nødt til å lage to forløpstabeller:

- F_UTD_KURS, heretter kalt Forløpstabell-Kurs
- F_UTD_VIKTIGSTE, heretter kalt Forløpstabell-Viktigste kurs

Hvordan disse tabellene blir bygget opp, og hva de inneholder, kommer vi tilbake til nedenfor.

For å få lagt data inn i basen, har vi vært nødt til å sette sammen de ulike filene til årganger. Ved ilegg i Forløpstabell-Kurs vil først avslutta fil år t bli benyttet, så vil igangværende fil år t bli lagt inn.

(Unntaket er første årgang som vil bestå av igangværende fil 1991, avslutta fil 1992 samt igangværende fil 1992). Dette vil da utgjøre en årgang. Etter å ha lagt inn en årgang i Forløpstabell-Kurs, benytter vi denne forløpstabellen til å legge inn data i Forløpstabell-Viktigste kurs.

5.4.2 Venstresensur

FD-Trygd starter 1/1-1992. For kurs som er løpende pr. 199201, tar vi med den startdatoen som er registrert. Alle kurs som er avsluttet før 199201 fjernes fra databasen. Vi skal ikke ha endringer i basen før 199202. Hvis det har vært endringer i kurs eller utdanning som skulle vært registrert før denne tid, så legges disse opplysningene inn i tilgangsrecorden.

5.4.3 Frigivning av data

Vi ønsker å kunne frigi data i databasen etter hvert som data blir lagt inn. Vi må imidlertid legge inn en årgang mer enn vi kan frigi. For data fra Forløpstabell-kurs skyldes dette at vi avslutter kurs som ikke kommer igjen på senere filer (verken igangværende- eller avsluttafiler). Disse blir avsluttet året etter at vi sist hadde noen registrering på kurset. For data fra Forløpstabell-Viktigste kurs derimot syr vi sammen kurs og utdanning som følger etter hverandre. En midlertidig avgang kan senere bli gjort om til en endringsrecord, hvis det kommer et kurs som løper videre. Det betyr at for å kunne frigi en årgang, må vi ha lagt inn årgangen etter. Altså må vi legge inn en årgang mer i databasen enn vi kan frigi.

5.5 Forløpstabell-Kurs

I denne tabellen registrerer vi alle kurs en person har vært registrert med, enten på en avslutta fil, eller på en igangværende fil. Igangværende filene er netto-filer, som vil si at det ikke er registrert parallelle kurs på en igangværende fil. På miniregistrene for avslutta utdanning, kan vi derimot ha flere like kurs, men like kurs kan ikke løpe samtidig. Når vi legger miniregistrene inn i Forløpstabell-Kurs, godtar vi at vi får parallelle kurs, men ett og samme kurs (lik NUS og samme HOVED) skal ikke overlappes i tid. Ved overlapp på kursnivå, setter vi opplysningene sammen, og lager endringsrecord hvis noen internvariable har endret verdi.

Dette betyr at enheten i denne forløpstabellen er person x kurs x hovedgruppe (FNR x NUS x HOVED). Vi registrerer ikke overganger mellom ulike kurs, og vi syr heller ikke sammen like kurs som følger etter hverandre. Hvert kurs består av tilgang, eventuelt endring, og avgang. En person kan ikke ha samme kurs overlappende i tid, men ulike kurs for samme person kan imidlertid være overlappende.

Ved ilegg i Forløpstabell-Kurs, benytter vi flere tabeller (se 7.10). Det er først og fremst hovedtabellen, hvor de siste opplysningene vi har om et kurs ligger. Denne benyttes for å se om det er

registrert noe på et tilsvarende kurs, samt om det har vært endringer i noen av internvariablene for kurset. I tillegg benytter vi en avgangstabell, for å se om kurset har vært avsluttet tidligere, og i tilfelle når.

5.5.1 Noen hovedprinsipper når vi bygger opp forløp på kursnivå:

Vi har prøvd å holde oss til noen hovedprinsipper når vi bygger opp forløp på kursnivå:

Vi skal stole på registeropplysningene.

- Det innebærer at hvis en person ikke er registrert som igangværende med et kurs på igangværende fil for år t, så skal ikke opplysninger fra miniregistre fra år t+1 (eller senere) ha startdatoer som går lenger tilbake enn november år t
- Hvis en person er registrert på igangværende fila for år t, men ikke på igangværende fil år t+1, så regner vi kurset som avbrutt, og vi setter avgang for kurset. Slike avbrudd får kursavgang lik juni år t+1 og ufall=1

Endringer i internvariable blir satt til

- august eller september (avhengig av hovedgruppe) år t hvis opplysningene kommer fra en igangværende fil
- november år t-1 hvis opplysningene kommer fra en avslutta fil.

5.5.2 Dato- og hendelseskontroll (pr. FNR x NUS x HOVED)

I beskrivelsen nedenfor benyttes begrepene aktiv bestand og inaktiv bestand. Aktiv bestand (hovedtabell) er alle tilfeller av FNR x NUS x HOVED for personer som har et løpende tilfelle (kurs). Inaktiv bestand (avgangstabellen) er siste tilfeller av FNR x NUS x HOVED som er avsluttet. Tilgangs- og endringsrecords som blir generert blir lagt i hovedtabellen (og i forløpstabellen). Avgangsrecord som blir generert blir lagt i avgangstabellen (og i forløpstabellen).

Variablene som er benyttet nedenfor er nærmere beskrevet i kap. 7.11.

Videre benyttes benevnelsene "ny" (VARIABELNAVNny), "gml" (VARIABELNAVNgml) og "avsl" (VARIABELNAVNavsl) i tilknytning til de ulike variablene:

- VARIABELNAVNny er en variabel fra den record som behandles i øyeblikket fra lastetabellen (miniregisteret).
- VARIABELNAVNgml er en variabel fra den record i hovedtabell (aktiv bestand) som matcher gjeldene record i lastetabellen mht. FNR x NUS x HOVED.
- VARIABELNAVNavsl er en variabel fra den record i avgangstabell (inaktiv bestand) som matcher gjeldene record i lastetabellen mht. FNR x NUS x HOVED.

Vi benytter en hjelpevariabel (STATUS2) for å kunne teste innleggingen og hvilke regler som lå til grunn ved danning av forløpet. STATUS2 initieres til **X99** ved innlesning av hver nye record, men denne verdien SKAL endres vha. testene nedenfor.

A. Hvis ikke match på FNR x NUS x HOVED i aktiv eller inaktiv bestand **Så**

1. Hvis STOPPDny er blank (NULL) **Så** (record kommer fra igangværende fila)

STATUS1 = 1

Hvis STARTDny > AARGANGny - 1år || '10' **Så**

Koder nytt kurs med:

STATUS2 = **A10**,

REGDATO = KURSTILG = STARTDny

Ellers (burde finnes på igangværende-fila for forrige år)

Hvis AARGANGny <> 1991 **Så**

Koder nytt kurs med:

STATUS2 = **A11**,

REGDATO = KURSTILG = AARGANGny - 1 || '11'

Ellers (*venstresensur*)
Koder nytt kurs med:
STATUS2 = **A12**,
REGDATO = KURSTILG = STARTDny

2. **Ellers** (*record fra avsluttafila*)
STATUS1 = 2
Hvis STARTDny > AARGANGny - 1år || '10' **Så**
Koder nytt kurs med:
STATUS2 = **B10**,
REGDATO = KURSTILG = STARTDny
Koder avgang fra kurs med:
STATUS2 = **B11**,
REGDATO = KURSAVG = STOPPDny
Ellers (*burde finnes på igangværende-fila for forrige år*)
Koder nytt kurs med:
STATUS2 = **B12**,
REGDATO = KURSTILG = AARGANGny - 1år || '11'
Koder avgang fra kurs med:
STATUS2 = **B13**,
REGDATO = KURSAVG = MAX(STOPPDny, REGDATO (kodet over))

B. **Hvis** ikke match på FNR x NUS x HOVED i aktiv bestand, men i inaktiv bestand **Så**

1. **Hvis** STOPPDny er blank (NULL) **Så** (*record fra igangværende fila*)
STATUS1 = 5
Hvis STARTDny > KURSAVGavsl **Så**
Koder nytt kurs med:
STATUS2 = **A14**,
REGDATO = KURSTILG = MAX(STARTDny, AARGANGny - 1år || '11')

Ellers (*flytter tilgangsdato til etter avgang for gammelt kurs*)
Hvis AARGANGny = AARGANGavsl **Så**
Hvis KURSAVGavsl = YYYY06 **Så** (YYYY = årstall)
Hvis HOVEDny = 4 **Så** (*universitet*)
Koder nytt kurs med:
STATUS2 = **A15**,
REGDATO = KURSTILG = AARGANGny || '09'

Ellers
Koder nytt kurs med:
STATUS2 = **A16**,
REGDATO = KURSTILG = AARGANGny || '08'

Ellers
Koder nytt kurs med:
STATUS2 = **A17**,
REGDATO = KURSTILG = KURSAVGavsl + 1mnd.

Ellers
Koder nytt kurs med:
STATUS2 = **A40**,
REGDATO = KURSTILG = AARGANGny - 1år || '11'

2. **Ellers** (*record fra avsluttafila*)
STATUS1 = 6
Hvis STARTDny > KURSAVGavsl **Så**
Koder nytt kurs med:
STATUS2 = **B18**,
REGDATO = KURSTILG = MAX(STARTDny, AARGANGny - 1år || '11')

Koder avgang fra kurs med:
STATUS2 = **B19**,
REGDATO = KURSAVG = MAX(REGDATO (kodet over), STOPPDny)

Ellers (*flytter tilgangsdato til etter avgang for gammelt kurs*)
Hvis AARGANGny = AARGANGavsl **Så**

Hvis KURSAVGavsl = YYYY06 **Så** (*YYYY = årstall*)

Hvis HOVEDny = 4 **Så**

Koder nytt kurs med:

STATUS2 = **B20**,

REGDATO = KURSTILG = AARGANGny || '09'

Koder avgang fra kurs med:

STATUS2 = **B21**,

REGDATO = MAX(STOPPDny, REGDATO (kodet over))

Ellers

Koder nytt kurs med:

STATUS2 = **B22**,

REGDATO = KURSTILG = AARGANGny || '08'

Koder avgang fra kurs med:

STATUS2 = **B23**,

REGDATO = MAX(STOPPDny, REGDATO (kodet over))

Ellers (*KURSAVGavsl <> YYYY06 (YYYY = årstall)*)

Koder nytt kurs med:

STATUS2 = **B24**,

REGDATO = KURSTILG = KURSAVGavsl + 1mnd.

Koder avgang fra kurs med:

STATUS2 = **B25**,

REGDATO = MAX(STOPPDny, REGDATO (kodet over))

Ellers

Koder nytt kurs med:

STATUS2 = **B60**,

REGDATO = KURSTILG = AARGANGny - 1år || '11'

Koder avgang fra kurs med:

STATUS2 = **B61**,

REGDATO = KURSAVG = MAX(STOPPDny, REGDATO (kodet over))

C. **Hvis** match på FNR x NUS x HOVED i aktiv bestand **Så**

1. **Hvis** STOPPDny er blank (NULL) **Så** (*record fra igangværende fila*)

STATUS1 = **4**

a) **Hvis** STARTDny = KURSTILGgml **Så**

Hvis ingen endringer i internvariablene, heller ikke klasstrinn **Så**

(kan være ett 1.årig kurs tatt over to år)

Løpende kurs uten endring;

oppdaterer AARGANG og STATUS2 i hovedtabell:

STATUS2 = **A18**

Ellers

Hvis KLTRINNny = KLTRINNgml + 1 **Så** (*"normal" progresjon*)

Løpende kurs med endring, koder en endringsrecord med:

STATUS2 = **A19**,

Hvis HOVEDny = 4 **Så** (*universitet*)

REGDATO = AARGANGny || '09'

Ellers

REGDATO = AARGANGny || '08'

Hvis KLTRINNny > KLTRINNgml + 1 **Så**

(klasstrinn økt med mer enn 1)

Løpende kurs med endring, koder en endringsrecord med:

STATUS2 = **A20**,

Hvis HOVEDny = 4 **Så** (*universitet*)

REGDATO = AARGANGny || '09'

Ellers

REGDATO = AARGANGny || '08'

Hvis KLTRINNny = KLTRINNgml **Så** (*klasstrinn er uendret*)

Løpende kurs med endring; koder en endringsrecord med:

STATUS2 = **A21**,

- Hvis** HOVEDny = 4 **Så** (*universitet*)
 REGDATO = AARGANGny || '09'
- Ellers**
 REGDATO = AARGANGny || '08'
- Ellers** (*klassetrinn har gått ned*)
Løpende kurs med endring, koder en endringsrecord med:
 STATUS2 = **A22**,
- Hvis** HOVEDny = 4 **Så** (*universitet*)
 REGDATO = AARGANGny || '09'
- Ellers**
 REGDATO = AARGANGny || '08'
- b) **Hvis** STARTDny > KURSTILGgml **Så**
Hvis ingen endringer i internvariablene, heller ikke klassetrinn **Så**
(vi har en sideveis bevegelse i utdanningsystemet)
Avslutter gammelt kurs med:
 STATUS2 = **A23**,
 REGDATO = KURSAVG = AARGANGny || '06'
Starter opp nytt kurs etter avslutning av det gamle med:
 STATUS2 = **A24**,
- Hvis** HOVEDny = 4 **Så** (*universitet*)
 REGDATO = AARGANGny || '09'
- Ellers**
 REGDATO = AARGANGny || '08'
- Ellers** (*endringer i internvariablene*)
Hvis KLTRINNny = KLTRINNgml + 1 **Så** (*"normal" progresjon*)
Løpende kurs med endring, koder en endringsrecord med:
 STATUS2 = **A25**,
- Hvis** HOVEDny = 4 **Så** (*universitet*)
 REGDATO = AARGANGny || '09'
- Ellers**
 REGDATO = AARGANGny || '08'
- Hvis** KLTRINNny > KLTRINNgml + 1 **Så**
(klassetrinn økt med mer enn 1)
Løpende kurs med endring, koder en endringsrecord med:
 STATUS2 = **A26**,
- Hvis** HOVEDny = 4 **Så** (*universitet*)
 REGDATO = AARGANGny || '09'
- Ellers**
 REGDATO = AARGANGny || '08'
- Hvis** KLTRINNny = KLTRINNgml **Så** (*klassetrinn er uendret*)
Løpende kurs med endring; koder en endringsrecord med:
 STATUS2 = **A27**,
- Hvis** HOVEDny = 4 **Så** (*universitet*)
 REGDATO = AARGANGny || '09'
- Ellers**
 REGDATO = AARGANGny || '08'
- Ellers** (*klassetrinn har gått ned*)
Løpende kurs med endring, koder en endringsrecord med:
 STATUS2 = **A28**,
- Hvis** HOVEDny = 4 **Så** (*universitet*)
 REGDATO = AARGANGny || '09'
- Ellers**
 REGDATO = AARGANGny || '08'
- c) **Ellers** (*STARTDny < KURSTILGgml*)
Hvis ingen endringer i internvariablene, heller ikke klassetrinn **Så**
Løpende kurs uten endring;
oppdaterer AARGANG og STATUS2 i hovedtabell:
 STATUS2 = **A29**

Ellers**Hvis** KLTRINNny = KLTRINNgml + 1 **Så** ("normal" progresjon)*Løpende kurs med endring, koder en endringsrecord med:*STATUS2 = **A30**,**Hvis** HOVEDny = 4 **Så** (universitet)

REGDATO = AARGANGny || '09'

Ellers

REGDATO = AARGANGny || '08'

Hvis KLTRINNny > KLTRINNgml + 1 **Så***(klassetrinn økt med mer enn 1)**Løpende kurs med endring, koder en endringsrecord med:*STATUS2 = **A31**,**Hvis** HOVEDny = 4 **Så** (universitet)

REGDATO = AARGANGny || '09'

Ellers

REGDATO = AARGANGny || '08'

Hvis KLTRINNny = KLTRINNgml **Så** (klassetrinn er uendret)*Løpende kurs med endring; koder en endringsrecord med:*STATUS2 = **A32**,**Hvis** HOVEDny = 4 **Så** (universitet)

REGDATO = AARGANGny || '09'

Ellers

REGDATO = AARGANGny || '08'

Ellers (klassetrinn har gått ned)*Løpende kurs med endring, koder en endringsrecord med:*STATUS2 = **A33**,**Hvis** HOVEDny = 4 **Så** (universitet)

REGDATO = AARGANGny || '09'

Ellers

REGDATO = AARGANGny || '08'

2. **Ellers** (record fra avsluttafila)STATUS1 = **3**a) **Hvis** STARTDny = KURSTILGml **Så****Hvis** ingen endringer i internvariablene, heller ikke klassetrinn **Så***Koder avgang fra kurs med:*STATUS2 = **B26**,

REGDATO = KURSAVG = STOPPDny

Hvis kun endr. i HELDEL **Og** HOVEDgml = 4 **Og** HELDELgml = 2 **Så***(bare deltid ved universitet er endret)**Koder avgang fra kurs med:*STATUS2 = **B27**,

REGDATO = KURSAVG = STOPPDny

Ellers**Hvis** HOVEDgml = 4 **Og** HELDELgml = 2 **Så***(deltid ved universitet endres ikke)***Hvis** AARGANGny - 1år || '11' < 199202 **Så***Kan ikke ha en endring som er før 199202 i basen;**legger endrede internvariable ut den venstresensurerte tilgangsrecorden.*STATUS2 = **B31**,**Ellers***Koder en endringsrecord med:*STATUS2 = **B32**,

REGDATO = AARGANGny - 1år || '11'

*Koder avgang fra kurs med:*STATUS2 = **B33**,

REGDATO = MAX(STOPPDny, REGDATO (kodet over))

Ellers

Hvis AARGANGny - 1år || '11' < 199202 **Så**
*Kan ikke ha en endring som er før 199202 i basen;
legger endrede internvariable ut den venstresensurerte
tilgangsrecorden.*
STATUS2 = **B34**,

Ellers

Koder en endringsrecord med:
STATUS2 = **B35**,
REGDATO = AARGANGny - 1år || '11'
Koder avgang fra kurs med:
STATUS2 = **B36**,
REGDATO = MAX(STOPPDny, REGDATO (kodet over))

b) **Ellers** (STARTDny <> KURSTILGgml)

Hvis STARTDny <= AARGANGny - 1år || '10' **Så**

Hvis ingen endringer i internvariablene **Så**

Koder avgang fra kurs med:
STATUS2 = **B37**,
REGDATO = KURSAVG = STOPPDny

Hvis kun endring i HELDEL **Og** HOVEDgml = 4 **Og**

HELDELgml=2 **Så** (*deltid ved universitet endres ikke*)

Koder avgang fra kurs med:
STATUS2 = **B38**,
REGDATO = KURSAVG = STOPPDny

Ellers

Hvis HOVEDgml = 4 **Og** HELDELgml = 2 **Så**

(*deltid ved universitet endres ikke*)

Hvis AARGANGny - 1år || '11' < 199202 **Så**
*Kan ikke ha en endring som er før 199202 i basen;
legger endrede internvariable ut den
venstresensurerte tilgangsrecorden.*
STATUS2 = **B42**,

Ellers

Koder en endringsrecord med:
STATUS2 = **B43**,
REGDATO = AARGANGny - 1år || '11'
Koder avgang fra kurs med:
STATUS2 = **B44**,
REGDATO=MAX(STOPPDny, REGDATO (kodet over))

Ellers

Hvis AARGANGny - 1år || '11' < 199202 **Så**

*Kan ikke ha en endring som er før 199202 i basen; legger
endrede internvariable ut den venstresensurerte
tilgangsrecorden.*
STATUS2 = **B45**,

Ellers

Koder en endringsrecord med:
STATUS2 = **B46**,
REGDATO = AARGANGny - 1år || '11'
Koder avgang fra kurs med:
STATUS2 = **B47**,
REGDATO = MAX(STOPPDny, REGDATO (kodet over))

Ellers (STARTDny > 19YY10, hvor YY er avsluttafilas årgang - 1år)

Koder avgang fra gml.kurs med:
STATUS2 = **B48**,
REGDATO = KURSAVG = AARGANGny - 1år || '10'
Koder tilgang til nytt kurs med:
STATUS2 = **B49**,
REGDATO = KURSTILG = STARTDny

Koder avgang fra kurs med:
STATUS2 = **B50**,
REGDATO = KURSAVG = STOPPDny

D. **Ellers** (*skal IKKE kunne oppstå*)
STATUS2 = **Z99**

5.5.3 Avgangsrutine

Hvis ingen ny record for løpende tilfelle **Og** gap på mer enn ett år **Og** REGDATgml >= 199201 **Så**

Koder avgang fra kurs med:
REGDATO = AVGDATO = Årgang som akkurat er lest inn || '06',
UTFALL = '1'

5.6 Forløpstabell-Viktigste kurs

Etter å ha lagt alle kurs for en årgang inn i Forløpstabell-Kurs, bruker vi denne forløpstabellen som load-tabell for ilegg i Forløpstabell-Viktigste kurs.

Forløpstabell-Viktigste kurs er dannet for å lette uttaket av igangværende utdanning fra databasen. I denne tabellen er person enhet, og det ligger bare en utdanning løpende av gangen. Like kurs sys isammen, det samme gjøres med ulike kurs innen samme hovedgruppe (utdanning). Og ulike hovedgrupper sys sammen til (utdannings)tilfeller. Her innføres altså samme struktur på forløpstabellen som i resten av FD-Trygd, med sammenhengende utdanningstilfeller, overgang mellom hovedgrupper og endringsrecords når vi har endringer i internvariable. Det er viktig å merke seg at utdanningens art (NUS) som er en del av "nøkkelen" i Forløpstabell-Kurs er blitt en internvariabel i Forløpstabell-Viktigste kurs.

Om vi her bare får registrert en liten del av en kursperiode, har vi tatt med datoer for når kurset egentlig startet og stoppet. Vi har dermed fått med oss hele perioden for alle kurs som har vært registrert som viktigste kurs i en eller annen periode. Kurs som derimot ikke blir registrert som viktigst i noen periode, vil det ikke ligge noe opplysning om i denne tabellen. Skal man f.eks. finne alle som har bestått en bestemt eksamen, må dette plukkes ut fra Forløpstabell-Kurs.

Vi har også her benyttet både en Hovedtabell og en Avgangstabell for å få lagt data inn i Forløpstabell-Viktigste kurs (se kap. 7.10).

5.6.1 Noen hovedprinsipper når vi bygger opp forløp på viktigste kursnivå:

Vi har prøvd å holde oss til noen hovedprinsipper når vi bygger opp forløp på viktigste kursnivå:

- Vi bruker de recordene som ligger i Forløpstabell-Kurs som input. Disse recordene er enten tilgangs-, endrings- eller avgangsrecord, og har fått kodet en REGDATO
- Vi tar ikke med datoer lenger tilbake enn utgangen av forrige årgangs utløp. Altså når vi f.eks. legger inn 1995-årgangen, skal ingen kurs herfra starte opp før november 1994.
- Skal ikke ha tilgang og endring på samme dag (utdanning innen samme hovedgruppe). Kan imidlertid ha endring og avgang på samme dag, eller tilgang og avgang

5.6.2 Betingelser ved ilegg i Forløpstabell-Viktigste kurs

Variabler benyttet nedenfor er nærmere beskrevet i 7.11.

Benevnelsene "ny" (VARIABELNAVNny), "gml" (VARIABELNAVNgml) og "avsl" (VARIABELNAVNavsl) benyttes i tilknytning til de ulike variablene:

- VARIABELNAVNny er en variabel fra den record som behandles i øyeblikket fra lastetabellen (F_UTD_KURS).

- VARIABELNAVNgml er en variabel fra den record i hovedtabell (aktiv bestand) som matcher gjeldene record i lastetabellen mht. FNR.
- VARIABELNAVNavsl er en variabel fra den record i avgangstabell (inaktiv bestand) som matcher gjeldene record i lastetabellen mht. FNR.

Tilgangs-, endrings- og midlertidige avgangsrecords som blir generert blir lagt i hovedtabellen (og i forløpstabellen). Avgangsrecords som genereres som følge av avgansrutinen, blir lagt i avgangstabellen (og i forløpstabellen).

5.6.3 Sortering

Vi har prøvd å følge samme kriterier som fagseksjonen bruker når de plukker ut de kursene som er registrert i igangværende filene. Vi har imidlertid vært nødt til å ta med noen kriterier i tillegg for å få en entydig sortering. Records fra Forløpstabell-Kurs benyttes for å danne forløp i Forløpstabell-Viktigste kurs. Disse leses inn årgang for årgang (en årgang er definert som avsluttaYY + igangværendeYY) og sorteres etter:

FNR +
 (Hvis KURSTILG = YYYY09 Så KURSTILG = YYYY08, Ellers KURSTILG) +
 (Hvis KODE = 0, Så 9, Hvis KODE = 3 Så 0, Ellers KODE) +
 HELDEL +
 (Hvis UTFALL = NULL Så 8, Ellers UTFALL) (synkende sortering) +
 NUS1 (synkende sortering) +
 KLTRINN (synkende sortering) +
 NUS (synkende sortering) +
 REGDATO +
 (Hvis KURSAVG = NULL Så 0, Ellers KURSAVG) +
 HOVED (synkende sortering)

5.6.4 Vurdering av recordenes "viktighet" (pr. FNR)

Den innleste record matches (på FNR) mot record i hovedtabell. Ved match testes det om ny innlest record (fra F_UTD_KURS) er viktigere enn eksisterende record i hovedtabellen (H_UTD_VIKTIGSTE) etter følgende regler:

- A. Disse IF-testene fanger opp alle nye tilfeller som er viktigere enn gammelt tilfelle, uansett hva gammelt, eller nytt tilfelle måtte være (tilgang, endring eller avgang):

Hvis HELDELny < HELDELgml **Så**
 STATUS9 = 1

Hvis HELDELny = HELDELgml **Så**
Hvis NUS1ny > NUS1gml **Så**
 STATUS9 = 1

Hvis NUS1ny = NUS1gml **Så**
Hvis KLTRINNny > KLTRINNgml **Så**
 STATUS9 = 1

Hvis KLTRINNny = KLTRINNny **Så**
Hvis NUSny > NUSgml **Så**
 STATUS9 = 1

Hvis NUSny = NUSgml
Hvis KODEny er 0 eller 2
Og siste siffer i UTDKODEgml er 1, 2 eller 3
Og HOVEDny = første siffer i UTDKODEgml **Så**
(for de tilfellene der ny innlest record er rangert som like viktig som gammel record, -for samme kurstillfelle, men hvor den nye recorden skal behandles som en viktigere record likevel)
 STATUS9 = 2

Hvis KODEny = 1

Og siste siffer i UTDKODEgml = 0
Og HOVEDny = første siffer i UTDKODEgml Så
STATUS9 = 3

Ellers (*ny innlest record er mindre viktig enn eksisterende record i hovedtabell*)
STATUS9 = 9

- B. Den neste testen er for de tilfellene der ny innlest record ikke er rangert som like viktig eller viktigere enn eksisterende record i hovedtabell, -for samme kurstillfelle, men hvor den nye recorden skal behandles som en viktigere record likevel. Record i hovedtabell er enten en tilgang eller endring, og ny innlest record er en endringsrecord:

Hvis KODEny = 2
Og siste siffer i UTDKODEgml er 1, 2 eller 3
Og NUSny = NUSgml
Og (HELDELny > HELDELgml Eller KLTRINNny < KLTRINNgml) Så
STATUS9 = 4

5.6.5 Dato- og hendelseskontroll (pr. FNR)

I beskrivelsen nedenfor benyttes begrepene aktiv bestand og inaktiv bestand. Aktiv bestand (hovedtabell) er alle tilfeller for personer (pr. FNR) som har et løpende tilfelle. Inaktiv bestand er alle tilfeller for personer (pr. FNR) som er avsluttet det siste året (avgangstabellen).

Vi benytter en hjelpevariabel (STATUS7) for å kunne teste innleggingen og hvilke regler som lå til grunn ved danning av forløpet. STATUS7 initieres til **X99** ved innlesning av hver nye record, men denne verdien SKAL endres vha. testene nedenfor.

- A. **Hvis** ikke match på FNR i aktiv eller inaktiv bestand Så
Hvis KODEny = 3 Så (*venstresensur*)
Koder nytt utdanningstilfelle med:
STATUS7 = **B10**,
REGDATO = TILGDATO = REGDATOny
Hvis KODEny = 1 Så
Hvis REGDATOny > (AARGANGny - 1) || 10 Så
Koder nytt utdanningstilfelle med:
STATUS7 = **B11**,
REGDATO = TILGDATO = REGDATOny
Ellers (*skal IKKE forekomme*)
STATUS7 = **B98**
Ellers (*skal IKKE forekomme*)
STATUS7 = **B99**
- B. **Hvis** ikke match på FNR i aktiv bestand, men i inaktiv bestand Så
Hvis KODEny = 1 eller 3 Så (*tidligere avsluttet tilfelle og TILGANGny*)
Hvis REGDATOny > REGDATOavsl Så
Koder nytt utdanningstilfelle med med:
STATUS7 = **C10**,
REGDATO = TILGDATO = REGDATOny
Ellers (*skal IKKE forekomme*)
STATUS7 = **C97**
- Hvis** KODEny = 2 Så (*tidligere avsluttet tilfelle og ENDRINGny*)
(*skal IKKE forekomme*)
STATUS7 = **C98**
- Ellers** (*tidligere avsluttet tilfelle og AVGANGny*)

(skal IKKE forekomme)

STATUS7 = C99

C. Hvis match på FNR i aktiv bestand Så

1. Hvis record i hovedtabell er en tilgang (andre siffer i UTDKODEgml = 1 eller 3) Så

a) Hvis HOVEDny = andre siffer i UTDKODEgml Så (lik hovedgruppe)

Hvis STATUS9 er 1, 2, 3 eller 4 Så (ny record er viktigere enn gammel record)

Hvis KODENy er 1 eller 3 Så (TILGANGgml, TILGANGny, lik hovedgruppe)

Hvis REGDATOny > REGDATOgml Ellers

(REGDATOny <= REGDATOgml Og REGDATOgml = 199201 Og

UTDKODEgml = 3) Så

Hvis KURSTOMny = 199201 Så

Denne recorden tas ikke med i forløpstabellen, men logges

STATUS7 = D60

Ellers

Koder en endringsrecord med:

STATUS7 = D10,

REGDATO = MAX(REGDATOny, AARGANGny - 1 år || 11, 199202)

Ellers

Hvis REGDATOny <= REGDATOgml Og REGDATOgml > KURSTILGgml Så

Hvis KURSTOMny > REGDATOgml Så

Koder en endringsrecord med:

STATUS7 = D11,

REGDATO = REGDATOgml + 1 md.

Ellers

Denne recorden tas ikke med i forløpstabellen, men logges

STATUS7 = D61

Hvis REGDATny = REGDATOgml Og AARGANGny > AARGANGgml Så

STATUS7 = D90

Ellers (skal IKKE forekomme)

STATUS7 = D98

Hvis KODENy = 2 Så (TILGANGgml, ENDRINGny, lik hovedgruppe)

Hvis REGDATOny > REGDATOgml

Koder en endringsrecord med:

STATUS7 = D12,

REGDATO = MAX(REGDATOny, AARGANGny - 1 år || 11, 199202)

Hvis REGDATOny = REGDATOgml Og AARGANGny > AARGANGgml Så

STATUS7 = D91

Ellers (skal normalt IKKE forekomme)

STATUS7 = D99

Ellers (TILGANGgml, AVGANGny, lik hovedgruppe)

Hvis REGDATOny > REGDATOgml Så

Tester i UTD_FORDEL_VIKTIGSTE_KURS om vi skal kode en endring

STATUS7 = D13,

REGDATO = REGDATOny

Koder en midlertidig avgang med:

STATUS7 = D14,

REGDATO = REGDATOny

Hvis REGDATOny = REGDATOgml Og NUSny = NUSgml Så

Koder en midlertidig avgang med:

STATUS7 = D15,

REGDATO = REGDATOny

Ellers

STATUS7 = D80

Ellers (ny record er IKKE viktigere enn gammel record)

Hvis KODENy er 1 eller 3 Så (TILGANGgml, TILGANGny, lik hovedgruppe)

*Tilfellet er (hittil) ikke viktigst i sin periode og tas (foreløpig)
ikke med i forløpstabellen*

STATUS7 = D70

Hvis KODENy = 2 **Så** (TILGANGgml, ENDRINGny, lik hovedgruppe)

*Tilfellet er (hittil) ikke viktigst i sin periode og tas (foreløpig)
ikke med i forløpstabellen*

STATUS7 = D71

Ellers (TILGANGgml, AVGANGny, lik hovedgruppe)

*Finner ut om kurset er rangert som viktigste kurs i noen perioder i
forløpstabellen (F_UTD_VIKTIGSTE). Dersom dette er tilfelle oppdaterer vi
med opplysninger om UTFALL og KURSAVG for dette tilfellet.*

STATUS7 = D81

b) **Ellers** (ulik hovedgruppe)

Hvis STATUS9 er 1, 2, 3 eller 4 **Så** (ny record er viktigere enn gammel record)

Hvis KODENy er 1 eller 3 **Så** (TILGANGgml, TILGANGny, ulik hovedgruppe)

Hvis REGDATOny > REGDATOgml **Eller**

(REGDATOny <= REGDATOgml **Og** REGDATOgml = 199201 **Og**
UTDKODEgml = 3) **Så**

Hvis KURSTOMny = 199201 **Så**

Denne recorden tas ikke med i forløpstabellen, men logges

STATUS7 = E60

Ellers

(UTFALL og KURSAVG kodes senere ifm. avgangsrecord for kurset)

Koder en avgang fra gammel utdanningsgruppe med:

STATUS7 = E10,

REGDATO = UTDAVG =

MAX(REGDATOny, AARGANGny - 1 år || 10, 199201)

Koder tilgang til ny utdanningsgruppe med:

STATUS7 = E11,

REGDATO = UTDTILG = REGDATO (kodet over) + 1 mnd

Ellers

Hvis REGDATOny <= REGDATOgml **Og** REGDATOgml > KURSTILGgml **Så**

Hvis KURSTOMny > REGDATOgml **Så**

(UTFALL og KURSAVG kodes senere ifm. avgangsrecord for kurset)

Koder en avgang fra gammel utdanningsgruppe med:

STATUS7 = E12,

REGDATO = UTDAVG = REGDATOgml

Koder en tilgang til nyl utdanningsgruppe med:

STATUS7 = E13,

REGDATO = UTDTILG = REGDATO (kodet over) + 1 md.

Ellers

Denne recorden tas ikke med i forløpstabellen, men logges

STATUS7 = E61

Ellers (skal IKKE forekomme)

STATUS7 = E97

Hvis KODENy = 2 **Så** (TILGANGgml, ENDRINGny, ulik hovedgruppe)

Hvis REGDATOny > REGDATOgml **Så** (overgang)

*(UTFALL og KURSAVG kodes senere ifm. avgangsrecord for kurset)
Koder en avgang fra gammel utdanningsgruppe med:*

STATUS7 = E14,

Tester hvilken dato vi skal bruke for gruppeavgangen:

Hvis REGDATOny er YYYY08 eller YYYY09 (YYYY = årstall) **Så**

REGDATO = UTDAVG =

MAX(YYYY fra REGDATOny || 06, REGDATOgml)

Ellers

REGDATO = UTDAVG = REGDATOny - 1mnd.

Koder en tilgang til ny utdanningsgruppe med:

STATUS7 = **E15**,
REGDATO = UTDTILG = REGDATOny
Ellers (skal IKKE være mulig)
STATUS7 = **E98**

Ellers (TILGANGgml, AVGANGny, ulik hovedgruppe)
Hvis REGDATOgml > KURSTILGgml **Så**
Vi tar ikke med denne recorden
STATUS7 = **E90**
Ellers (skal IKKE forekomme)
STATUS7 = **E99**

Ellers (ny record er IKKE viktigere enn gammel record)
Hvis KODEny er 1 eller 3 **Så** (TILGANGgml, TILGANGny, ulik hovedgruppe)
Tilfellet er (hittil) ikke viktigst i sin periode og tas (foreløpig) ikke med i forløpstabelen F_UTD_VIKTIGSTE
STATUS7 = **E70**
Hvis KODEny = 2 **Så** (TILGANGgml, ENDRINGny, ulik hovedgruppe)
Tilfellet er (hittil) ikke viktigst i sin periode og tas (foreløpig) ikke med i forløpstabelen F_UTD_VIKTIGSTE
STATUS7 = **E71**
Ellers (TILGANGgml, AVGANGny, ulik hovedgruppe)
Finner ut om kurset er rangert som viktigste kurs i noen perioder i forløpstabelen (F_UTD_VIKTIGSTE). Dersom dette er tilfelle oppdaterer vi med opplysninger om UTFALL og KURSAVG for dette tilfellet.
STATUS7 = **E80**

2. **Hvis** record i hovedtabell er en endring (andre siffer i UTDKODEgml = 2) **Så**
a) **Hvis** HOVEDny = andre siffer i UTDKODEgml **Så** (lik hovedgruppe)
Hvis STATUS9 er 1, 2, 3 eller 4 **Så** (ny record er viktigere enn gammel record)
Hvis KODEny er 1 eller 3 **Så** (ENDRINGgml, TILGANGny, samme hovedgruppe)
Hvis KURSTOMny <= REGDATOgml **Så**
Koder en endringsrecord med:
STATUS7 = **F10**,
REGDATO = MAX(REGDATOny, REGDATOgml + 1 md.)
Ellers
Denne recorden tas ikke med i forløpstabelen, men logges
STATUS7 = **F60**

Hvis KODEny = 2 **Så** (ENDRINGgml, ENDRINGny, samme hovedgruppe)
Hvis KURSTOMny > REGDATOgml **Så**
Koder en endringsrecord med:
STATUS7 = **F12**,
REGDATO = MAX(REGDATOny, REGDATOgml + 1 mnd.)
Ellers
Denne recorden tas ikke med i forløpstabelen, men logges
STATUS7 = **F61**

Ellers (ENDRINGgml, AVGANGny, samme hovedgruppe)
Hvis REGDATOny > REGDATOgml **Så**
Tester i UTD_FORDEL_VIKTIGSTE_KURS om vi skal kode en endring
STATUS7 = **F13**,
REGDATO = REGDATOny
Koder en midlertidig avgang for utdanningstilfellet med:
STATUS7 = **F14**,
REGDATO = AVGDATO = REGDATOny
Hvis REGDATOny = REGDATOgml **Og** NUSny = NUSgml **Så**
Koder en midlertidig avgang for utdanningstilfellet med:

STATUS7 = **F15**,
REGDATO = AVGDATO = MAX(REGDATO_{ny}, REGDATO_{gml})

Ellers

Finner ut om kurset er rangert som viktigste kurs i noen perioder i forløpstabellen (F_UTD_VIKTIGSTE). Dersom dette er tilfelle oppdaterer vi med opplysninger om UTFALL og KURSAVG for dette tilfellet
STATUS7 = **F80**

Ellers (ny record er IKKE viktigere enn gammel record)

Hvis KOD_{ny} er 1 eller 3 **Så** (ENDRING_{gml}, mindre viktig TILGANG_{ny},)
Tilfellet er (hittil) ikke viktigst i sin periode og tas (foreløpig) ikke med i forløpstabellen F_UTD_VIKTIGSTE
STATUS7 = **F70**

Hvis KOD_{ny} = 2 **Så** (ENDRING_{gml}, mindre viktig ENDRING_{ny}, samme hovedgruppe)
Tilfellet er (hittil) ikke viktigst i sin periode og tas (foreløpig) ikke med i forløpstabellen F_UTD_VIKTIGSTE
STATUS7 = **F71**

Ellers (ENDRING_{gml}, mindre viktig AVGAN_{ny}, samme hovedgruppe)
Finner ut om kurset er rangert som viktigste kurs i noen perioder i forløpstabellen (F_UTD_VIKTIGSTE). Dersom dette er tilfelle oppdaterer vi med opplysninger om UTFALL og KURSAVG for dette tilfellet.
STATUS7 = **F81**

b) **Ellers** (ulik hovedgruppe)

Hvis STATUS9 er 1, 2, 3 eller 4 **Så** (ny record er viktigere enn gammel record)

Hvis KOD_{ny} er 1 eller 3 **Så** (ENDRING_{gml}, TILGANG_{ny}, ulik hovedgruppe)

Hvis KURSTOM_{ny} > REGDATO_{gml} **Så** (overgang)
(UTFALL og KURSAVG kodes senere ifm. avgangsrecord for dette kurset)
Koder en avgang fra gammel utdanningsgruppe med:
STATUS7 = **G12**,

Hvis REGDATO_{ny} er YYYY08 eller YYYY09 (YYYY = årstall) **Så**
REGDATO = UTDAVG = AARGANG_{ny} || 06

Ellers

REGDATO = UTDAVG = MAX(REGDATO_{gml}, REGDATO_{ny} - 1 mnd.)
Koder en tilgang til ny utdanningsgruppe med:

STATUS7 = **G13**,
REGDATO = UTDTILG = MAX(REGDATO_{ny}, UTDAVG_{gml} + 1 mnd.)

Ellers (overgang)

Denne recorden tas ikke med i forløpstabellen, men logges
STATUS7 = **G60**

Hvis KOD_{ny} = 2 **Så** (ENDRING_{gml}, ENDRING_{ny}, ulik hovedgruppe)

Hvis KURSTOM_{ny} > REGDATO_{gml} **Så** (overgang)

(UTFALL og KURSAVG kodes senere ifm. avgangsrecord for dette kurset)
Koder en avgang fra gammel utdanningsgruppe med:
STATUS7 = **G16**,

Tester hvilken dato vi skal bruke for gruppeavgangen:

Hvis REGDATO_{ny} er YYYY08 eller YYYY09 (YYYY = årstall) **Så**
REGDATO = UTDAVG =
MAX/YYYY fra REGDATO_{ny} || 06, REGDATO_{gml})

Ellers

REGDATO = UTDAVG = MAX(REGDATO_{ny} - 1 mnd., REGDATO_{gml})
Koder en tilgang til ny utdanningsgruppe med:

STATUS7 = **G17**,
REGDATO = UTDTILG = MAX(REGDATO_{ny}, UTDAVG_{gml} + 1 mnd.)

Undersøker om det skal kodes en endringsrecord i tillegg
(utd_fordeel_viktigste_kurs)

STATUS7 = **G18**,
REGDATO = REGDATO_{ny}

Ellers

Denne recorden tas ikke med i forløpstabellen, men logges
 STATUS7 = **G61**

Ellers (ENDRINGgml, AVGANGny, ulik hovedgruppe)**Hvis** REGDATOny <= REGDATOgml **Så**

Finner ut om kurset er rangert som viktigste kurs i noen perioder i forløpstabellen (F_UTD_VIKTIGSTE). Dersom dette er tilfelle oppdaterer vi med opplysninger om UTFALL og KURSAVG for dette tilfellet.
 STATUS7 = **G80**

Ellers

Koder avgang fra gammel utdanningsgruppe med:

STATUS7 = **G19**,

REGDATO = UTDAVG = REGDATOgml

Koder en tilgang til ny utdanningsgruppe med:

STATUS7 = **G20**,

Hvis REGDATOgml = YYYY06 **Og** HOVEDny < 4 **Så**

Hvis REGDATOny = YYYY07 **Så**

REGDATO = UTDTILG = REGDATOny

Ellers

REGDATO = UTDTILG = årstall fra UTDAVGgml || 08

Hvis REGDATOgml = YYYY06 **Og** HOVEDny = 4 **Så**

Hvis REGDATOny er YYYY07 eller YYYY08 **Så**

REGDATO = UTDTILG = REGDATOny

Ellers

REGDATO = UTDTILG = årstall fra REGDATOgml || 09

Ellers

REGDATO = UTDTILG = UTDAVGgml + 1mnd.

Koder (midlertidig) avgang for utdanningstilfellet med:

STATUS7 = **G21**,

REGDATO=AVGDATO=MAX(REGDATOny,REGDATO (kodet over))

Ellers (ny record er IKKE viktigere enn gammel record)

Hvis KODEny er 1 eller 3 **Så** (ENDRINGgml, mindre viktig TILGANGny)

Tilfellet er (hittil) ikke viktigst i sin periode og tas (foreløpig) ikke med i forløpstabellen F_UTD_VIKTIGSTE

STATUS7 = **G70**

Hvis KODEny = 2 **Så** (ENDRINGgml, mindre viktig ENDRINGny)

Tilfellet er (hittil) ikke viktigst i sin periode og tas (foreløpig) ikke med i forløpstabellen F_UTD_VIKTIGSTE

STATUS7 = **G71**

Ellers (ENDRINGgml, mindre viktig AVGANGny)

Finner ut om kurset er rangert som viktigste kurs i noen perioder i forløpstabellen (F_UTD_VIKTIGSTE). Dersom dette er tilfelle oppdaterer vi med opplysninger om UTFALL og KURSAVG for dette tilfellet.

STATUS7 = **G81**

3. **Ellers** (tilfellet i hovedtabell er en avgang)

a) **Hvis** HOVEDny = andre siffer i UTDKODEgml **Så** (lik hovedgruppe)

Hvis KODEny er 1 eller 3 **Så** (AVGANGgml, TILGANGny, samme hovedgruppe)

Hvis REGDATOny <= REGDATOgml **Så**

KURSTOMny > REGDATOgml **Så**

Koder en eller to endringsrecords

Hvis STATUS6gml = 3 **Så**

AVGANGgml har lik dato som gammel tilgang for dette tilfellet, vi overfører avgangsupplysningene til TILGANGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H06**,

REGDATO = REGDATOgml

Hvis STATUS6gml = 4 **Så**

AVGANGgml har lik dato som gammel endring for dette tilfellet, vi overfører avgangsupplysningene til ENDRINGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H07**,

REGDATO = REGDATOgml

Ellers

Gjør gammel avgang om til endring

STATUS7 = **H08**,

REGDATO = REGDATOgml

ENDRKODE = 2048

Koder en endringsrecord med:

STATUS7 = **H09**,

Hvis REGDATOgml = YYYY06 **Og** HOVEDny = 4 **Så**

REGDATO = MIN(årstall fra REGDATOgml || 09, KURSTOMny)

Hvis REGDATOgml = YYYY06 **Og** HOVEDny < 4 **Så**

REGDATO = MIN(årstall fra REGDATOgml || 08, KURSTOMny)

Ellers

REGDATO = REGDATOgml + 1 mnd.

Ellers

Denne recorden tas ikke med i forløpstabellen, men logges

STATUS7 = **H60**

Ellers

Hvis REGDATOgml = REGDATOny - 1mnd. **Så**

(REGDATOny er måneden etter REGDATOgml.)

Koder en eller to endringsrecords

Hvis STATUS6gml = 3 **Så**

AVGANGgml har lik dato som gammel tilgang for dette tilfellet, vi overfører avgangsupplysningene til TILGANGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H10**,

REGDATO = REGDATOgml

Hvis STATUS6gml = 4 **Så**

AVGANGgml har lik dato som gammel endring for dette tilfellet, vi overfører avgangsupplysningene til ENDRINGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H11**,

REGDATO = REGDATOgml

Ellers

Gjør gammel avgang om til endring

STATUS7 = **H12**,

REGDATO = REGDATOgml

ENDRKODE = 2048

Koder en endringsrecord med:

STATUS7 = **H13**,

REGDATO = REGDATOny

Ellers *(det er et gap mellom ny og gammel record på mer enn en måned)*

Hvis (UTDAVGgml = YYYY06 **Og** REGDATOny = YYYY08 eller 09)

Eller (UTDAVGgml = YYYY07 **Og** REGDATOny = YYYY09) **Så**

(utdanningen fortsetter etter sommeren vi koder to endringsrecord)

Hvis STATUS6gml = 3 **Så**

AVGANGgml har lik dato som gammel tilgang for dette tilfellet, vi overfører avgangsupplysningene til TILGANGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H14**,

REGDATO = REGDATOgml

Hvis STATUS6gml = 4 **Så**

AVGANGgml har lik dato som gammel endring for dette tilfellet, vi overfører avgangsupplysningene til ENDRINGgml og sletter AVGANGgml fra hovedtabellen
STATUS7 = **H15**,
REGDATO = REGDATOgml

Ellers

Gjør gammel avgang om til endring
STATUS7 = **H16**,
REGDATO = REGDATOgml
ENDRKODE = 2048

Koder en endringsrecord med:

STATUS7 = **H17**,
REGDATO = REGDATOny

Ellers

Koder avgang for gammelt utdanningstilfelle med:

STATUS7 = **H18**,
REGDATO = AVGDATO = REGDATOgml

Koder nytt utdanningstilfelle med:

STATUS7 = **H19**,
REGDATO = TILGDATO = REGDATOny

Hvis KODEny = 2 **Så** (*AVGANGgml, ENDRINGny, samme hovedgruppe*)

Hvis KURSTOMny > REGDATOgml **Så**

Koder avgangsrecord i hovedtabell om til en endringsrecord med:

Hvis STATUS6gml = 3 **Så**

AVGANGgml har lik dato som gammel tilgang for dette tilfellet, vi overfører avgangsupplysningene til TILGANGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H20**,
REGDATO = REGDATOgml

Hvis STATUS6gml = 4 **Så**

AVGANGgml har lik dato som gammel endring for dette tilfellet, vi overfører avgangsupplysningene til ENDRINGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H21**,
REGDATO = REGDATOgml

Ellers

Gjør gammel avgang om til endring

STATUS7 = **H22**,
REGDATO = REGDATOgml
ENDRKODE = 2048

Tester om det skal kodes en eller to endringsrecord i tillegg til H17:

Hvis REGDATOgml = YYYY06 **Og** HOVEDny = 4 **Så** (*YYYY = årstall*)

REGDATO = MIN(årstall i REGDATOgml || 09, KURSTOMny)

Hvis årstall i REGDATOgml = 06 **Og** HOVEDny < 4 **Så**

REGDATO = MIN(årstall i REGDATOgml || 08, KURSTOMny)

Ellers

REGDATO = REGDATOgml + 1mnd.

Hvis REGDATOny <= REGDATO (kodet over) **Så**

Koder en endringsrecord med:

STATUS7 = **H23**,
REGDATO = REGDATOny

Ellers (*koder to ekstra endringsrecord*)

Koder en endringsrecord med:

STATUS7 = **H24**,
REGDATO = REGDATO (kodet over)

Koder en endringsrecord til med:

STATUS7 = **H25**,

REGDATO = REGDATOny

Ellers

Denne recorden tas ikke med i forløpstabellen, men logges

STATUS7 = **H61**

Ellers (AVGANGgml, AVGANGny, samme hovedgruppe)

Hvis REGDATOny <= REGDATOgml **Så**

Finner ut om kurset er rangert som viktigste kurs i noen perioder i forløpstabellen (F_UTD_VIKTIGSTE). Dersom dette er tilfelle oppdaterer vi med opplysninger om UTFALL og KURSAVG for dette tilfellet.

STATUS7 = **H80**

Ellers

Koder avgangsrecord i hovedtabell om til en endringsrecord med:

Hvis STATUS6gml = 3 **Så**

AVGANGgml har lik dato som gammel tilgang for dette tilfellet, vi overfører avgangsopplysningene til TILGANGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H26**,

REGDATO = REGDATOgml

Hvis STATUS6gml = 4 **Så**

AVGANGgml har lik dato som gammel endring for dette tilfellet, vi overfører avgangsopplysningene til ENDRINGgml og sletter AVGANGgml fra hovedtabellen

STATUS7 = **H27**,

REGDATO = REGDATOgml

Ellers

Gjør gammel avgang om til endring

STATUS7 = **H28**,

REGDATO = REGDATOgml

ENDRKODE = 2048

Tester for endringer mellom record i F_UTD_KURS og H_UTD_VIKTIGSTE og koder en endringsrecord med:

STATUS7 = **H29**,

Hvis REGDATOgml = YYYY06 **Og** HOVEDny < 4 **Så** (YYYY = årstall) **Så**

Hvis REGDATOny = YYYY07 **Så**

REGDATO = REGDATOny

Ellers

REGDATO = årstall i REGDATOgml || 08

Hvis REGDATOgml = YYYY06 **Og** HOVEDny = 4 **Så**

Hvis REGDATOny er YYYY07 eller YYYY08 **Så**

REGDATO = REGDATOny

Ellers

REGDATO = årstall i REGDATOgml || 09

Ellers

REGDATO = REGDATOgml + 1mnd.

Koder en midlertidig avgang for utdanningstilfellet med:

Hvis REGDATOny >= REGDATO (kodet over) **Så** (noramaltifellene)

STATUS7 = **H30**,

REGDATO = AVGDATO = MAX(REGDATOny, REGDATO (kodet over))

b) **Ellers** (ulike hovedgrupper; dette kan være en overgang)

Hvis KODeny er 1 eller 3 **Så** (AVGANGgml, TILGANGny, ulik hovedgruppe)

Hvis REGDATOny <= REGDATOgml **Så**

KURSTOMny > REGDATOgml **Så**

Koder avgang fra gammel utdanningagruppe med:

STATUS7 = **108**,

REGDATO = REGDATOgml

Koder tilgang til ny utdanningsgruppe

STATUS7 = **109**

Hvis REGDATOgml = YYYY06 **Og** HOVEDny = 4 **Så** (YYYY = årstall)
REGDATO = MIN(årstall i REGDATOgml || 09, KURSTOMny)

Hvis årstall i REGDATOgml = 06 **Og** HOVEDny < 4 **Så**
REGDATO = MIN(årstall i REGDATOgml || 08, KURSTOMny)

Ellers

REGDATO = REGDATOgml + 1mnd.

Ellers

Denne recorden tas ikke med i forløpstabellen, men logges

STATUS7 = **I60**

Ellers

Hvis REGDATOgml = REGDATOny - 1mnd. **Så**

(REGDATOny er en måned etter REGDATOgml; vi koder en overgang)

NULLer AVGDATO i gml. record og sender denne til forløpstabell som gruppeavgang med:

STATUS7 = **I10**,

REGDATO = UTDAVG = REGDATOgml

Koder tilgang til ny utdanningsgruppe med:

STATUS7 = **I11**,

REGDATO = UTDTILG = REGDATny

Ellers *(det er et gap mellom ny og gammel record på mer enn en måned)*

Hvis (REGDATOgml = YYYY06 **Og** REGDATOny=YYYY08 eller 09)

Eller (REGDATOgml = YYYY06 **Og** REGDATOny=YYYY09) **Så**

(utdanningen fortsetter etter sommeren; vi koder en overgang og syr i hop periodene)

NULLer AVGDATO i gml. record og sender denne til forløpstabell som gruppeavgang med:

STATUS7 = **I12**,

REGDATO = UTDAVG = REGDATOgml

Koder tilgang til ny utdanningsgruppe med:

STATUS7 = **I13**,

REGDATO = UTDTILG = REGDATOny

Ellers

Gapet er for stort; koder en avgang for gammelt utdanningstilfelle med:

STATUS7 = **I14**,

REGDATO = AVGDATO = REGDATOgml

Koder nytt utdanningstilfelle med:

STATUS7 = **I15**,

REGDATO = TILGDATO = REGDATOny

Hvis KODEny = 2 **Så** (AVGANGgml, ENDRINGny, ulik hovedgruppe)

Hvis KURSTOMny > REGDATOgml **Så**

Koder om gammel avgangsrecord til en avgang for utdanningsgruppe med:

STATUS7 = **I16**,

REGDATO = UTDAVG = REGDATOgml

Koder ny endringsrecord om til tilgang til ny utdanningsgruppe med:

STATUS7 = **I17**,

Hvis REGDATOgml = YYYY06 **Og** HOVEDny = 4 **Så** (YYYY = årstall)

REGDATO = UTDTILG = MIN(årstall i REGDATOgml || 09, KURSTOMny)

Hvis REGDATOgml = YYYY06 **Og** HOVEDny < 4 **Så** (YYYY = årstall)

REGDATO = UTDTILG = MIN(årstall i REGDATOgml || 08, KURSTOM)

Ellers

REGDATO = UTDTILG = UTDAVGgml + 1mnd.

Hvis REGDATOny > REGDATO (kodet over) **Så**

Koder en endringsrecord til med:

STATUS7 = **I18**,

REGDATO = REGDATOny

Ellers

Denne recorden tas ikke med i forløpstabellen, men logges

STATUS7 = I61

Ellers (AVGANGgml, AVGANGny, ulik hovedgruppe)

Hvis REGDATOny <= REGDATOgml **Så**

Finner ut om kurset er rangert som viktigste kurs i noen perioder i forløpstabellen (F_UTD_VIKTIGSTE). Dersom dette er tilfelle oppdaterer vi med opplysninger om UTFALL og KURSAVG for dette tilfellet

STATUS7 = I80

Ellers

Koder om gammel avgangsrecord til avgang for utdanningsgruppe med:

STATUS7 = I20,

REGDATO = UTDAVG = REGDATOgml

Koder en tilgang til ny utdanningsgruppe med:

STATUS7 = I21,

Hvis REGDATOgml = YYYY06 **Og** HOVEDny < 4 **Så** (YYYY = årstall) **Så**

Hvis REGDATOny = YYYY07 **Så**

REGDATO = REGDATOny

Ellers

REGDATO = årstall i REGDATOgml || 08

Hvis REGDATOgml = YYYY06 **Og** HOVEDny = 4 **Så**

Hvis REGDATOny er YYYY07 **eller** YYYY08 **Så**

REGDATO = REGDATOny

Ellers

REGDATO = årstall i REGDATOgml || 09

Ellers

REGDATO = REGDATOgml + 1mnd.

Koder (midlertidig) avgang for utdanningstilfellet med:

STATUS7 = I22,

REGDATO = AVGDATO = MAX(REGDATOny, REGDATO kodet over)

D. **Ellers** (skal IKKE kunne oppstå)

STATUS7 = Z99

5.6.6 Avgangsrutine

Hvis ingen ny record for løpende tilfelle **Og** gap på mer enn ett år **Og** REGDATgml >= 199201

Og andre siffer i UTDKODEgml = 0 (Recorden er en avgang) **Så**

Koder avgang fra kurs med;

REGDATO = AVGDATO = AVGDATOgml (satt for den midlertidige avgangen),

STATUS6 = 1

5.7 Avvik fra offisiell statistikk

Når det gjelder opplysningene fra igangværende- og avsluttafilene, er det flere forhold som medfører at vi ikke får samsvar med offisiell statistikk:

- statistikken for igangværende utdanning for 1991-1993 bygger på bruttofiler, og ikke på de nettofilene som vi bruker (og som vi er blitt anbefalt av fagseksjonen å bruke)
- statistikken bygger mer på skoleslaget som utdanningen er tatt ved, enn selve utdanningens art, som vi bruker
- vi flytter på noen av startdatoene for kurs, slår sammen noen kurs, og avslutter de kursene som har ligget som igangværende, men som vi ikke finner igjen siden. Dette fører til at vi får andre tall for hvem som er igang med et kurs pr. oktober, og hvor mange som har avslutta et kurs i løpet av ett år.

6 Kontroller

I dette kapittelet har vi sett på noen resultater fra kjøringene på filer for utdanning, og sammenlignet med offisiell statistikk.

6.1 Kontroller på filene for igangværende utdanning, pr. 1/10 hvert år

	1991	1992	1993	1994
Antall records på grunnlagsregistrene	401 231	414 510	421 507	414 425
Antall dubletter (fjernet)	-	-	-	-
Antall records som har startdato etter filas utløp (slettes)	-	1	-	10
Antall records i miniregisteret	401 231	414 509	421 507	414 415
<u>Utdanningsnivå (miniregisteret):</u>				
Grunnskole	1 704	1 995	2 464	2 708
Videregående skole	257 806	257 939	254 197	252 279
Lærlinger	15 866	16 563	16 813	17 955
Arbeidsmarkedskurs	21 669	23 870	26 145	24 137
Høyskole	74 349	81 666	87 819	79 919
Universitet	67 372	72 909	77 027	79 509
Herav som har skiftet verdi på utdanningens art	49 237	52 571	55 325	57 256
<u>Dato:</u>				
Antall records som mangler startdato	-	-	-	-
Antall records hvor elevens startdato er brukt (mrk_dl=1)	43 389	48 098	52 614	58 925
Antall records hvor kursets startdato er brukt (mrk_dl=3)	357 697	366 369	368 845	355 471
Antall records hvor uoppgitt startdato er satt (mrk_dl=5)	145	42	48	19
<u>Fødselsnummerkontroll:</u>				
Antall med gyldig fødselsnummer (mrk_fnr=0)	400 163	413 472	420 423	413 209
Antall med gyldig d-nummer (mrk_fnr=1)	-	-	-	-
Antall med nuller i personnummer (mrk_fnr=2)	1	-	9	1
Antall med ugyldig fødselsnummer (mrk_fnr=3)	1 067	1 037	1 075	1 192
<u>Offisiell statistikk:</u>				
Grunnskole
Videregående skole	259 477	260 431	257 420	244 938
Høyskole	80 616	88 390	94 623	89 797
Universitet	68 249	73 778	77 951	79 509

6.1 (forts.) Kontroller på filer for igangværende utdanning, pr. 1/10 hvert år

	1995	1996	1997	1998
Antall records på grunnlagsregistrene	412 415	417 405	414 559	414 178
Antall dubletter (fjernet)	-	-	-	-
Antall records som har startdato etter filas utløp (slettes)	106	2	5	-
Antall records i miniregisteret	412 309	417 403	414 554	414 178
<u>Utdanningsnivå (miniregisteret):</u>				
Grunnskole	2 677	1 960	1 080	585
Videregående skole	247 402	234 285	233 016	229 953
Lærlinger	19 374	27 216	31 827	32 223
Arbeidsmarkedskurs	19 310	16 361	10 706	7 719
Høyskole	79 273	96 043	96 974	102 512
Universitet	82 957	85 115	83 484	81 128
Herav som har skiftet verdi på utdanningens art	59 926	60 439	.	.
<u>Dato:</u>				
Antall records som mangler startdato	-	-	-	-
Antall records hvor elevens startdato er brukt (mrk_dl=1)	71 485	78 282	83 558	100 808
Antall records hvor kursets startdato er brukt (mrk_dl=3)	339 293	339 106	330 989	313 363
Antall records hvor uoppgitt startdato er satt (mrk_dl=5)	1 531	15	7	7
<u>Fødselsnummerkontroll:</u>				
Antall med gyldig fødselsnummer (mrk_fnr=0)	410 810	415 908	412 735	412 635
Antall med gyldig d-nummer (mrk_fnr=1)	-	3	2	6
Antall med nuller i personnummer (mrk_fnr=2)	7	4	2	4
Antall med ugyldig fødselsnummer (mrk_fnr=3)	1 492	1 488	1 815	1 533
<u>Offisiell statistikk:</u>				
Grunnskole
Videregående skole	235 501	235 496	233 818	230 115
Høyskole	93 788	96 786	97 257	102 935
Universitet	82 957	84 955	83 484	81 128

6.2 Kontroller på filene for avslutta utdanning i “skole”-året

	1992	1993	1994	1995
Antall records på grunnlagsregisteret	334 901	357 255	360 201	374 804
<u>Utdanningsnivå på grunnlagsregisteret:</u>				
Ungdomsskole	55 472	54 433	55 732	55 967
Videregående skole	195 345	214 017	215 414	231 416
Høyskole	47 260	49 253	48 515	45 568
Universitet	36 824	39 552	40 540	41 853
<u>Dubletter:</u>				
Antall rene dubletter som er fjernet	6	5	-	2
Antall records som er fjernet fordi perioden blir gal	48	62	59	11
Antall records som fjernes fordi kurs og periode er likt	101	50	27	63
Antall records som fjernes/slås sammen fordi periode er overlappende	1 863	2 148	1 972	1 944
<u>Datoer:</u>				
Antall med startdato etter utløpet av filas tidsperiode (slettes)	-	-	-	-
Antall med stoppdato etter utløpet av filas tidsperiode (slettes)	-	-	-	-
Antall med stoppdato før filas tidsperiode (slettes)	-	1	37	127
Antall records som det blir satt/rettet startdato på ¹	44 860	48 064	42 739	53 441
Antall records fjernet pga. stoppdato er før 199201	31 630	-	-	-
Antall records på miniregisteret	301 253	354 989	358 104	372 657
<u>Fødselsnummerkontroll:</u>				
Antall med gyldige fødselsnummer	300 692	354 129	357 292	371 641
Antall med gyldig d-nummer	-	-	-	1
Antall med nuller i personnummer	10	5	8	8
Antall med ugyldig fødselsnummer	551	855	804	1 007
<u>Utdanningsnivå på miniregisteret:</u>				
Ungdomsskole	55 388	54 402	55 728	55 966
Videregående skole	188 040	218 251	218 887	230 434
Høyskole	38 727	43 772	43 806	45 518
Universitet	19 098	38 564	39 683	40 739
<u>Offisiell statistikk:</u>				
Grunnskole	55 093
Videregående skole	200 551	220 432	221 559	199 159
Universitet og høyskole	67 774	70 447	72 960	76 046

¹ De fleste som får satt startdato er universitetsstudenter som mangler startdato på avsluttafila.

6.2 (forts.) Kontroller på filene for avslutta utdanning i "skole"-året










	1996	1997	1998
Antall records på grunnlagsregisteret	395 533	396 650	396 810
<u>Utdanningsnivå på grunnlagsregisteret:</u>			
Ungdomsskole	56 026	54 472	53 443
Videregående skole	246 470	235 222	246 003
Høyskole	50 101	52 143	47 336
Universitet	42 936	54 813	50 028
<u>Dubletter:</u>			
Antall rene dubletter som er fjernet	-	14	30
Antall records som er fjernet fordi perioden blir gal	13	96	47
Antall records som fjernes fordi kurs og periode er likt	41	211	106
Antall records som fjernes/slås sammen fordi periode er overlappende	1 928	4 056	3 544
<u>Datoer:</u>			
Antall med startdato etter utløpet av filas tidsperiode (slettes)	-	-	-
Antall med stoppdato etter utløpet av filas tidsperiode (slettes)	-	-	-
Antall med stoppdato før filas tidsperiode (slettes)	136	170	9
Antall records som det blir satt/rettet startdato på ²	59 992	75 278	83 713
Antall records på miniregisteret	393 415	392 103	393 074
<u>Fødselsnummerkontroll:</u>			
Antall med gyldige fødselsnummer	392 052	390 875	391 517
Antall med gyldig d-nummer	6	3	8
Antall med nuller i personnummer	8	1	3
Antall med ugyldig fødselsnummer	1 349	1 224	1 546
<u>Utdanningsnivå på miniregisteret:</u>			
Ungdomsskole	56 022	54 467	53 423
Videregående skole	245 676	234 784	245 616
Høyskole	49 825	51 745	47 144
Universitet	41 892	51 107	46 891
<u>Offisiell statistikk:</u>			
Grunnskole	56 026	53 926	53 056
Videregående skole	243 075	208 421	214 968
Universitet og høyskole	96 432	80 487	73 061

² De fleste som får satt startdato er universitetsstudenter som mangler startdato på avsluttafila.



7 Diagrammer – Igangværende utdanning

Det er utarbeidet flere typer diagrammer over IT-rutinene. Diagrammene er også laget på flere nivåer. Dataflytdiagram nivå 1 gir en grov oversikt over systemet. Fysisk dataflytdiagram viser en detaljert oversikt over alle program, og sammenhengen mellom data og program. Datamodellen viser det endelige resultat, dvs. oversikt/spesifikasjon av data (forløpsdata) lagret i databasen (Oracle). Det er utarbeidet diagrammer både for rutiner i forbindelse med danning av miniregister, og rutiner for danning av forløpsdata i databasen. IT-dokumentasjonen er utarbeidet i hht. SSB's egne metode for systemutvikling «Håndbok i utvikling av statistikkssystemer - Med vekt på IT-metode».

Følgende standardsymboler fra IT-metoden er benyttet:

Symbol	Symbolnavn	Tilhørende teknikk
	Database/datasett (ORACLE og SAS)	Fysisk dataflyt
	Sekvensielle filer (flate filer)	Fysisk dataflyt
	Dokument (papirtabeller)	Fysisk dataflyt
	Program	Fysisk dataflyt, dialogsystem, DFD0
	Dataflytretning	Fysisk dataflyt/ DFD0, DFD1, dialogsystem
	Peker til fortsettelse	Fysisk dataflyt
	Prosess	DFD1
	Datalager (entitet)	(DFD0), DFD1
	Fysisk datatabell i databasen	Datamodell

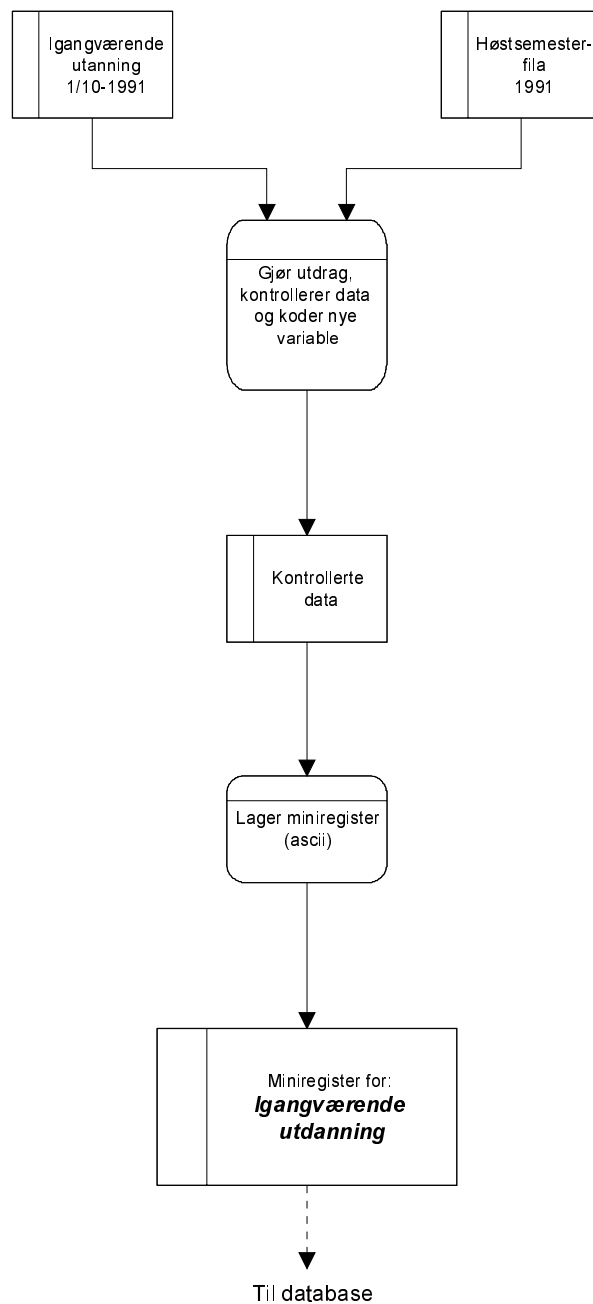
I tillegg benyttes følgende standardsymboler:

Symbol	Symbolnavn	Tilhørende teknikk
	Arbeidsoperasjon	Flytdiagram for program-rutiner
	Test	Flytdiagram for program-rutiner

7.1 Dataflytdiagram nivå 1 - miniregister (SAS)

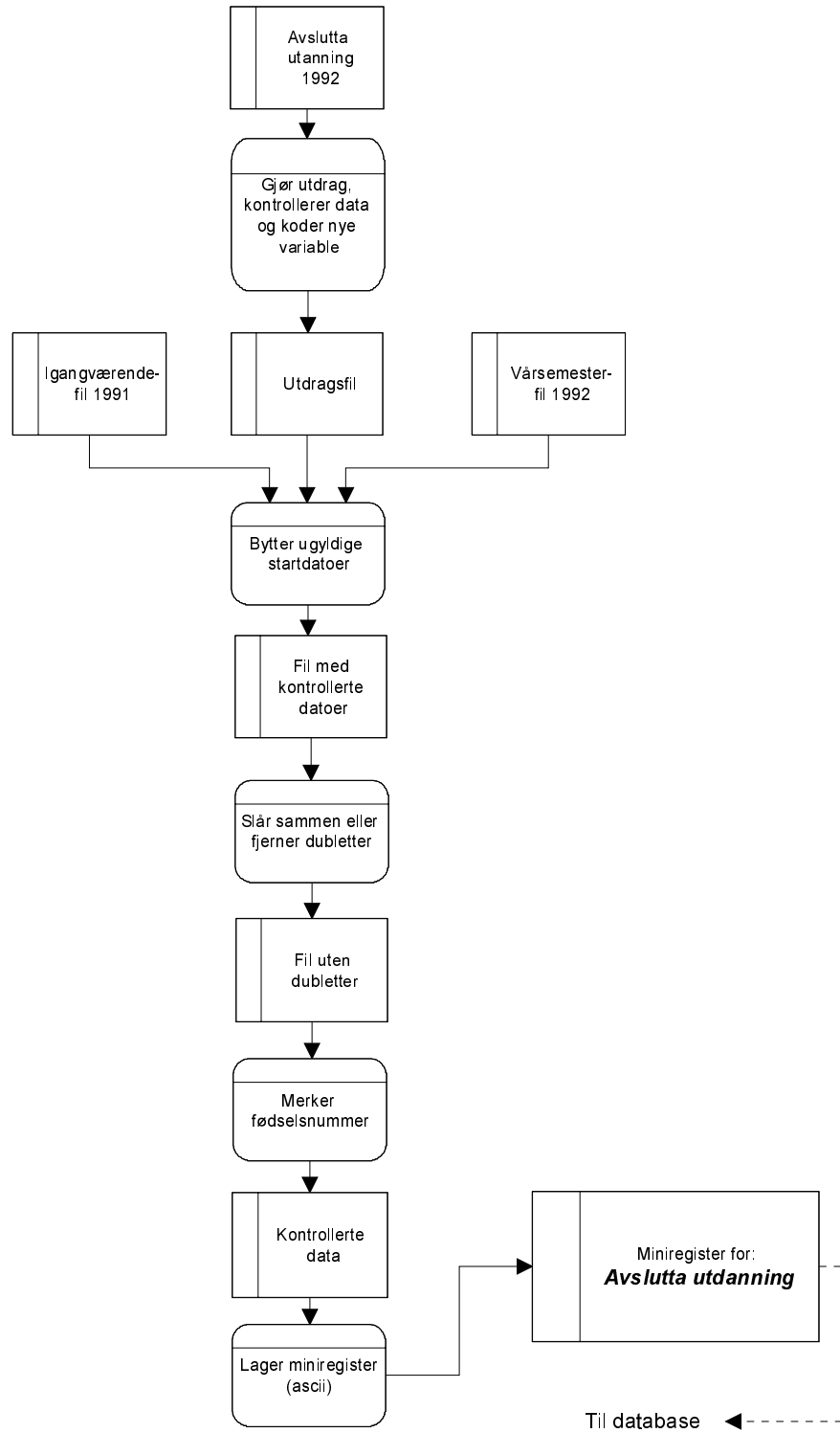
<i>FD-TRYGD</i>	X:350\TRYGDFOB\IEDBDOK\DIAGRAM\UTDANN\IGANG\MR_FLYT.AF3
	Diagramansvarlig: JLa 5/4-2000
DFD1 - Danning av miniregister for igangværende utdanning	

DFD1 gjelder tilsvarende for de senere årgangene.



DFD1 - Danning av miniregister for avsluttet utdanning

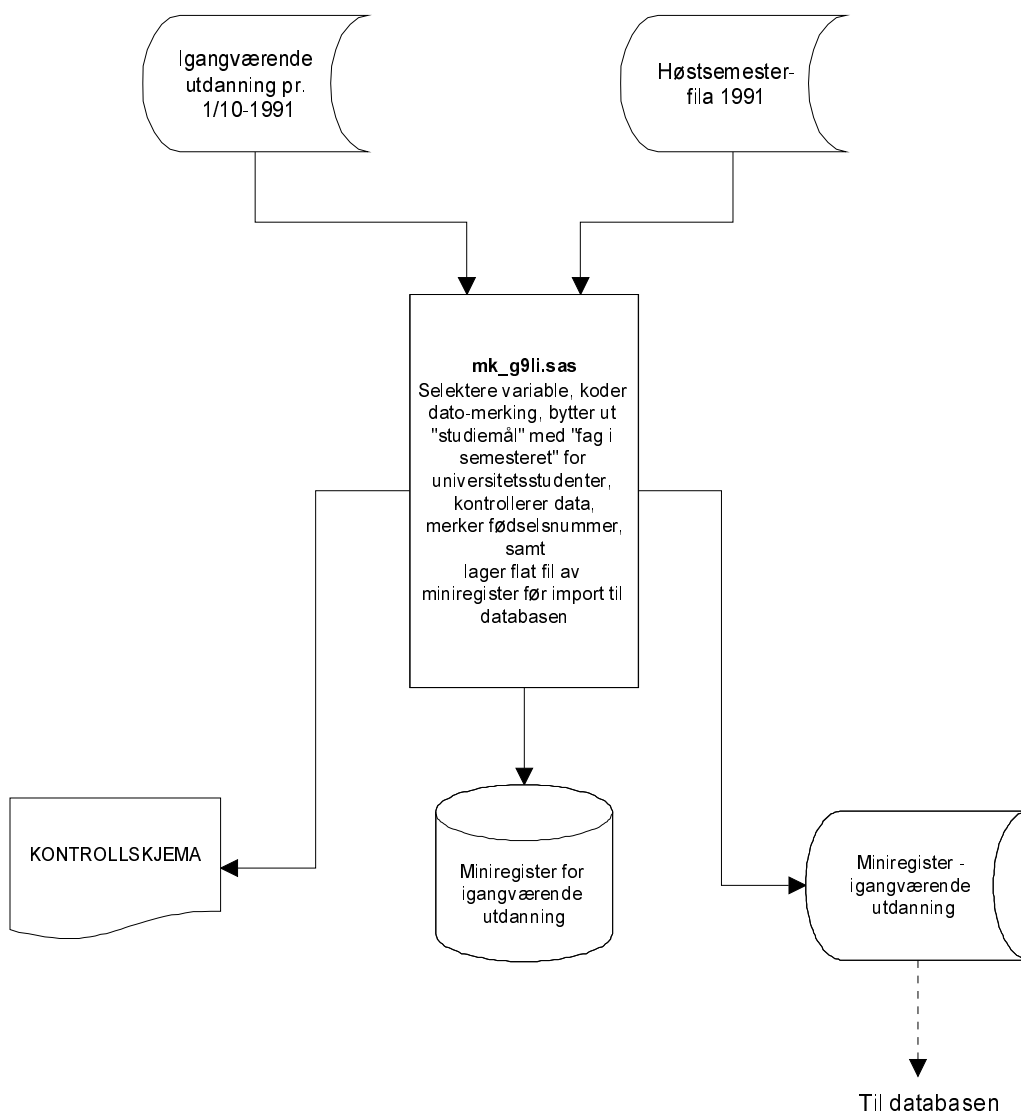
DFD1 gjelder tilsvarende for de senere årgangene.



7.2 Fysisk dataflytdiagram - miniregister (SAS)

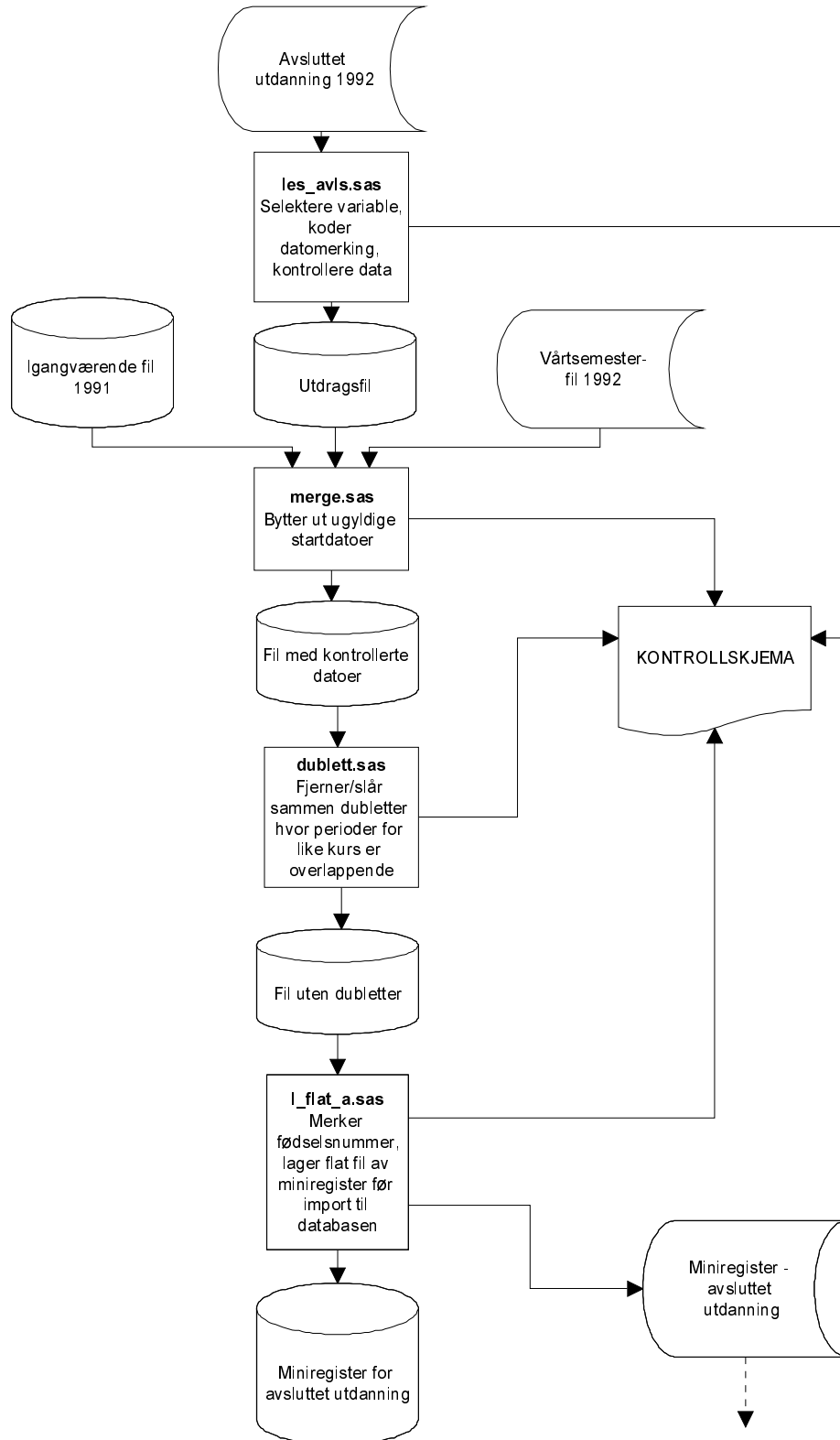
<i>FD-TRYGD</i>	X:350\TRYGD\FOB\EDBDOK\DIAGRAM\UTDANN\IGANGV\MR_FLYT.AF3
	Diagramansvarlig: JLa 5/4-2000
Fysisk dataflytdiagram - Danning av miniregister for igangværende utdanning	

Diagrammet gjelder tilsvarende for de senere årgangene.



Fysisk dataflytdiagram - Danning av miniregister for avsluttet utdanning

Diagrammet gjelder tilsvarende for de senere årgangene.



7.3 Programbeskrivelse av miniregister-rutiner (SAS)

Nedenfor følger programheadinger til alle programmer som er benyttet til danning og kontroll av miniregistre på utdanning. Programmer som er benyttet, ligger på Unix.

7.3.1 Igangværende utdanning

Programmet ligger under \$TRYGDFOB/prog/SAS/utdann/gYY/igangv/...

```
*****
Prosjekt .....: FD-Trygd
Program navn .....: mk_g91i.sas
Skrevet av .....: KBs
Dato .....: 19.09.1997
Versjon .....: 1.00
Programmets funksjon .: Lage miniregister av igangværendefiler,
                        dvs. leser inn igangværendefila, omkoder,
                        datokontoll, bytter ut studiemål med fag i
                        semesteret for universitetsstudenter
                        samt fødselsnummerkontroll og dublettkontroll
                        leser til slutt ut til tekstfil.
Programmet kaller ....: $TRYGDFOB/prog/felles/sas/mrk_fnr.sas
Filer inn.....: $TRYGDFOB/data/utdann/g91/grunnlag/igangv/g91.txt
                  $TRYGDFOB/data/utdann/g91/grunnlag/fagisem/g91h2br.txt
Filer ut.....: $TRYGDFOB/data/utdann/g91/minireg/igangv/g91.ssd01
                  $TRYGDFOB/data/utdann/g91/minireg/igangv/g91.txt
Endret når .....: 23.01.1998
Endret av .....: JLa
Grunn til endring ....: Lage rutinen fulstendig
*****/
```

7.3.2 Avsluttet utdanning

Programmene ligger under \$TRYGDFOB/prog/SAS/utdann/gYY/avslutt/...

```
/******
Prosjekt .....: FD-Trygd
Program navn .....: les_avsl.sas
Skrevet av .....: KBs,JLa
Dato .....: 19.09.1997
Versjon .....: 1.02
Programmets funksjon .: Leser inn avslutta fil fra utdanning, omkoder,
                        og skriver til permanent sasfil, som brukes
                        videre.
Programmet kaller ....:
Filer inn.....: $TRYGDFOB/data/utdann/g92/grunnlag/avslutt/g92.txt
Filer ut.....: $TRYGDFOB/data/utdann/g92/minireg/avslutt/g92a.ssd01
Endret når .....: 24.03.1998
Endret av .....:
Grunn til endring ....: Ny seleksjon etter datovariable.
*****/
```

```
/******
Prosjekt .....: FD-Trygd
Program navn .....: merge.sas
Skrevet av .....: JLa
Dato .....: 10.03.1998
Versjon .....: 1.00
Programmets funksjon .: Legge inn startdatoer for de som har ugyldige
                        startdatoer på avsluttafila
Programmet kaller ....:
Filer inn.....: $TRYGDFOB/data/utdann/g91/minireg/igangv/g91.ssd01
                  $TRYGDFOB/data/utdann/g92/grunnlag/fagisem/g92h1.txt
                  $TRYGDFOB/data/utdann/g92/minireg/avslutt/g92a.ssd01
Filer ut.....: $TRYGDFOB/data/utdann/g92/minireg/avslutt/g92b.ssd01
Endret når .....: 24.03.1998
Merknader .....: les_avsl.sas må være kjørt før dette
                  programmet kjøres
*****/
```



```

/*****
Prosjekt .....: FD-Trygd
Program navn .....: dublett.sas
Skrevet av .....: KBs
Dato .....: 13.03.1998
Versjon .....: 1.00
Programmets funksjon .: Sjekker dubletter, og fjerner/slår sammen
                        dubletter hvor perioden er overlappende
Programmet kaller ....:
Filer inn.....: $TRYGDFOB/data/utdann/g92/minireg/avslutt/g92b.ssd01
Filer ut.....: $TRYGDFOB/data/utdann/g92/minireg/avslutt/g92c.ssd01
Endret når .....: 24.03.1998
Endret av .....: JLa
Merknader .....: merge.sas må være kjørt før dette programmet
                  kjøres
*****/

```

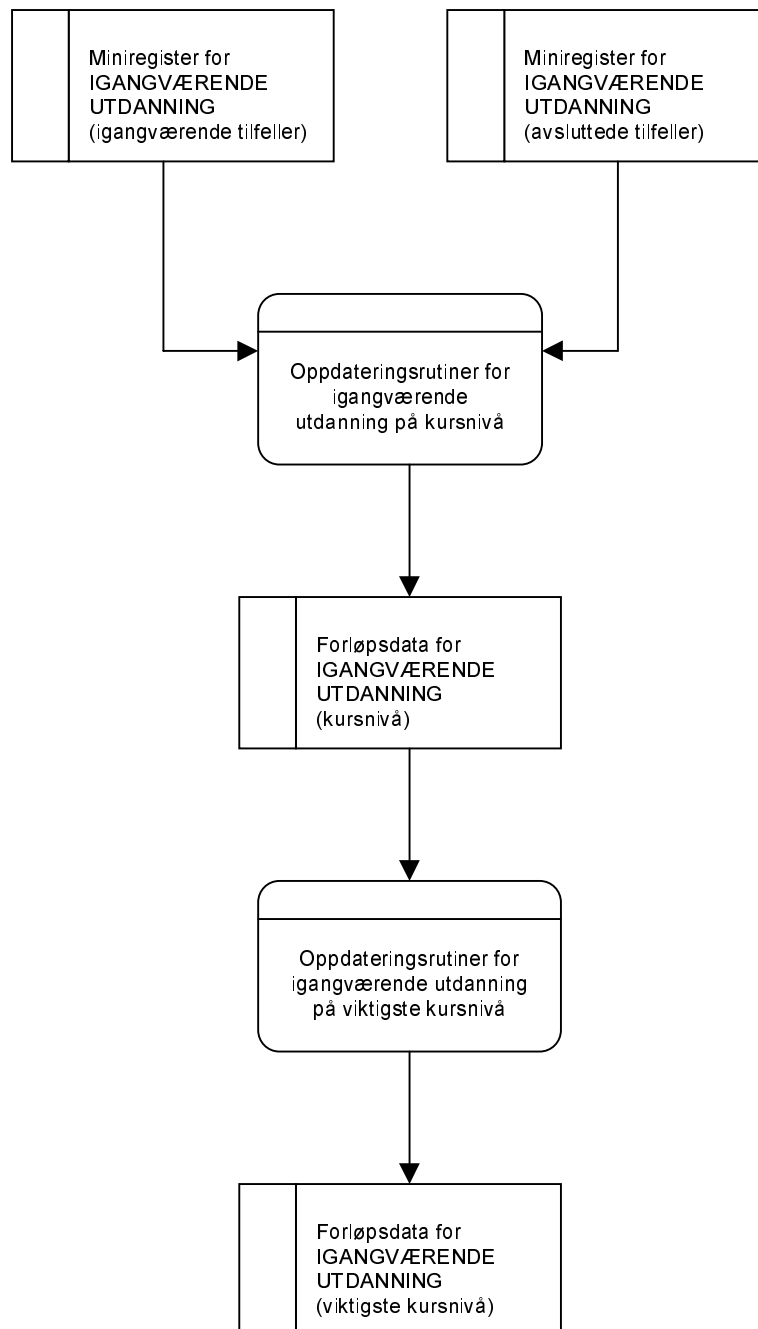
```

/*****
Prosjekt .....: FD-Trygd
Program navn .....: l_flat_a.sas
Skrevet av .....: JLa
Dato .....: 16.03.1998
Versjon .....: 1.00
Programmets funksjon .: Kjører fødselsnummerkontroll og lager tekst-
                        fil av resultatet
Programmet kaller ....: $TRYGDFOB/prog/felles/sas/mrk_fnr.sas
Filer inn.....: $TRYGDFOB/data/utdann/g92/minireg/avslutt/g92c.ssd01
Filer ut.....: $TRYGDFOB/data/utdann/g92/minireg/avslutt/g92.ssd01
                  $TRYGDFOB/data/utdann/g92/minireg/avslutt/g92.txt
Endret når .....: 24.03.1998
Endret av .....:
Merknader .....: dublett.sas må være kjørt før dette
                  programmet kjøres
*****/

```

7.4 Dataflytdiagram nivå 1 - database (Oracle)

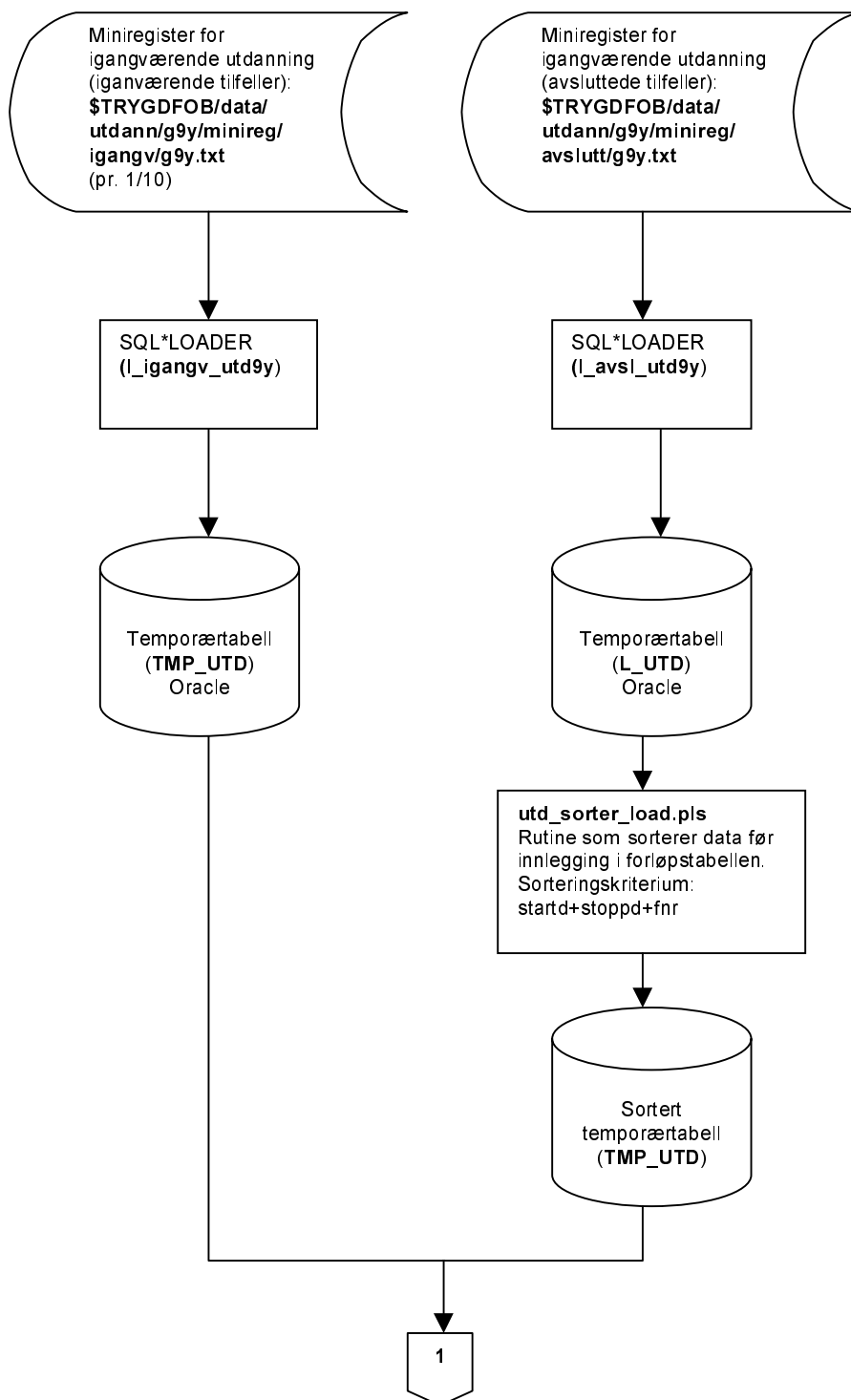
<i>FD-TRYGD</i>	Diagramansvarlig: JoJ	Dato: 28.02.2000
DFD1 - Databaseprosesser som danner forløp for IGANGVÆRENDE UTDANNING		

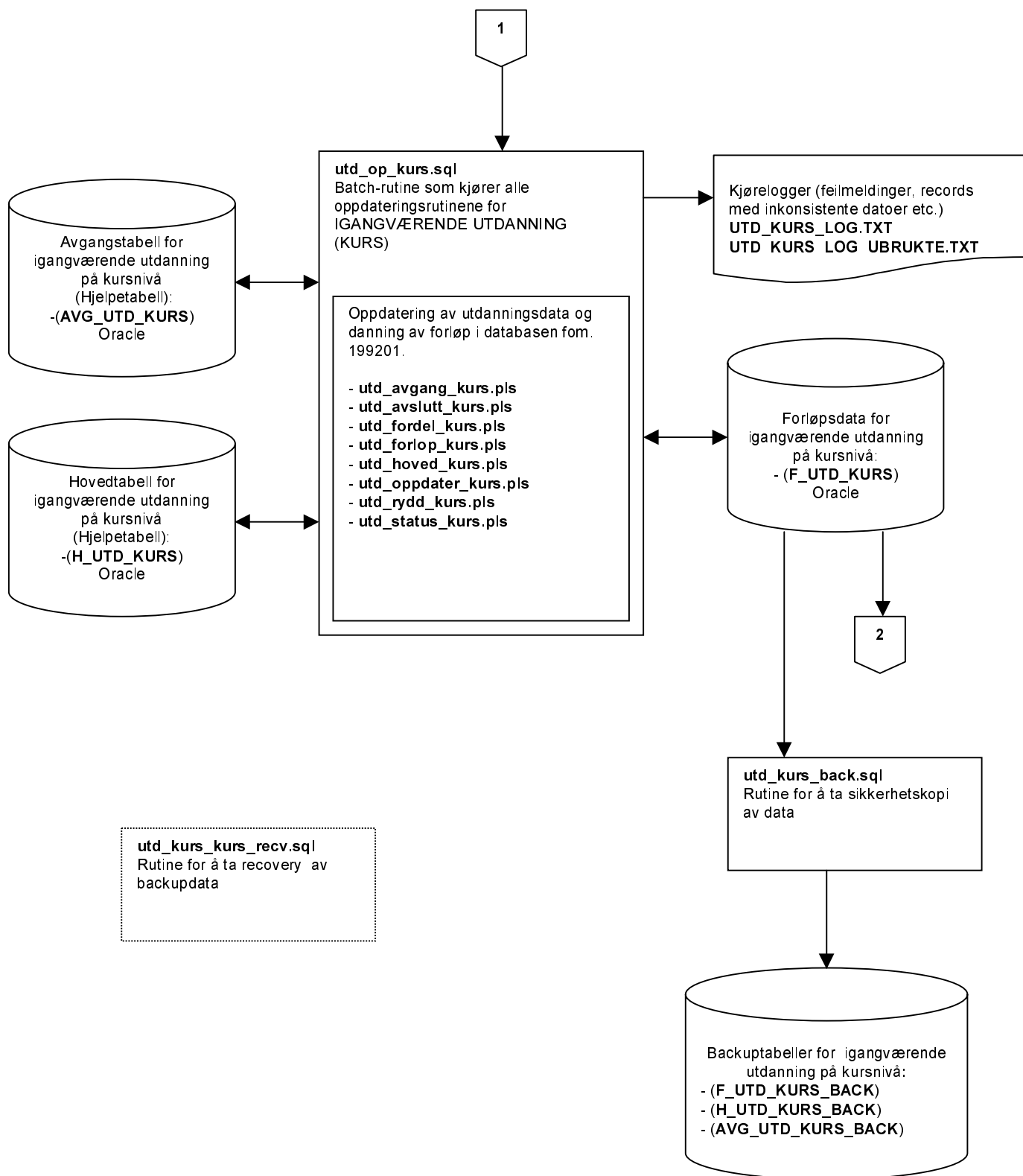


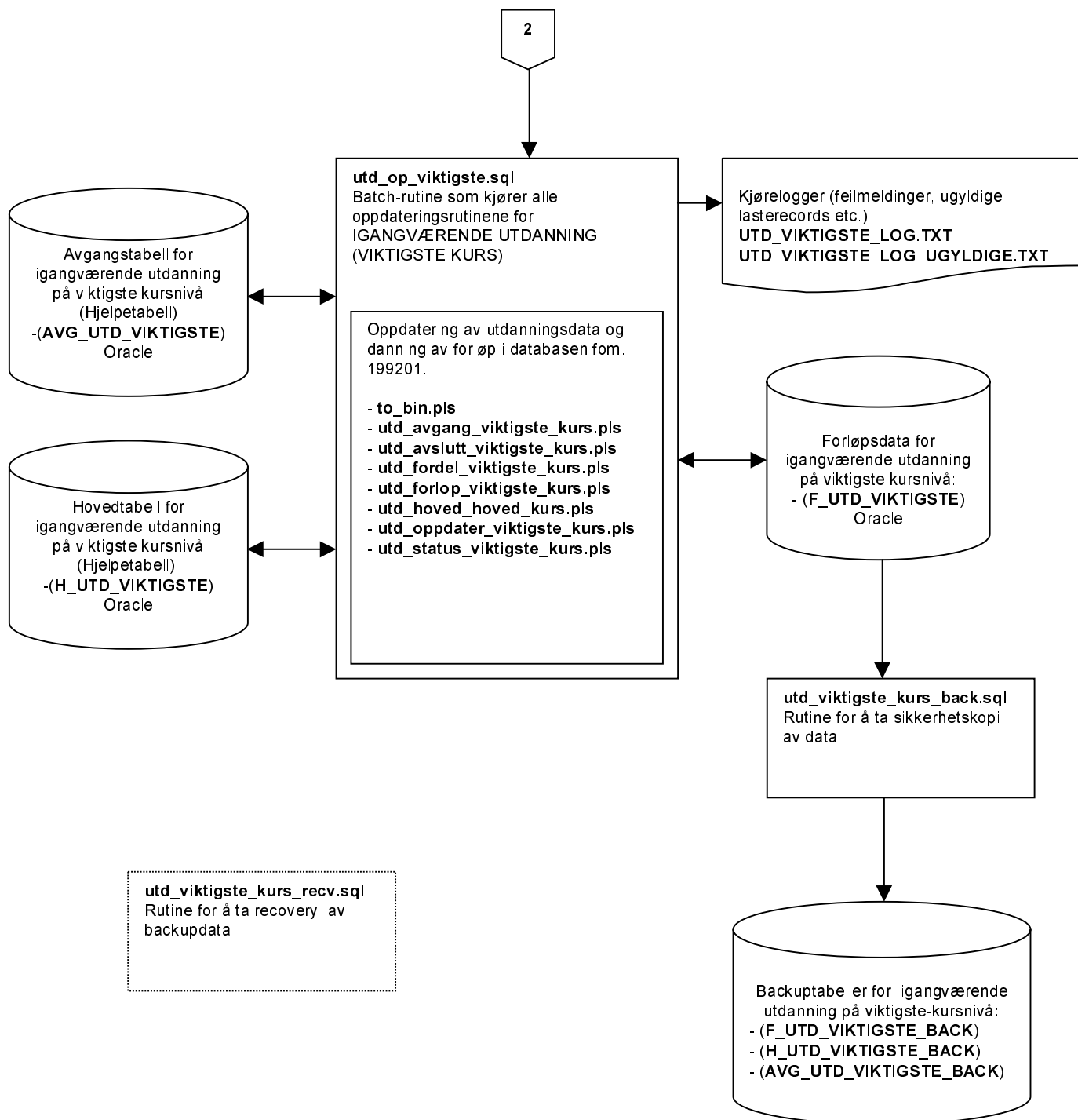
7.5 Fysisk dataflytdiagram - database (Oracle)

FD-TRYGD	Diagramansvarlig: JoJ	Dato: 25.02.2000
Fysisk dataflytdiagram - databaserutiner for IGANGVÆRENDE UTDANNING		

y = siste siffer i årstall







7.6 Databaserutiner for igangværende utdanning (Oracle)

Datagrunnlaget for å danne forløp for igangværende utdanning består av filer fra avslutta utdanning og fra igangværende utdanning. Avslutta-filene er såkalte brutto-filer som betyr at de inneholder alle avsluttede kurs i løpet av ett år (YYYY₁10 - YYYY09). Igangværendefilene er derimot nettofiler og inneholder kun en record pr. person pr. fil pr. YYYY10.

Filenes struktur er gjort helt like ved tilrettelegging av miniregisteret for å lette arbeidet med å danne forløp i databasen. Vi danner dermed forløp i databasen ved hjelp av de samme PL/SQL-rutinene både for avslutta- og igangværende utdanning. Data fra miniregisteret blir imidlertid lastet inn ved hjelp av ulike load-rutiner for igangværende- og avsluttafilene.

Vi danner forløp på to nivåer i to ulike steg i to forskjellige forløpstabeller. Det første forløpsnivået er på kursnivå som er det fineste (laveste) nivå for et utdanningstilfelle. På dette nivået registrerer vi alle tilganger, endringer og avganger for hvert kurstilfelle i forløpstabellen F_UTD_KURS. Deretter danner vi forløp for det viktigste kurset til enhver tid for hvert utdanningstilfelle. Kriteriene for å finne viktigste kurs er beskrevet i 5.6 og senere i kapittel 7.9.

7.6.1 Lagringssted for databaserutiner

Programmene som er benyttet for å danne forløp for igangværende utdanning i Oracle-databasen er lagret på UNIX under følgende kataloger:

For igangværende utdanning (kursnivå):

```
$STRYGDFOB/prog/ORACLE/utdann/igangv/kurs/BACKUP/.. .  
$STRYGDFOB/prog/ORACLE/utdann/igangv/kurs/DATATEST/.. .  
$STRYGDFOB/prog/ORACLE/utdann/igangv/ kurs/OPPDATER/..  
$STRYGDFOB/prog/ORACLE/utdann/igangv/ kurs/SORTER/.. .  
$STRYGDFOB/prog/ORACLE/utdann/igangv/ kurs/SQLLOAD/.. .
```

For igangværende utdanning (viktigste-kursnivå):

```
$STRYGDFOB/prog/ORACLE/utdann/igangv/viktigste/BACKUP/.. .  
$STRYGDFOB/prog/ORACLE/utdann/igangv/viktigste/DATATEST/.. .  
$STRYGDFOB/prog/ORACLE/utdann/igangv/viktigste//OPPDATER/..
```

7.6.2 Innlastingshastighet under danning av forløp

Rutinene som danner forløp for utdanning er store og meget krevende for serveren. Disse bør ikke parallellkjøres med andre innleggingsrutiner eller andre krevende spørringer. Erfaring har vist at innleggingshastigheten synker dramatisk i perioder hvor serveren har mange prosesser å arbeide med samtidig (innlastingstiden kan variere med flere timer for en og samme årgang alt etter som hvor mye serveren har å gjøre).

TIPS:

Følg med på *top* på UNIX før danning av forløp startes for å se om andre krevende og/eller lange kjøring foregår.

1. Prøv å finne ut hvilke tidspunkter/tidsluker hvor det er generelt liten trafikk på serveren. DBA og/eller systemoperatørene på serveren kan gi svar på dette.
2. Prøv å avtale med andre som bruker serveren til tunge oppgaver om å få en «ledig» tidsluke for deg selv på serveren.

3. Det er laget en liten prosedyre som baserer seg på UTL_FILE (logg_status) som gjør det mulig å følge med på innlastingshastigheten under danning av forløp.

På UNIX: se på filene:

/u01/app/oracle/admin/TFH3/ora_out/utd_kurs_log.txt

/u01/app/oracle/admin/TFH3/ora_out/utd_kurs_log_ubrukte.txt

/u01/app/oracle/admin/TFH3/ora_out/utd_viktigste_log.txt

/u01/app/oracle/admin/TFH3/ora_out/utd_viktigste_log_ugyldige.txt

Filene vil inneholde all informasjon om innlastingen (hastighet, tidsforbruk samt evt. feil som har oppstått). En innlastingshastighet på over 180 records pr. sekund er bra, under 100-120 records pr. sekund er mindre bra. "Samplingshastigheten" kan endres i rutinene UTD_OPPDATER_KURS.PLS, UTD_OPPDATER_VIKTIGSTE_KURS.PLS.

7.6.3 Kontroll av batch-kjøringer i databasen

Kontroller at resultatet av batch-kjøringen er som forventet. Loggfilene som er beskrevet i punkt 7.6.2 skal **alltid** sjekkes for ev. feilmeldinger etter hver kjøring!

7.7 Programbeskrivelse av databaserutiner på kursnivå (Oracle)

7.7.1 Innlastingsrekkefølge for igangværende- og avsluttafilene

For at korrekt forløp skal dannes er det viktig å laste inn data fra avslutta- og igangværende filene i kronologisk riktig rekkefølge: først **igangv.91**, deretter **avsl.92**, så **igangv.92**, **avsl.93**, **igangv.93** osv. En årgang regnes som avslutta + igangværende for samme år (unntaket er 1992 hvor vi først laster igangv.91):

1992 : igangv.91 + avsl.92 + igangv.92

1993 : avsl.93 + igangv.93

1994 : avsl.94 + igangv.94

.

.

1998 : avsl.98 + igangv.98

NB! Det er viktig å merke seg at etter at hver årgang er kjørt mot basen og forløp er dannet så må det dannes forløp på viktigste kursnivå for samme årgang **før** flere filer for kurs legges inn (se punkt 7.9).

Etter at en årgang (avslutta.YY + igangv.YY) er etablert i databasen (forløp er dannet), er innholdet i tabellene (F_UTD_KURS, H_UTD_KURS og AVG_UTD_KURS) tatt ut og lagret på flate filer.

Dette er gjort for å slippe å etablere forløp på kursnivå på nytt i tilfelle en må kjøre viktigste-kursnivå på nytt senere. Flate filer med filbeskrivelse er lagret på Unix under:

\$TRYGDFOB/data/ora_back/utdann/..

7.7.2 SQL*Loader-rutine for import til Oracle

Før nye data fra et årssett legges inn i de temporære tabellene **L_UTD** og **TMP_UTD** må en først slette evt. gammelt innhold i disse. Dette gjøres enklest med kommandoen «**TRUNCATE TABLE <tabellnavn>**».

Vi starter med å laste data fra flate filer inn i de temporære tabellene vha. egne innlastingskript for SQL*Loader.

For igangværendefila:

Under UNIX-katalogen \$TRYGDFOB/prog/ORACLE/utdann/igangv/kurs/SQLLOAD/ kjøres innlastingskriptene slik: **SQLLOAD trygdfob/<passord>@tfh3 l_igangv_utd<YY>.ctl DIRECT = TRUE** hvor YY = år. Data er nå lagt inn i tabellen **TMP_UTD**, og er klar for oppdateringsrutinene for å danne forløp på kurs-nivå i databasen.

For avsluttafila:

Under UNIX-katalogen \$TRYGDFOB/prog/ORACLE/utdann/igangv/kurs/SQLLOAD/ kjøres innlastingskriptene slik: **SQLLOAD trygdfob/<passord>@tfh3 l_avsl_utd<YY>.ctl DIRECT = TRUE** hvor YY = år. Data er nå lagt inn i tabellen **L_UTD**, og må sorteres (se neste punkt) for oppdateringsrutinene for å danne forløp på kurs-nivå i databasen kan kjøres.

7.7.3 UTD_SORTER_LOAD.PLS (Klargjør avsluttadata for hovedprogrammet)

Før data fra avsluttafila kan legges inn i databasen må **L_UTD** sorteres først og legges over i tabellen **TMP_UTD**. Dette gjøre vi ved hjelp av rutinen **STRYGDFOB/prog/ORACLE/utdann/igangv/kurs/SORTER/utd_sorter_load.pls**.

```

/*****
Prosjekt .....: FD-Trygd (IGANGVÆRENDE UTDANNING)
Program navn .....: utd_sorter_load.pls
Skrevet av .....: JoJ
Dato .....: 17.06.1999
Versjon .....: 1.0
Programmets funksjon ..: Sorterer data i loadtabellen (records fra
avsluttafila) L_UTD og legger resultatet ut i
TMP_UTD. Records sorteres for å sikre at kursene
fra avslutta-fila prosesseres i riktig rekkefølge.
Sorteringskriterium: STARTD, STOPPD, FNR. Startes
som batch-jobb i SQL*PLUS: SQL>@utd_sorter_load.pls

Filer tmp .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

7.7.4 UTD_OP_KURS.SQL (Hovedprogram for oppdatering på kursnivå i Oracle)

Rutinen fungerer som hovedprogram for oppdatering av kurs for igangværende utdanning **fom. 199201** i databasen og kaller ulike sub-program for danning av forløpsdata. Data lastes inn i loadtabellen **L_UTD**, og temporærtabelen **TMP_UTD** for ett og ett år som beskrevet i punktene over. Hovedprogrammet prosesserer hvert årssett av data fortløpende. Dette gjøres med rutinen **STRYGDFOB/prog/ORACLE/utdann/igangv/kurs/OPPDATER/utd_op_kurs.sql**.

Alle personer i nyeste datagrunnlag, dvs. personer i den temporære tabellen **TMP_UTD**, sjekkes mot forrige års datagrunnlag (innholdet i hovedtabellen **H_UTD_KURS**), record for record. Ut i fra bestemte regler, som angitt i (5.5), avgjøres det om det skal kodes en *tilgang*, *avgang* eller *endring* for hver enkelt record som prosesseres. *Endring* betyr endret verdi i en eller flere av internvariablene siden forrige registrering.

Kildekoden til dette programmet er lagt inn i dokumentasjonen for at det skal være lettere å se rekkefølgen rutinene kompiles i.

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING - KURSNIVÅ)
Program navn .....: utd_op_kurs.sql
Skrevet av .....: JoJ
Dato .....: 01.07.1999
Versjon .....: 1.0
Programmets funksjon ..: Batch-jobb som kaller alle oppdateringsrutiner for
igangværende utdanning på kursnivå.
NB! Før jobben startes bør det tas sikkerhetskopi
av alle berørte databasetabeller (Benytt
rutinen utd_kurs_back.sql).

Filer inn .....: Alle oppd.rutiner på kursnivå for igangv. UTD.
Filer ut .....:
Endret når .....:
Endret av .....:
Grunn til endring ....:
*****/
```

```
/* Denne sender meldinger til en egen logg-fil */
```

SPOOL UTD_OP_KURS

```
/* Kompilerer alle PL/SQL-prosedyrer som skal benyttes */
@utd_forlop_kurs.pls /
@utd_hoved_kurs.pls /
@utd_avgang_kurs.pls /
@utd_rydd_kurs.pls /
@utd_avslutt_kurs.pls /
@utd_status_kurs.pls /
@utd_fordel_kurs.pls /
@utd_oppdater_kurs.pls /
COMMIT;

/* Setter på en timer for å se hvor lang tid jobben tar */
SET TIMING ON;
SET TIME ON;

/* Her kalles styrerutine for å lese inn fra riktig månedsfil */
EXECUTE utd_oppdater_kurs;
COMMIT;
-- Til slutt slettes alle prosedyrer
DROP PROCEDURE utd_avslutt_kurs;
DROP PROCEDURE utd_fordel_records;
DROP PROCEDURE utd_forlop_kurs;
DROP PROCEDURE utd_hoved_kurs;
DROP PROCEDURE utd_rydd_kurs;
DROP PROCEDURE utd_avgang_kurs;
DROP PROCEDURE utd_oppdater_kurs;
DROP PROCEDURE utd_bestem_status;
COMMIT;

/* Lukker spool-meldingsfilen */
SPOOL OFF

EXIT;
```

7.7.5 UTD_FORLOP_KURS.PLS

```
/******
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING - KURSNIVÅ)
Program navn .....: utd_forlop_kurs.pls
Skrevet av .....: JoJ
Dato .....: 18.06.1999
Versjon .....: 1.0
Programmets funksjon ..: Legger behandlede records ut i forløpstabellen for
kurs: F_UTD_KURS.

Programmet kaller ..:
Programmet kalles fra : utd_fdel.pls
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ..:
*****/
```

7.7.6 UTD_HOVED_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING - KURSNIVÅ)
Program navn .....: utd_hoved_kurs.pls
Skrevet av .....: JoJ
Dato .....: 18.06.1999
Versjon .....: 1.0
Programmets funksjon .: Legger behandlede records ut i hovedtabellen for
                        kurs: H_UTD_KURS.

Programmet kaller ....:
Programmet kalles fra : utd_fdel.pls
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

7.7.7 UTD_AVGANG_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING - KURSNIVÅ)
Program navn .....: utd_avgang_kurs.pls
Skrevet av .....: JoJ
Dato .....: 05.07.1999
Versjon .....: 1.0
Programmets funksjon .: Legger record inn i AVG_UTD_KURS. Tabellen inne-
                        holder en record for hvert avsluttede kurstilfelle,
                        men bare kurs som er avsluttet det siste året,
                        eldre avganger fjernes automatisk i prosedyren
                        UTD OPPDATER. Kun ett samtidig tilfelle av
                        kombinasjonen FNRxNUSxHOVED er lovlig i tabellen.

Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

7.7.8 UTD_AVSLUTT_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING - KURSNIVÅ)
Program navn .....: utd_avslutt_kurs.pls
Skrevet av .....: JoJ
Dato .....: 15.06.1999
Versjon .....: 1.0
Programmets funksjon .: Kurs avsluttes ett år etter siste "livstegn" fra
                        igangværendefila.

Programmet kaller ....:
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
*****/
```

7.7.9 UTD_STATUS_KURS.PLS

/*****

Prosjekt: FD-Trygd (IGANGV. UTDANNING - KURSNIVÅ)
 Program navn: utd_status_kurs.pls
 Skrevet av: JoJ
 Dato: 31.05.1999
 Versjon: 1.0
 Programmets funksjon .: Rutine som bestemmer status for record fra igang-

værende- og avsluttafila. Status settes bl.a ved å sjekke om ny_record finnes i databasen fra før av, og om det er dato-konsistens mellom ny og gammel record. Vi kobler m.a.o. loadfilene mot hovedtabell og evt. mot avgangstabell. Avgangstabellen inneholder alle kodete avganger det siste året. Basert på data fra loadtabellen og hjelpetabellene i basen danner vi forløp.

Koblingsnøkkelen mot avgangstabell og hovedtabell er: FNRxNUSxHOVED.

Vi får da seks mulige kombinasjoner for match mellom loadtabellen TMP_UTD og hjelpetabellene AVG_UTD_KURS og H_UTD_KURS. Statuskoden STATUS1 viser hvilken match som oppsto.

STATUS1	Igang (n)	Avsl. (n+1)	Igang (n+1)	Avsl. (n+2)	gml. STATUS1	Match hoved	Match avg.
1			X			0	0
2				X		0	0
3	X	X			1	1	
4	X		X		1	1	
5		X	X		2	0	1
6		X		X	2	0	1

ENDRKODE inneholder en akkumulert tallkode som viser hvilke internvariable som har endret seg.

ENDRKODE : Variabel som er endret

- 0 : Ingen endring i noen internvariabler
- 1 : KLTRINN er endret
- 2 : HELDEL er endret
- 4 : VOKSEN er endret
- 8 : STUDRET er endret
- 16 : LAERLING er endret
- 32 : AMO er endret

ENDRKODE = 13 betyr da at KLTRINN, VOKSEN og STUDRET (1 + 4 + 8 = 13) har endret verdi siden forrige registrering.

Hver record får i tillegg en egen statuskode (STATUS2) som forteller mer detaljert hvilken kombinasjon som oppsto og hva som er lagt til grunn

for hvordan den er behandlet senere.

For å oppdatere tabellene brukes DELETE og INSERT i kombinasjon da dette har vist seg å være mer effektivt enn UPDATE med det RAID-systemet (RAID5) SSB har valgt.

```
Programmet kaller ....:
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

7.7.10 UTD_FORDEL_KURS.PLS

```
*****/
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING - KURSNIVÅ)
Program navn .....: utd_fordel_kurs.pls
Skrevet av .....: JoJ
Dato .....: 01.06.1999
Versjon .....: 1.0
Programmets funksjon .: Koder og fordeler records ut til de ulike
                        prosedyrene for innlegging i tabellene. Ulike
                        status som er satt underveis under prosesseringen
                        av recorden avgjør i hvilken tabell recorden til
                        slutt ender.
Filer inn .....:
Filer ut .....:
Endret når .....:
Endret av .....:
Grunn til endring ....:
*****/
```

7.7.11 UTD_OPPDATER_KURS.PLS

```
*****/
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING -KURSNIVÅ)
Program navn .....: utd_oppdater_kurs.pls
Skrevet av .....: JoJ
Dato .....: 31.05.1999
Versjon .....: 1.0
Programmets funksjon .: Dette er hovedprosedyren som kaller opp de ulike
                        sub-rutinene under prosessering av årsfilene.
                        Prosedyren styrer og kaller opp prosedyrer for
                        innlegging av data i hovedtabell, forløpstabell og
                        andre temporære hjelpetabeller.
Programmet kaller ....:
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

7.7.12 UTD_RYDD_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING - KURSNIVÅ)
Program navn .....: utd_rydd_kurs.pls
Skrevet av .....: JoJ
Dato .....: 13.07.1999
Versjon .....: 1.0
Programmets funksjon .: Alle kurs som er avsluttet før 199201 fjernes fra
                        databasen (hoved-, og forløpstabell). Opplysninger
                        om slettede records logges i en egen fil på UNIX:
                        /u01/app/oracle/admin/TFH3/ora_out/
                        utd_kurs_log_ubrukte.txt

Programmet kaller ....:
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

7.7.13 UTD_KURS_BACK.SQL (backup-rutine)

Det er spesielt viktig å merke seg at batch-rutinene som legger ett og ett årssett med utdanningsdata i databasen (danner forløp), bygger på / jobber mot forrige års data (resultatet av forrige års kjøring). Det vil igjen si at dersom en har lagt inn n antall årsett med data og en batch-rutine bryter/krasjer, må ALLE data legges inn på nytt. For å unngå dette er det utarbeidet en egen backup-rutine.

Denne kjøres med skriptet

\$STRYGDFOB/prog/ORACLE/utdann/igangv/kurs/BACKUP/utd_kurs_back.sql.

Kjør rutinen etter innlegging av hvert årssett når du er sikker på at forrige kjøring gikk bra.

7.7.14 UTD_KURS_RECV.SQL (recovery-rutine)

Dersom innlegging av nye data forårsaker feil som stopper rutinen under kjøring, eller på annen måte ødelegger tidligere innlagte data, kan gamle data hentes tilbake ved hjelp av denne rutinen. Det forutsettes at rutinen UTD_KURS_BACK.SQL er kjørt FØR siste innleggingsforsøk. Rutinen henter da tilbake alle data tom. forrige årssett. Rutinen kjøres med skriptet

\$STRYGDFOB/prog/ORACLE/utdann/igangv/kurs/BACKUP/utd_kurs_recv.sql.

7.8 Programbeskrivelse av databaserutiner på viktigste-kursnivå (Oracle)

7.8.1 Innlastingsrekkefølge

For at korekt forløp skal dannes er det viktig å laste inn data fra forløpstabellen for kurs (F_UTD_KURS) ved rett tidspunkt. Dvs. at når en ny årgang er dannet på kursnivå kjøres rutinene for å danne forløp på viktigste kursnivå for denne årgangen **før** neste årgang for kurs kjøres inn.

Kjøringene blir da som beskrevet under:

1992 : Først igangv.91, så avsl.92, deretter igangv.92 og til slutt rutiner for viktigste kursnivå
1993 : Først avsl.93, deretter igangv.93 og til slutt rutiner for viktigste kursnivå
1994 : Først avsl.94 deretter igangv.94 og til slutt rutiner for viktigste kursnivå
.
.
1998 : Først avsl.98, deretter igangv.98 og til slutt rutiner for viktigste kursnivå

7.8.2 UTD_OP_VIKTIGSTE_KURS.SQL (Hovedprog. for oppd. på viktigste-kursnivå)

Rutinen fungerer som hovedprogram for oppdatering av viktigste-kurs for igangværende utdanning **for 1992/01** og kaller opp ulike sub-program for danning av forløpsdata. Data lastes inn fra forløpstabellen **F_UTD_KURS** for ett og ett år som beskrevet i punktet over. Hovedprogrammet prosesserer hvert årssett av data fortløpende. Dette gjøres med rutinen **\$STRYGDFOB/prog/ORACLE/utdann/igangv/kurs/OPPDATER/utd_op_kurs.sql**.

Alle personer i nyeste datagrunnlag, dvs. personer i tabellen **F_UTD_KURS**, sjekkes mot forrige års datagrunnlag (innholdet i hovedtabellen **H_UTD_VIKTIGSTE**), record for record. Ut i fra bestemte regler, som angitt i 5.6, avgjøres det om det skal kodes en *tilgang*, *avgang*, *overgang* eller *endring* for hver enkelt record som prosesseres. *Endring* betyr endret verdi i en eller flere av internvariablene siden forrige registrering, og *overgang* betyr avgang fra en hovedgruppe med tilgang til en annen hovedgruppe, uten opphold.

Kildekoden til dette programmet er lagt inn i dokumentasjonen for at det skal være lettere å se rekkefølgen rutinene kompiles i.

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING - VIKTIGSTE-KURSNIVÅ)
Program navn .....: utd_op_viktigste_kurs.sql
Skrevet av .....: JoJ
Dato .....: 19.10.1999
Versjon .....: 1.0
Programmets funksjon ..: Batch-jobb som kaller alle oppdateringsrutiner for
                          igangværende utdanning på viktigste-kursnivå.

                          NB! Før jobben startes bør det tas sikkerhets kopi
                          av alle berørte databasetabeller (Benytt
                          rutinen UTD_VIKTIGSTE_KURS_BACK.SQL).

Filer inn .....: Alle oppdateringsrutiner på viktigste-kursnivå for
                          igangværende UTDANNING.
Filer ut .....:
Endret når .....:
Endret av .....:
*****/

/* Denne sender meldinger til en egen logg-fil */
```

```
SPOOL UTD_OP_VIKTIGSTE_KURS
```

```
/* Kompilerer alle PL/SQL-prosedyrer som skal benyttes */
```

```
@to_bin.pls/  
@utd_forlop_viktigste_kurs.pls /  
@utd_hoved_viktigste_kurs.pls /  
@utd_avgang_viktigste_kurs.pls /  
@utd_avslutt_viktigste_kurs.pls /  
@utd_status_viktigste_kurs.pls /  
@utd_fordel_viktigste_kurs.pls /  
@utd_oppdater_viktigste_kurs.pls /  
COMMIT;
```

```
/* Setter på en timer for å se hvor lang tid jobben tar */  
SET TIMING ON;  
SET TIME ON;
```

```
/* Her kalles styrerutine for å lese inn fra riktig årsfil.  
Oppgi hvilken årgang (på formatet YYYY) som skal kjøres. */  
EXECUTE utd_oppdater_viktigste(&&yyyy);  
COMMIT;
```

```
-- Til slutt slettes alle prosedyrer  
DROP PROCEDURE utd_avslutt_viktigste_kurs;  
DROP PROCEDURE utd_fordel_viktigste_kurs;  
DROP PROCEDURE utd_forlop_viktigste_kurs;  
DROP PROCEDURE utd_hoved_viktigste_kurs;  
DROP PROCEDURE utd_avgang_viktigste_kurs;  
DROP PROCEDURE utd_oppdater_viktigste_kurs;  
DROP PROCEDURE utd_bestem_status_for_viktigste_kurs;  
COMMIT;
```

```
/* Lukker spool-meldingsfilen */  
SPOOL OFF
```

```
EXIT;
```

7.8.3 UTD_FORLOP_VIKTIGSTE_KURS.PLS

```
/******  
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING VIKTIGSTE-KURSNIVÅ)  
Program navn .....: utd_forlop_viktigste.pls  
Skrevet av .....: JoJ  
Dato .....: 28.09.1999  
Versjon .....: 1.0  
Programmets funksjon ..: Legger behandlede records ut i forløpstabellen for  
viktigste kurs: F_UTD_VIKTIGSTE.  
  
Programmet kaller ..: .  
Programmet kalles fra : utd_fdel_viktigste.pls  
Filer inn .....: .  
Filer ut .....: .  
Endret når .....: DD.MM.ÅÅ  
Endret av .....: .  
Grunn til endring ..: .  
*****/
```


7.8.4 UTD_HOVED_VIKTIGSTE_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING VIKTIGSTE-KURSNIVÅ)
Program navn .....: utd_hoved_viktigste.pls
Skrevet av .....: JoJ
Dato .....: 28.09.1999
Versjon .....: 1.0
Programmets funksjon .: Legger behandlede records ut i hovedtabellen for
                        viktigste kurs: H_UTD_VIKTIGSTE.

Programmet kaller ..::
Programmet kalles fra : utd_fdel_viktigste.pls
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ..::
*****/
```

7.8.5 UTD_AVGANG_VIKTIGSTE_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING VIKTIGSTE-KURSNIVÅ)
Program navn .....: utd_avgang_viktigste.pls
Skrevet av .....: JoJ
Dato .....: 28.09.1999
Versjon .....: 1.0
Programmets funksjon .: Legger behandlede records ut i avgangstabellen for
                        viktigste kurs: AVG_UTD_VIKTIGSTE.

Programmet kaller ..::
Programmet kalles fra : utd_fdel_viktigste.pls
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ..::
*****/
```

7.8.6 UTD_AVSLUTT_VIKTIGSTE_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING VIKTIGSTE-KURSNIVÅ)
Program navn .....: utd_avslutt_viktigste_kurs.pls
Skrevet av .....: JoJ
Dato .....: 13.10.1999
Versjon .....: 1.0
Programmets funksjon .: Avslutter midlertidige avganger ett år etter siste
                        "livstegn" fra forløpstabellen for kurs
                        (F_UTD_KURS).

Programmet kaller ..::
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ..::
*****/
```

7.8.7 UTD_STATUS_VIKTIGSTE_KURS.PLS

/*****

Prosjekt: FD-Trygd (IGANGV. UTDANNING - VIKTIGSTE KURSNIVÅ)
Program navn: utd_status_viktigste_kurs.pls
Skrevet av: JoJ
Dato: 01.09.1999
Versjon: 1.0
Programmets funksjon .: Rutine som bestemmer status for record fra forløps-
tabellen for kurs (F_UTD_KURS). Status settes bl.a
ved å sjekke om ny_record finnes i databasen fra
før av, og om det er dato-konsistens mellom ny og
gammel record. Vi kobler m.a.o. F_UTD_KURS mot
H_UTD_VIKTIGSTE og evt. mot AVG_UTD_VIKTIGSTE.

Basert på data fra F_UTD_KURS, H_UTD_VIKTIGSTE,
H_UTD_VIKTIGSTE og AVG_UTD_VIKTIGSTE danner vi
forløp på viktigste-kursnivå.

FNR er koblingsnøkkel mot avgangs- og hovedtabell.

Hver record får en egen statuskode (STATUS7) som
forteller mer detaljert hvilken kombinasjon som
oppsto og hva som er lagt til grunn for hvordan
den ble behandlet senere.

ENDRKODE inneholder en akkumulert tallkode som
viser hvilke internvariable som evt. har endret
seg siden forrige registrerte record.

ENDRKODE : Variabel som er endret

0 : Ingen endring i noen internvariabler
1 : KLTRINN er endret
2 : HELDEL er endret
4 : VOKSEN er endret
8 : STUDRET er endret
16 : LAERLING er endret
32 : AMO er endret
64 : NUS er endret
128 : NUS1 er endret
256 : NUS2 er endret
512 : NUS3 er endret
1024 : NUS4 er endret
2048 : UTFALL, UTFDATO og KURSAVG er satt
4096 : UTFALL, UTFDATO og KURSAVG er NULL'et

For tilfeller der det bli kodet to eller flere
endringsrecord i en og samme operasjon, uten at det
nødvendigvis har skjedd en endring, har vi to
spesielle koder: 2048 og 4096. ENDRKODE settes lik
2048 for å markere den første og 4096 for å markere
den neste.

ENDRKODE = 13 betyr f.eks KLTRINN, VOKSEN og
STUDRET (1 + 4 + 8 = 13) har endret verdi siden
siste registrering.

STATUS6 forteller om kurset har fått sine avgangsopplysninger, og hvordan disse er kodet:

STATUS6 : Forklaring

-
- 0 : Record inneholder ingen avgangsopplysninger for kurset
 - 1 : Avgangsopplysningene er kodet ved normal avgang
 - 2 : Record i forløpstabellen er oppdatert med avgangsopplysninger i ettertid (for kurs som har vært viktigste kurs i en periode, men blitt avsluttet før avgangsopplysningene var tilgjengelige for kurset).
 - 3 : Avgang med lik dato som tilgangen. Tilganger med denne koden har fått overført avgangsopplysningene fra avgangsrecorden som er slettet (vi har hatt en tilgang og endring med lik dato).
 - 4 : Avgang med lik dato som gammel endring. Dersom avgangen senere gjøres om til en ny endring, så forkastes denne samtidig som avgangsopplysningene blir overført til gammel endring.

STATUS9 forteller om ny_record er viktigere enn gammel_record:

STATUS9 : Forklaring

-
- 0 : Det er ikke match mellom ny_record og record i hoved_tabell
 - 1 : Den nye recorden er viktigere
 - 2 : Den nye recorden er en "viktigere" avgang eller endring (samme NUS)
 - 3 : Den nye recorden er en IKKE viktigere endring, men kodes som viktigere likevel (samme NUS)
 - 9 : Recorden er IKKE viktigere

KURSTOM inneholder avgangsopplysninger fra F_UTD_KURS som viser når avgangen intreffer for en tilgang, avgang eller endring. Opplysningen benyttes for å sikre at vi ikke starter opp et tilfelle som vil avsluttes før tidligere dateringer for tilfellet i databasen.

For å oppdatere tabellene brukes DELETE og INSERT i kombinasjon da dette har vist seg å være mer effektivt enn UPDATE med det RAID-systemet (RAID5) SSB har valgt.

```
Programmet kaller .....:
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring .....:
*****/
```

7.8.8 UTD_FORDEL_VIKTIGSTE_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING VIKTIGSTE-KURSNIVÅ)
Program navn .....: utd_fordel_viktigste_kurs.pls
Skrevet av .....: JoJ
Dato .....: 16.07.1999
Versjon .....: 1.0
Programmets funksjon .: Koder og fordeler records ut til de ulike
                        prosedyrene for innlegging i tabeller for viktigste
                        kurs.

                        Ulike status som er satt underveis i prosessering
                        av record avgjør i hvilken tabell recorden til
                        slutt ender.

Filer inn .....:
Filer ut .....:
Endret når .....:
Endret av .....:
Grunn til endring ....:
*****/
```

7.8.9 UTD_OPPDATER_VIKTIGSTE_KURS.PLS

```

/*****
Prosjekt .....: FD-Trygd (IGANGV. UTDANNING VIKTIGSTE-KURSNIVÅ)
Program navn .....: utd_oppdater_viktigste_kurs.pls
Skrevet av .....: JoJ
Dato .....: 08.07.1999
Versjon .....: 1.0
Programmets funksjon .: Dette er hovedprosedyren som kaller opp de ulike
                        sub-rutinene under prosesseringen av
                        forløpstabellen for kurs (F_UTD_KURS).
                        Prosedyren styrer og kaller opp prosedyrer for
                        innlegging av data i forløpstabellen for viktigste
                        kurs (F_UTD_VIKTIGSTE).

Programmet kaller ....:
Programmet kalles fra :
Filer inn .....:
Filer ut .....:
Endret når .....: DD.MM.ÅÅ
Endret av .....:
Grunn til endring ....:
*****/
```

7.8.10 UTD_VIKTIGSTE_KURS_BACK.SQL (backup-rutine)

Det er spesielt viktig å merke seg at batch-rutinene som legger ett og ett årssett med utdanningsdata i databasen (danner forløp), bygger på / jobber mot, forrige års data (resultatet av forrige års kjøring). Det vil igjen si at dersom en har lagt inn n antall årssett med data og en batch-rutine bryter/krasjer, må ALLE data legges inn på nytt. For å unngå dette er det utarbeidet en egen backup-rutine.

Denne kjøres med skriptet

\$STRYGDFOB/prog/ORACLE/utdann/igangv/viktigste/BACKUP/utd_viktigste_kurs_back.sql.

Kjør rutinen etter innlegging av hvert årssett når du er sikker på at forrige kjøring gikk bra.

7.8.11 UTD_VIKTIGSTE_KURS_RECV.SQL (recovery-rutine)

Dersom innlegging av nye data forårsaker feil som stopper rutinen under kjøring, eller på annen måte ødelegger tidligere innlagte data, kan gamle data hentes tilbake ved hjelp av denne rutinen. Det forutsettes at rutinen UTD_VIKTIGSTE_KURS_BACK.SQL er kjørt FØR siste innleggingsforsøk.

Rutinen henter da tilbake alle data tom. forrige årssett. Rutinen kjøres med skriptet

\$STRYGDFOB/prog/ORACLE/utdann/igangv/viktigste/BACKUP/utd_viktigste_kurs_recv.sql.

7.8.12 TO_BIN.PLS (generell hjelperutine)

/*****

Prosjekt: FD-Trygd (GENERELLE RUTINER)
Program navn: to_bin.pls
Skrevet av: JoJ
Dato: 01.10.1999
Versjon: 1.0
Programmets funksjon .: Konverterer n av NUMBER-datatype, til et BINÆRT-
format av VARCHAR2-datatype.

Eks:
SELECT to_bin(22) "Binær verdi"
FROM DUAL;

Binær verdi

10110

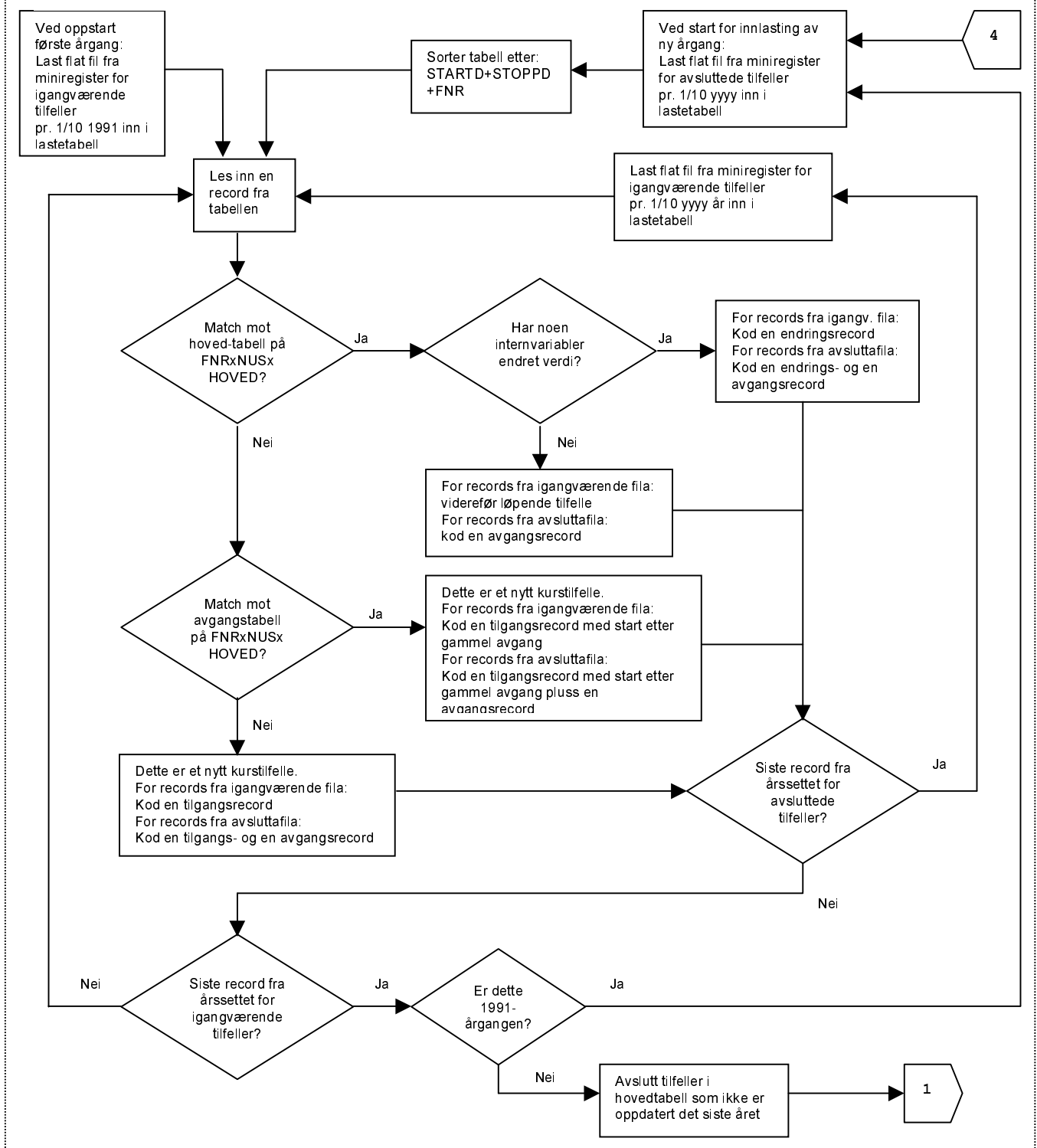
Programmet kaller:
Programmet kalles fra :
Filer inn:
Filer ut:
Endret når: DD.MM.ÅÅ
Endret av:
Grunn til endring:

*****/

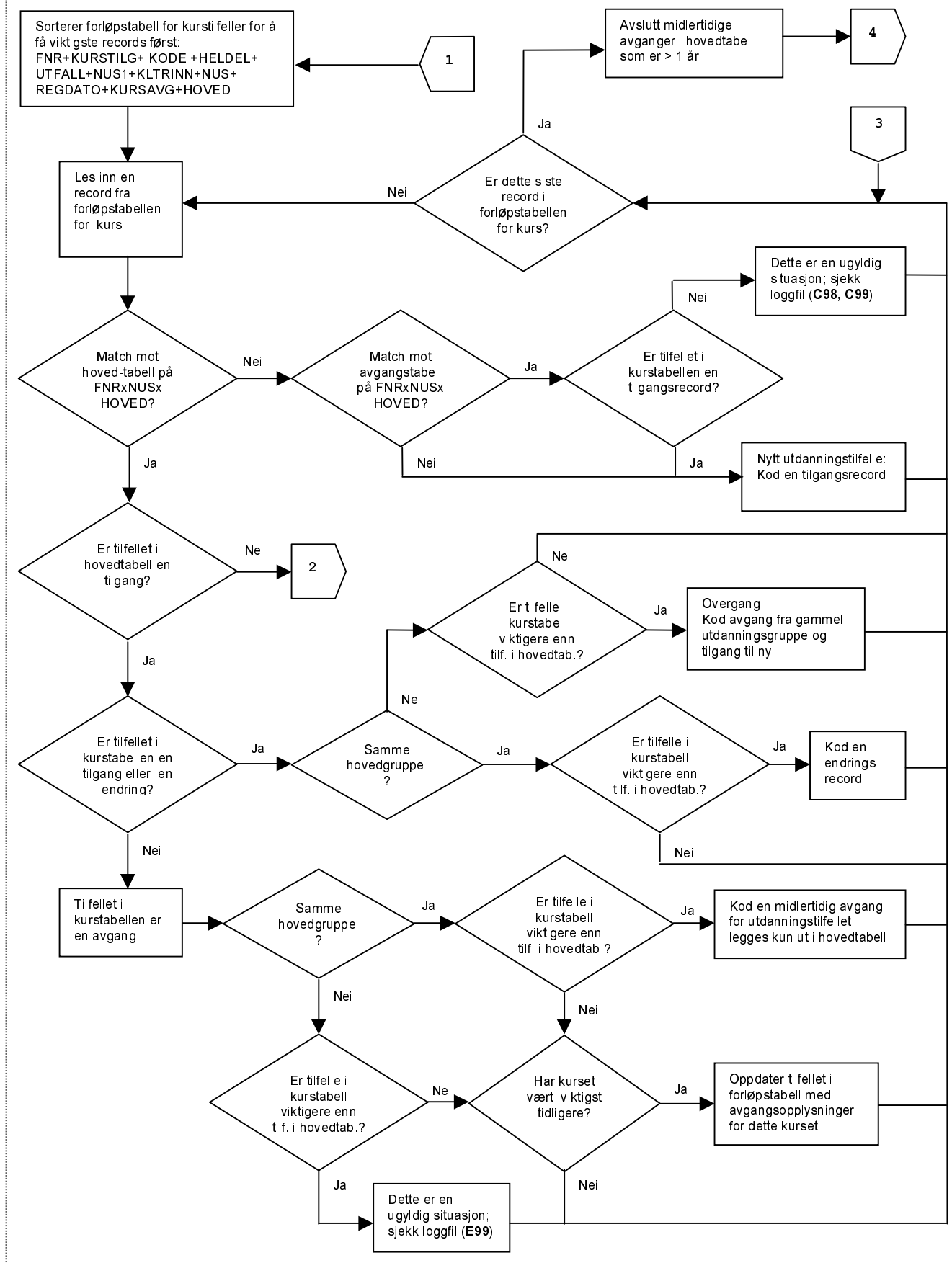
7.9 Flytdiagram for PL/SQL-kode til igangværende utdanning

FD-TRYGD	Diagramansvarlig: JoJ	Dato: 25.02.2000
Forenklet flytdiagram for PL/SQL-kode til IGANGVÆRENDE UTDANNING		

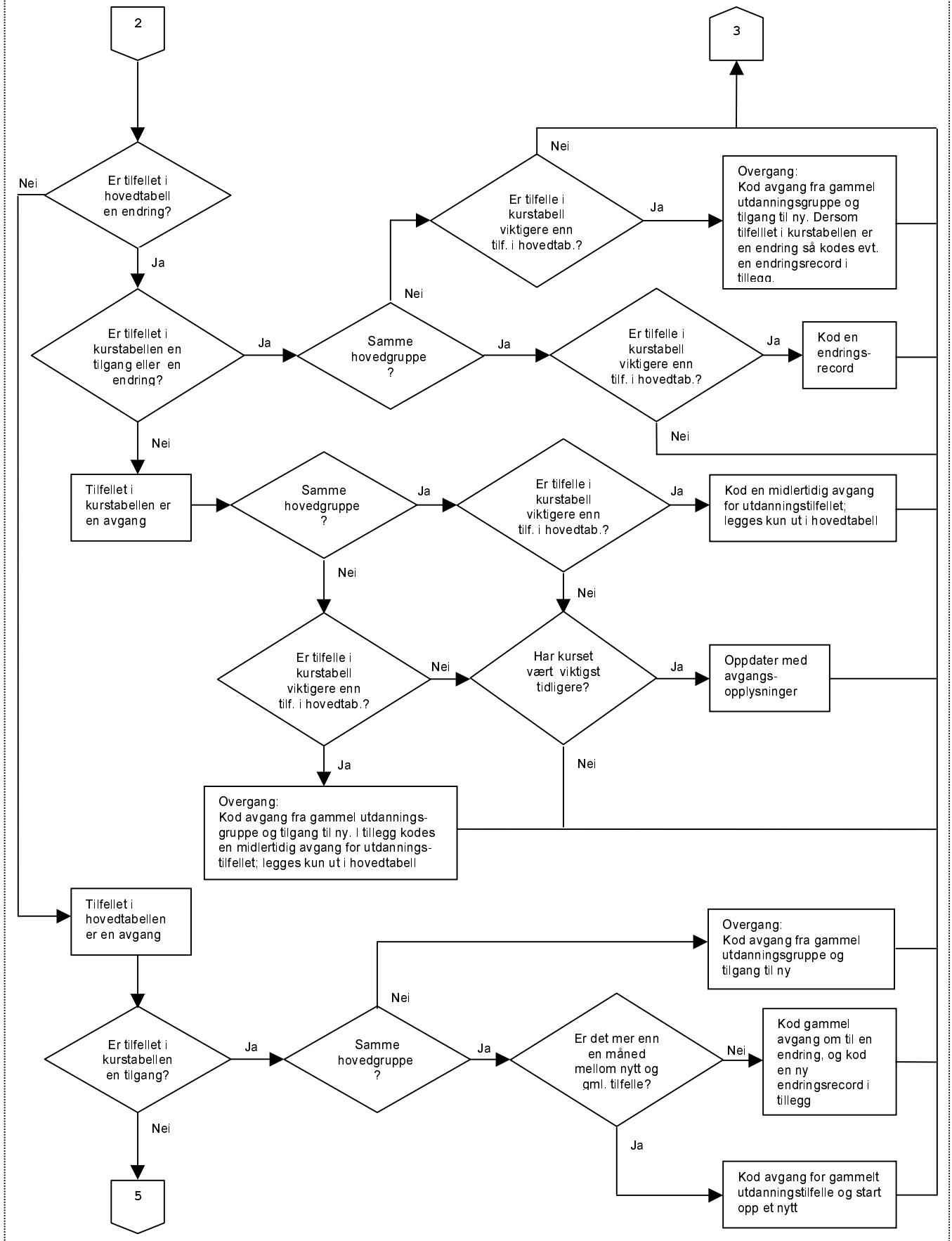
Rutiner for å danne forløp på kursnivå



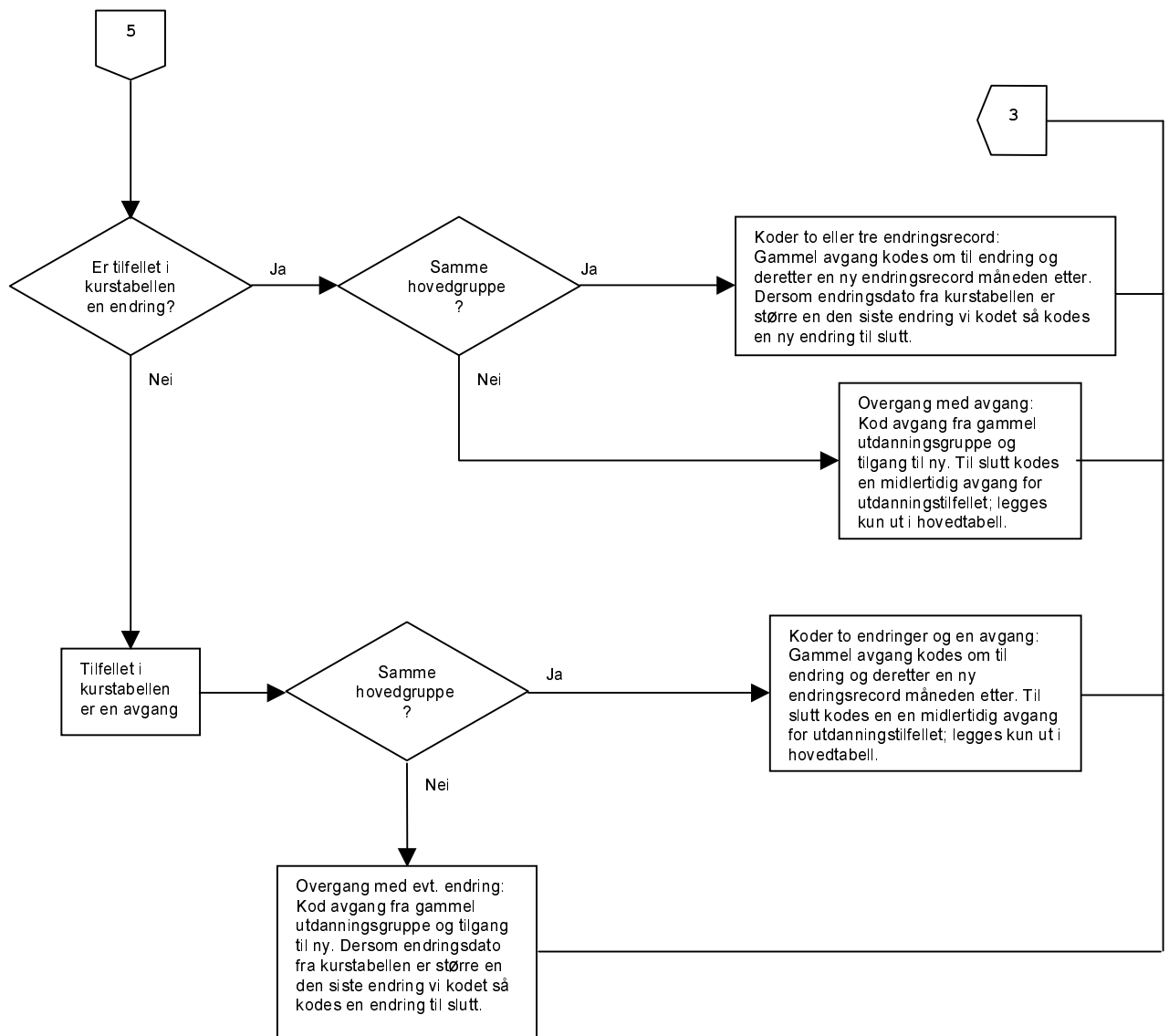
Rutiner for å danne forløp på viktigste-kursnivå (1 av 3)



Rutiner for å danne forløp på viktigste-kursnivå (2 av 3)



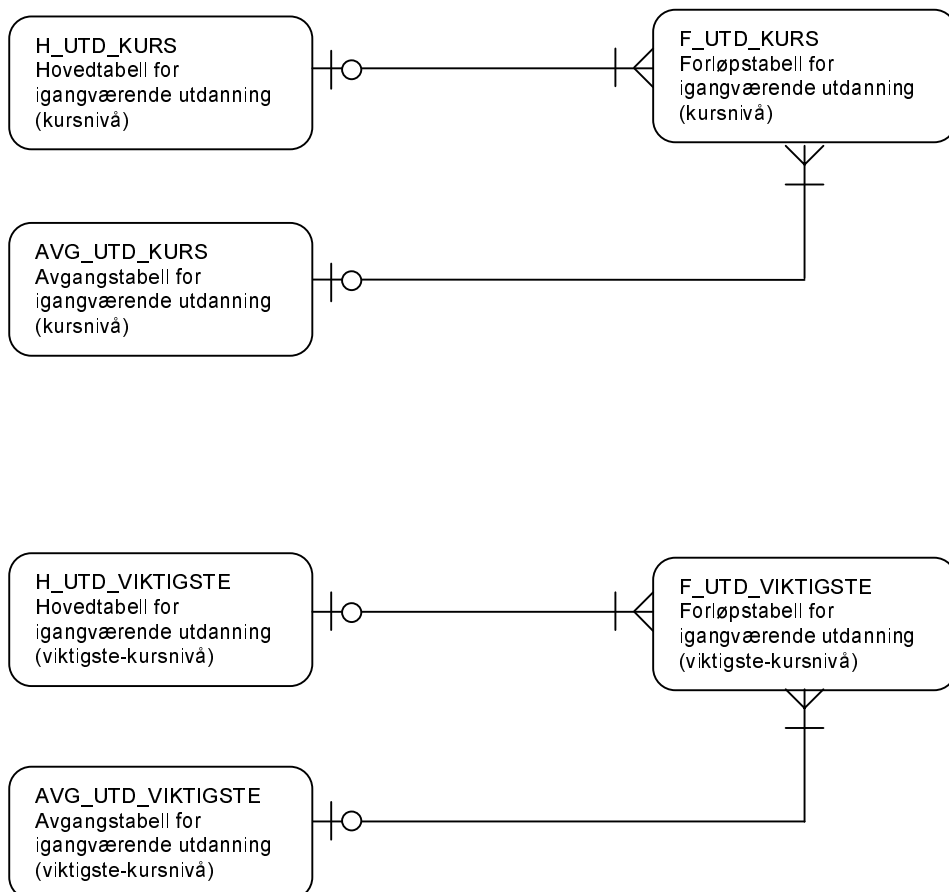
Rutiner for å danne forløp på viktigste-kursnivå (3 av 3)



7.10 Datamodell for igangværende utdanning

FD-TRYGD	Diagramansvarlig: JoJ	Dato: 25.02.2000
Datamodell - IGANGVÆRENDE UTDANNING		

Miniregisterene for IGANGVÆRENDE UTDANNING (IGANGV. og AVSLUTTA tilfeller) kjøres mot sine respektive hoved- og avgangstabeller (som inneholder status for nyeste datagrunnlag) for oppdatering og danning av forløp.



7.11 Modellspesifikasjoner for igangværende utdanning

Alle definisjoner av Oracle databasetabeller, med tilhørende kommentarer og kodelister, ligger lagret i Oracle Repository. Oracle Designer 2000 er benyttet som modelleringsverktøy.

I databasen er det også etablert backup-tabeller som ikke er beskrevet nærmere i denne rapporten da disse har samme struktur som sine respektive originaltabeller:

F_UTD_KURS_BACK
H_UTD_KURS_BACK
AVG_UTD_KURS_BACK

F_UTD_VIKTIGSTE_BACK
H_UTD_VIKTIGSTE_BACK
AVG_UTD_VIKTIGSTE_BACK

7.11.1 Load- og temporærtabell for innlasting av igangværende utdanning (kursnivå)

Tabellnavn: L_UTD (loadtabell)
TMP_UTD (sortert loadtabell)

Kommentarer til loadtabellen: L_UTD:

Årsfiler for avslutta utdanning, som er avsluttede kurs i løpet av ett år (YYYY-10 - YYYY09), lastes inn i en loadtabell (mottakstabell) før bearbeiding i databasen.

Kommentarer til sortert loadtabell: TMP_UTD:

Årsfilene for avslutta utdanning er bruttofiler og må sorteres før oppdateringsrutinene kan kjøres for å sikre at records blir prosessert i riktig rekkefølge. Tabellen er sortert etter: STARTD + STOPPD + FNR. Igangværendefilene er nettofiler (kun en record pr. tilfelle) og trenger derfor ingen sortering (skal lastes rett inn i TMP_UTD). Årsfiler for igangværende utdanning inneholder ca. 400.000 - 420.000 records, mens avsluttafilene inneholder ca. 300.000 - 400.000 records.

Generelle kommentarer:

Strukturen (alle kolonner) i tabellene L_UTD og TMP_UTD er identisk.

Variabelbeskrivelse for tabellene: L_UTD og TMP_UTD

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Fødselsnummer FNR VARCHAR2 11	Også records med ugyldige fødselsnumre tas med i FD-Trygd.
Hovedgruppevariabel HOVED VARCHAR2 1	Settes under danning av miniregisterene og angir eleven/studentens utdanningsnivå, og skiller de som studerer ved høyskole og universitet. Gyldige koder: 1 = Grunnskole 2 = Videregående skole 3 = Høyskoler 4 = Universitet 9 = Uoppgitt
Utdannings art NUS VARCHAR2 6	Disse kodene følger "Standard for utdanningsgruppering" . Dette er hele NUS-koden (6 siffer) For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.
Utdannings art NUS1 VARCHAR2 1	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder første siffer for NUS-koden. For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS1 som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.
Utdannings art NUS2 VARCHAR2 2	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de to første siffer for NUS-koden. For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS2 som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.
Utdannings art NUS3 VARCHAR2 3	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de tre første siffer for NUS-koden. For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS3 som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Utdannings art NUS4 VARCHAR2 4	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de fire første siffer for NUS-koden. For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS4 som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.
Kursets startdato STARTD NUMBER 6	Format: YYYYMM
Kursets stoppdato STOPPD NUMBER 6	Format: YYYYMM
Klassetrinn KLTRINN NUMBER 2	For studenter ved universitet blir klassetrinn justert ut fra studiemål og antatt progresjon. Ved videregående- og høyskole så blir klassetrinn satt av skolen med utgangspunkt i kurset. Hvis studenten skifter mellom høyskole og universitet, eller endrer studiemål kan klassetrinn være mindre representativt for den faktiske progresjon.
Heltid/deltid HELDEL VARCHAR2 1	Denne variabelen blir i utgangspunktet satt av skolen ut fra om kurset er ment å utgjøre et fulltidsstudium eller ikke. Gyldige koder: 1 = Heltid 2 = Deltid
Voksenopplæring VOKSEN VARCHAR2 1	Viser om kurset er en utdanning som går inn under lov om voksenopplæring. Gyldige koder: 1 = Voksenopplæring 2 = Ikke voksenopplæring
Studieretningskode STUDRET VARCHAR2 2	Opplysninger om hvilken studieretning på videregående nivå (linje) kurset tilhører.
Lærling LÆRLING VARCHAR2 1	De som er igang eller har avsluttet fagprøve (lærlinger): Gyldige koder: 0 = Ikke lærling 1 = Lærling
AMO-kurs AMO VARCHAR2 1	Viser om kurset er et AMO-kurs. Gyldige koder: 0 = Ikke AMO-kurs 1 = AMO-kurs
Utfallskode UTFALL VARCHAR2 1	Viser om personen har fullført eller avbrutt kurset. Gyldige koder: 1 = Er kodet som avbrutt i databasen 2 = Avbrutt 8 = Fullført

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Miniregisterkontroll - datokontroll MRK_DL VARCHAR2 1	Kode som viser hvilken startdato som benyttes når opplysningene hentes fra igangværendefilene: Gyldige koder: 1 = Elevens startdato er brukt. 3 = Kursets startdato er brukt. 5 = Uoppgitt settes til '19YY08' for igangværendefil (YY: år). NULL = STARTD er hentet fra avsluttafil.
Miniregisterkontroll - datokontroll MRK_DL1 VARCHAR2 1	Kode som viser hvilke start- og stoppdatoer som er benyttet når datoer hentes fra avsluttafilene: Gyldige koder: 1 = Elevens start- og stoppdato er brukt. 2 = Elevens startdato og kursets stoppdato er brukt. 3 = Kursets startdato og elevens stoppdato er brukt. 4 = Kursets start- og stoppdato er brukt. 5 = Start mangler, mens elevens stoppdato er brukt. 6 = Start mangler, mens kursets stoppdato er brukt (univ.stud.) 7 = Startdato er uoppgitt, mens elevens stoppdato er brukt. 8 = Startdato er uoppgitt, mens kursets stoppdato er brukt. 9 = Startdato >= stoppdato, mens elevens stoppdato er brukt. A = Startdato >= stoppdato, mens kursets stoppdato er brukt. NULL = Hvis MRK_DL1 er blank, er alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende, eller avsluttet i databasen.
Miniregisterkontroll - startdatokontroll på avslutta MRK_DL2 VARCHAR2 1	På avsluttafilene (år t) rettes ugyldige startdatoer ved kobling mot igangværende fil (år t-1) eller vårsemesterfila (år t). Kodene viser hvilke regler som er brukt: For universitetsstudenter: 1 = Ligger på igangværendefila med samme kurs, bruker startdato fra igangværende. 2 = Ligger på vårsemesterfila med samme kurs, startdato settes til start i vårsemesteret. 3 = Ligger på igangværendefila med ett annet kurs, setter startdato '19(t-1)09'. 4 = For resten settes startdato til start i det semesteret kurset er avsluttet. For andre som har ugyldige startdatoer: 7 = Ligger på igangværendefila med samme kurs, bruker startdato fra igangværende. 8 = Ligger på igangværendefila med ett annet kurs, setter startdato '19(t-1)08'. 9 = Personen ligger ikke på igangværendefila, setter startdato '19(t-1)08'. A = Avslutning av ordinær grunnskole; startdato settes lik stoppdato. 0 = Har ikke rettet startdatoer NULL = Hvis MRK_DL2 er blank, er alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende, eller avsluttet i databasen.

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Miniregisterkontroll - dublett kontroll MRK_DL3 VARCHAR2 1	Dersom like kurs er overlappende i tid på avsluttafilene, slår vi records sammen for å unngå slike dubletter. Vi merker hvor vi henter ulike variable fra etter følgende regler: 0 = Alle variable er hentet fra samme record (ikke dubletter, dublettene forlenger ikke kursperioden eller dubletter har ikke overlappende perioder). 1 = Startdato, men ikke stoppdato, hentes fra den recorden hvor andre opplysninger hentes fra. 2 = Stoppdato, men ikke startdato, hentes fra den recorden hvor andre opplysninger hentes fra. 3 = Start- og stoppdato hentes fra en record mens andre opplysninger hentes fra en annen record. 4 = Startdato, stoppdato og andre opplysninger hentes fra tre ulike records. NULL = Hvis MRK_DL3 er blank, er alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende, eller avsluttet i databasen.
Miniregisterkontroll - fødselsnummer MRK_FNR VARCHAR2 1	Merking av fødselsnummer. Gyldige koder: 0 = Gyldig fødselsnummer 1 = Ugyldig fødselsnummer, men gyldig d-nummer 2 = Ugyldig fødselsnummer bestående av blankt personnummer 3 = Ugyldig fødselsnummer som ikke omfattes av kode 1 eller 2
Årgang AARGANG NUMBER 4	Viser hvilken årsfil record kommer fra. Format: YYYY

7.11.2 Forløps-, hoved- og avgangstabell for igangværende utdanning (kursnivå)

Tabellnavn: F_UTD_KURS (forløpstabell)
H_UTD_KURS (hovedtabell)
AVG_UTD_KURS (avgangstabell)

Kommentarer til forløpstabellen: F_UTD_KURS:

Inneholder forløp for igangværende utdanning på kursnivå og er generert ved hjelp av årsfiler for igangværende utdanning (pr. 01.10.YYYY), og avslutta utdanning som er avsluttede kurs i løpet av ett år (YYYY-10 - YYYY09). For hver igangværendefil som "kjøres" mot forløpsdatabasen, genereres det ca. 400.000 - 450.000 nye records i forløpstabellen (tilganger, endringer og avganger). Tilsvarende tall for avsluttafilene er ca. 500.000 nye records.

Kommentarer til hovedtabellen: H_UTD_KURS:

Dette er en hjelpetabell som inneholder siste datagrunnlag som er benyttet for å danne forløp i databasen. Inneholder kun en record pr. kurstillfelle (ett kurstillfelle er definert som FNR x NUS x HOVED) og blir liggende i tabellen til vi koder en avgang for kurset. Inneholder til enhver tid ca. 170.000 - 420.000 records.

Kommentarer til avgangstabellen: AVG_UTD_KURS:

Dette er en hjelpetabell som inneholder SISTE avgangstillfelle (kun en record pr. kurstillfelle; definert som FNR x NUS x HOVED) for hver person som er registrert i databasen. Avgangstillfelle betyr her avgang fra et kurs. Tabellen benyttes bl.a. for datokontroll ved danning av nye forløpsdata og for å kontrollere databasen i ettetid.

Generelle kommentarer:

Strukturen (alle kolonner) i tabellene F_UTD_KURS, H_UTD_KURS og AVG_UTD_KURS er identisk. Alle utdanningsdata «knyttet til» en hovedvariabel kalt KODE som sier noe om hendinger/endringer som er tilknyttet kurstillfellet.

STATUS2 sier noe om hvordan recorden er behandlet, og er delt inn i to grupper: A og B:

- A** : Records som kommer fra igangværendefila (tilganger, endringer og avganger)
- B** : Records som kommer fra avsluttafila (tilganger, endringer og avganger)

For mer utfyllende informasjon om hvordan de ulike variabler kodes, se kapittel 5.5.

Variabelbeskrivelse for tabellene: F_UTD_KURS, H_UTD_KURS og AVG_UTD_KURS

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Fødselsnummer FNR VARCHAR2 11	Også records med ugyldige fødselsnumre tas med i FD-Trygd.
Hovedgruppevariabel HOVED VARCHAR2 1	Settes under danning av miniregisterene og angir eleven/studentens utdanningsnivå, og skiller de som studerer ved høyskole og universitet. Gyldige koder: 1 = Grunnskole 2 = Videregående skole 3 = Høyskoler 4 = Universitet 9 = Uoppgitt
Hendelseskode KODE VARCHAR2 1	Settes i databasen og sier noe om hendelser/endringer tilknyttet kurset. Gyldige koder: 0 = Kurs avgang 1 = Kurs tilgang 2 = Kurs endring (i internvariabel) 3 = Kurs venstresensur (startet før 199201)
Registreringsdato REGDATO NUMBER 6	Regdato vil fungere som en siste oppdaterings-/endringsdato for hele recorden sett som en helhet, uavhengig om det er tilgang, avgang eller endring. Format: YYYYMM
Kursets tilgangsdato KURSTILG NUMBER 6	Settes ved kursstart og endres ikke. Format: YYYYMM
Kursets avgangsdato KURSAVG NUMBER 6	Settes når kurset avsluttes, blank (NULL) ellers. Format: YYYYMM
Utdannings art NUS VARCHAR2 6	Disse kodene følger "Standard for utdanningsgruppering" . Dette er hele NUS-koden (6 siffer) For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet.
Endringsdato for NUS NUSDATO NUMBER 6	Denne endrer seg ikke innenfor et kurs. Format: YYYYMM

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Utdannings art NUS1 VARCHAR2 1	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder første siffer for NUS-koden. For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS1 som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet.
Endringsdato for NUS1 NUS1DATO NUMBER 6	Denne endrer seg ikke innenfor et kurs. Format: YYYYMM
Utdannings art NUS2 VARCHAR2 2	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de to første siffer for NUS-koden. For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS2 som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet.
Endringsdato for NUS2 NUS2DATO NUMBER 6	Denne endrer seg ikke innenfor et kurs. Format: YYYYMM
Utdannings art NUS3 VARCHAR2 3	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de tre første siffer for NUS-koden. For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS3 som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet.
Endringsdato for NUS3 NUS3DATO NUMBER 6	Denne endrer seg ikke innenfor et kurs. Format: YYYYMM
Utdannings art NUS4 VARCHAR2 4	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de fire første siffer for NUS-koden. For forløpstabellen F_UTD_KURS hvor vi danner forløp på kursnivå (FNR x NUS) vil ikke denne variabelen endre verdi. NUS4 som blir satt ved tilgang holder seg mao. uendret gjennom hele forløpet.
Endringsdato for NUS4 NUS4DATO NUMBER 6	Denne endrer seg ikke innenfor et kurs. Format: YYYYMM
Klasstrinn KLTRINN NUMBER 2	For studenter ved universitet blir klasstrinn justert ut fra studiemål og antatt progresjon. Ved videregående- og høyskole blir klasstrinn satt av skolen med utgangspunkt i kurset. Hvis studenten skifter mellom høyskole og universitet, eller endrer studiemål kan klasstrinn være mindre representativt for den faktiske progresjon.
Endringsdato for klasstrinn KLTRDATO NUMBER 6	STARTD brukes til å datere endringer. Format: YYYYMM

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Heltid/deltid HELDEL VARCHAR2 1	Denne variabelen blir i utgangspunktet satt av skolen ut fra om kurset er ment å utgjøre et fulltidsstudium eller ikke. Gyldige koder: 1 = Heltid 2 = Deltid
Endringsdato for heltids/deltids utdanning HELDDATO NUMBER 6	STARTD brukes til å datere endringer. Format: YYYYMM
Voksenopplæring VOKSEN VARCHAR2 1	Viser om kurset er en utdanning som går inn under lov om voksenopplæring. Gyldige koder: 1 = Voksenopplæring 2 = Ikke voksenopplæring
Endringsdato for voksenopplæringskode VOKSDATO NUMBER 6	STARTD brukes til å datere endringer. Format: YYYYMM
Studieretningskode STUDRET VARCHAR2 2	Opplysninger om hvilken studieretning på videregående nivå (linje) kurset tilhører.
Endringsdato for studiekode STUDDATO NUMBER 6	STARTD brukes til å datere endringer. Format: YYYYMM
Lærling LAERLING VARCHAR2 1	De som er igang eller har avsluttet fagprøve (lærlinger): Gyldige koder: 0 = Ikke lærling 1 = Lærling
Endringsdato for lærlingskode LAERDATO NUMBER 6	STARTD brukes til å datere endringer. Format: YYYYMM
AMO-kurs AMO VARCHAR2 1	Viser om kurset er et AMO-kurs. Gyldige koder: 0 = Ikke AMO-kurs 1 = AMO-kurs
Endr.dato for AMO-kurs AMODATO NUMBER 6	STARTD brukes til å datere endringer. Format: YYYYMM

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Utfallskode UTFALL VARCHAR2 1	Viser om personen har fullført eller avbrutt kurset. Gyldige koder: 1 = Er kodet som avbrutt i databasen 2 = Avbrutt 8 = Fullført
Endringsdato for utfallskode UTFDATO NUMBER 6	STOPPD brukes til å datere endringer. Format: YYYYMM
Miniregisterkontroll - datokontroll MRK_DL VARCHAR2 1	Kode som viser hvilken startdato som benyttes når opplysningene hentes fra igangværendefilene: Gyldige koder: 1 = Elevens startdato er brukt. 3 = Kursets startdato er brukt. 5 = Uoppgitt settes til '19YY08' for igangværendefil (YY: år). NULL = STARTD er hentet fra avsluttafil.
Miniregisterkontroll - datokontroll MRK_DL1 VARCHAR2 1	Kode som viser hvilke start- og stoppdatoer som er benyttet når datoer hentes fra avsluttafilene: Gyldige koder: 1 = Elevens start- og stoppdato er brukt. 2 = Elevens startdato og kursets stoppdato er brukt. 3 = Kursets startdato og elevens stoppdato er brukt. 4 = Kursets start- og stoppdato er brukt. 5 = Start mangler, mens elevens stoppdato er brukt. 6 = Start mangler, mens kursets stoppdato er brukt (univ.stud.) 7 = Startdato er uoppgitt, mens elevens stoppdato er brukt. 8 = Startdato er uoppgitt, mens kursets stoppdato er brukt. 9 = Startdato >= stoppdato, mens elevens stoppdato er brukt. A = Startdato >= stoppdato, mens kursets stoppdato er brukt. NULL = Hvis MRK_DL1 er blank, er alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende, eller avsluttet i databasen.
Miniregisterkontroll - startdatokontroll på avslutta MRK_DL2 VARCHAR2 1	På avsluttafilene (år t) rettes ugyldige startdatoer ved kobling mot igangværende fil (år t-1) eller vårsemesterfila (år t). Kodene viser hvilke regler som er brukt: For universitetsstudenter: 1 = Ligger på igangværendefila med samme kurs, bruker startdato fra igangværende. 2 = Ligger på vårsemesterfila med samme kurs, startdato settes til start i vårsemesteret. 3 = Ligger på igangværendefila med ett annet kurs, setter startdato '19(t-1)09'. 4 = For resten settes startdato til start i det semesteret kurset er avsluttet. For andre som har ugyldige startdatoer: 7 = Ligger på igangværendefila med samme kurs, bruker startdato fra igangværende. 8 = Ligger på igangværendefila med ett annet kurs, setter startdato '19(t-1)08'. 9 = Personen ligger ikke på igangværendefila, setter startdato '19(t-1)08'. A = Avslutning av ordinær grunnskole; startdato settes lik stoppdato. 0 = Har ikke rettet startdatoer NULL = Hvis MRK_DL2 er blank; alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende/avsluttet i basen.

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Miniregisterkontroll - dublettkontroll MRK_DL3 VARCHAR2 1	Dersom like kurs er overlappende i tid på avsluttafilene, slår vi records sammen for å unngå slike dubletter. Vi merker hvor vi henter ulike variable fra etter følgende regler: 0 = Alle variable er hentet fra samme record (ikke dubletter, dublettene forlenger ikke kursperioden eller dubletter har ikke overlappende perioder). 1 = Startdato, men ikke stoppdato, hentes fra den recorden hvor andre opplysninger hentes fra. 2 = Stoppdato, men ikke startdato, hentes fra den recorden hvor andre opplysninger hentes fra. 3 = Start- og stoppdato hentes fra en record mens andre opplysninger hentes fra en annen record. 4 = Startdato, stoppdato og andre opplysninger hentes fra tre ulike records. NULL = Hvis MRK_DL3 er blank, er alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende, eller avsluttet i databasen.
Miniregisterkontroll - fødselsnummer MRK_FNR VARCHAR2 1	Merking av fødselsnummer. Gyldige koder: 0 = Gyldig fødselsnummer 1 = Ugyldig fødselsnummer, men gyldig d-nummer 2 = Ugyldig fødselsnummer bestående av blankt personnummer 3 = Ugyldig fødselsnummer som ikke omfattes av kode 1 eller 2
Årgang AARGANG NUMBER 4	Viser hvilken årsfil record kommer fra. Oppdateres for løpende tilfeller uten endringer. Format: YYYY
Hjelpevariabel STATUS1 NUMBER 1	Ved kobling mellom igangværende-/avsluttafila og hovedtabell (løpende record)/avgangstabell (tidligere avsluttede tilfeller) er det seks forskjellige kombinasjoner for match på FNR x NUS som kan oppstå. STATUS1 viser hvilken kombinasjon som oppsto. Gyldige koder: 1 = igangværende fil matcher hverken mot hoved- eller avgangstabell 2 = avslutta fil matcher hverken mot hoved- eller avgangstabell 3 = avslutta fil matcher mot hovedtabell 4 = igangværende fil matcher mot hovedtabell 5 = igangværende fil matcher ikke mot hovedtabell, men mot avgangstab. 6 = avslutta fil matcher ikke mot hovedtabell, men mot avgangstabell
Hjelpevariabel STATUS2 VARCHAR2 3	Verdien sier noe om hvordan recorden er behandlet i oppdateringsrutinen (se kapittel 5.5 for utfyllende informasjon). Er til hjelp under danning av forløp i databasen og til kontroll senere.
Hjelpevariabel STATUS3 NUMBER 2	Benyttes ikke
Hjelpevariabel STATUS4 NUMBER 1	Benyttes ikke
Hjelpevariabel STATUS5 NUMBER 1	Her er STATUS2 gruppert etter tilfeller som kan behandles noenlunde likt (for å lette kodingen i andre rutiner).

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Endringskode for internvariablene ENDRKODE NUMBER 6	<p>Settes ved danning av forløp i databasen og viser hvilke internvariable som evt. har endret verdi siden forrige record i forløpstabellen.</p> <p>Hver internvariabel får en unik endrekode (verdi) og disse verdiene summeres dersom endringer har funnet sted på flere internvariable samtidig. Summen angir en unik kombinasjon av endringer. De enkelte endringene har følgende verdier:</p> <p>0 = Ingen endring i noen internvariabler 1 = KLTRINN er endret 2 = HELDEL er endret 4 = VOKSEN er endret 8 = STUDRET er endret 16 = LAERLING er endret 32 = AMO er endret</p> <p>Hvis en record har f.eks. endrkode = 3 betyr det at både KLTRINN og HELDEL er endret.</p>
Tom (til og med) dato for kurset KURSTOM NUMBER 6	KURSTOM inneholder avgang opplysninger for kurset og viser når avgangen inntreffer for en tilgang, avgang eller endring. Opplysningen benyttes for å sikre at vi ikke starter opp et tilfelle på viktigste kursnivå som vil avsluttes før tidligere dateringer for det samme tilfellet i databasen.

7.11.3 Forløps-, hoved- og avgangstabell for igangværende utdanning (viktigste-kursnivå)

Tabellnavn: F_UTD_VIKTIGSTE (forløpstabell)
H_UTD_VIKTIGSTE (hovedtabell)
AVG_UTD_VIKTIGSTE (avgangstabell)

Kommentarer til forløpstabellen: F_UTD_VIKTIGSTE:

Inneholder forløp for igangværende utdanning på viktigste-kursnivå og er generert ved hjelp av årganger (avsluttaYY + igangværendeYY) fra forløpstabellen for kurs (F_UTD_KURS). For hver årgang som "kjøres" mot forløpsdatabasen, genereres det ca. 1.000.000 nye records i forløpstabellen (tilganger, endringer, overganger og avganger).

Kommentarer til hovedtabellen: H_UTD_VIKTIGSTE:

Dette er en hjelpetabell som inneholder siste datagrunnlag som er benyttet for å danne forløp i databasen. Inneholder kun en record pr. utdanningstilfelle (ett unikt utdanningstilfelle er definert vha. FNR) og blir liggende i tabellen til vi koder en avgang for tilfellet; helt ut av systemet. Inneholder til enhver tid ca. 400.000 - 420.000 records.

Kommentarer til avgangstabellen: AVG_UTD_VIKTIGSTE:

Dette er en hjelpetabell som inneholder Siste avgangstilfelle (kun en record pr. utdanningstilfelle; definert vha. FNR) for hver person som er registrert i databasen. Avgangstilfelle betyr her avgang for utdanningstilfelle; helt ut av systemet. Tabellen benyttes bl.a. for datokontroll ved danning av nye forløpsdata og for å kontrollere databasen i ettertid.

Generelle kommentarer:

Strukturen (alle kolonner) i tabellene F_UTD_VIKTIGSTE, H_UTD_VIKTIGSTE og AVG_UTD_VIKTIGSTE er identisk.

Alle utdanningsdata «knyttes til» en hovedvariabel kalt UTDKODE, som tilsvarer et overordnet løpende tilfelle. Variabelen UTDKODE har lengde to posisjoner. Første siffer angir hovedgruppe for utdanning (i alt 5 forskjellige grupper kan forekomme) og annet siffer sier noe om hendinger/endringer som er tilknyttet utdanningsgruppen.

STATUS7 sier noe om hvordan recorden er behandlet, og er delt inn i åtte grupper: B til I:

B : Ny record matcher ikke på FNR hverken mot hovedtabell eller avgangstabell
C : Ny record matcher ikke på FNR mot hovedtabell, men mot avgangstabell
D : Ny record matcher på FNR på en tilgang innen samme hovedgruppe i hovedtabell
E : Ny record matcher på FNR på en tilgang mellom ulike hovedgrupper i hovedtabell
F : Ny record matcher på FNR på en endring innen samme hovedgruppe i hovedtabell
G : Ny record matcher på FNR på en endring mellom ulike hovedgrupper i hovedtabell
H : Ny record matcher på FNR på en avgang innen samme hovedgruppe i hovedtabell
I : Ny record matcher på FNR på en avgang mellom ulike hovedgrupper i hovedtabell

For mer utfyllende informasjon om hvordan de ulike variabler kodes, se kapittel 5.6.

Variabelbeskrivelse for tabellene:**F_UTD_VIKTIGSTE, H_UTD_VIKTIGSTE og AVG_UTD_VIKTIGSTE**

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Fødselsnummer FNR VARCHAR2 11	Også records med ugyldige fødselsnumre tas med i FD-Trygd.
Hovedgruppevariabel UTDKODE VARCHAR2 2	1. siffer settes under danning av miniregisterene Denne angir eleven/studentens utdanningsnivå, og skiller bl.a. de som studerer ved høyskole og universitet. 2. siffer settes i databasen og sier noe om hendelser/endringer tilknyttet utdanningen. Gyldige koder som viser elevens/studentens utdanningsnivå: 10 = Grunnskole avgang 11 = Grunnskole tilgang 12 = Grunnskole endring 13 = Grunnskole venstresensur 20 = Videregående skole avgang 21 = Videregående skole tilgang 22 = Videregående skole endring 23 = Videregående skole venstresensur 30 = Høyskoler avgang 31 = Høyskoler tilgang 32 = Høyskoler endring 33 = Høyskoler venstresensur 40 = Universitet avgang 41 = Universitet tilgang 42 = Universitet endring 43 = Universitet venstresensur 90 = Uoppgitt avgang 91 = Uoppgitt tilgang 92 = Uoppgitt endring 93 = Uoppgitt venstresensur
Registreringsdato REGDATO NUMBER 6	Regdato fungerer som en siste oppdaterings-/endringsdato for hele recorden sett som en helhet, uavhengig om det er tilgang, avgang eller endring. Format: YYYYMM
Tilfelleets tilgangsdato TILGDATO NUMBER 6	Tilgangsdato for utdanningstilfellet. Settes ved tilgang til nytt utdanningstilfellet og endres ikke. Format: YYYYMM
Tilfelleets avgangsdato AVGDATO NUMBER 6	Avgangsdato for utdanningstilfellet. Settes når utdanningstilfellet avsluttes, ellers blank (NULL). Format: YYYYMM
Hovedgruppens tilgangsdato UTDTILG NUMBER 6	Tilgangsdato for hovedgruppen. Settes ved tilgang til ny hovedgruppe og endres ikke innenfor denne hovedgruppen. Format: YYYYMM
Hovedgruppens avgangsdato UTDAVG NUMBER 6	Avgangsdato for hovedgruppen. Settes når utdanningstilfellet avsluttes, eller ved avgang for hovedgruppen ved overgang fra gammel hovedgruppe til ny hovedgruppe, ellers blank (NULL). Format: YYYYMM

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Kursets tilgangsdato KURSTILG NUMBER 6	Settes ved kursstart og endres ikke for dette kurset. Format: YYYYMM
Kursets avgangsdato KURSAVG NUMBER 6	Settes når kurset avsluttes, ellers blank (NULL). Format: YYYYMM
Utdannings art NUS VARCHAR2 6	Disse kodene følger "Standard for utdanningsgruppering" . Dette er hele NUS-koden (6 siffer) Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.
Endringsdato for NUS NUSDATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Utdannings art NUS1 VARCHAR2 1	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder første siffer for NUS-koden. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.
Endringsdato for NUS1 NUS1DATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Utdannings art NUS2 VARCHAR2 2	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de to første siffer for NUS-koden. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.
Endringsdato for NUS2 NUS2DATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Utdannings art NUS3 VARCHAR2 3	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de tre første siffer for NUS-koden. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.
Endringsdato for NUS3 NUS3DATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Utdannings art NUS4 VARCHAR2 4	Disse kodene følger "Standard for utdanningsgruppering" . Variabelen inneholder de fire første siffer for NUS-koden. Ved koding av records til forløpstabellen F_UTD_VIKTIGST vil denne variabelen være en ordinær internvariabel hvor vi daterer evt. endringer innenfor samme utdanningstilfelle over tid.

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Endringsdato for NUS4 NUS4DATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Klassetrinn KLTRINN NUMBER 2	For studenter ved universitet blir klassetrinn justert ut fra studiemål og antatt progresjon. Ved videregående- og høgscole så blir klassetrinn satt av skolen med utgangspunkt i kurset. Hvis studenten skifter mellom høgscole og universitet, eller endrer studiemål kan klassetrinn være mindre representativt for den faktiske progresjon.
Endringsdato for klassetrinn KLTRDATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Heltid/deltid HELDEL VARCHAR2 1	Denne variabelen blir i utgangspunktet satt av skolen ut fra om kurset er ment å utgjøre et fulltidsstudium eller ikke. Gyldige koder: 1 = Heltid 2 = Deltid
Endringsdato for heltids/deltids utdanning HELDDATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Voksenopplæring VOKSEN VARCHAR2 1	Viser om kurset er en utdanning som går inn under lov om voksenopplæring. Gyldige koder: 1 = Voksenopplæring 2 = Ikke voksenopplæring
Endringsdato for voksenopplæringskode VOKSDATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Studieretningskode STUDRET VARCHAR2 2	Opplysninger om hvilken studieretning på videregående nivå (linje) kurset tilhører.
Endringsdato for studiekode STUDDATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Lærling LAERLING VARCHAR2 1	De som er igang eller har avsluttet fagprøve (lærlinger): Gyldige koder: 0 = Ikke lærling 1 = Lærling
Endringsdato for lærlingskode LAERDATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
AMO-kurs AMO VARCHAR2 1	Viser om kurset er et AMO-kurs. Gyldige koder: 0 = Ikke AMO-kurs 1 = AMO-kurs
Endringsdato for AMO- kurs kode AMODATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Utfallskode UTFALL VARCHAR2 1	Viser om personen har fullført eller avbrutt kurset. Gyldige koder: 1 = Er kodet som avbrutt i databasen 2 = Avbrutt 8 = Fullført
Endringsdato for utfallskode UTFDATO NUMBER 6	REGDATO fra F_UTD_KURS-tabellen benyttes for å datere endringer. Format: YYYYMM
Miniregisterkontroll - datokontroll MRK_DL VARCHAR2 1	Kode som viser hvilken startdato som benyttes når opplysningene hentes fra igangværendefilene: Gyldige koder: 1 = Elevens startdato er brukt. 3 = Kursets startdato er brukt. 5 = Uoppgitt settes til '19YY08' for igangværendefil (YY: år). NULL = STARTD er hentet fra avsluttafil.
Miniregisterkontroll - datokontroll MRK_DL1 VARCHAR2 1	Kode som viser hvilke start- og stoppdatoer som er benyttet når datoer hentes fra avsluttafilene: Gyldige koder: 1 = Elevens start- og stoppdato er brukt. 2 = Elevens startdato og kursets stoppdato er brukt. 3 = Kursets startdato og elevens stoppdato er brukt. 4 = Kursets start- og stoppdato er brukt. 5 = Start mangler, mens elevens stoppdato er brukt. 6 = Start mangler, mens kursets stoppdato er brukt (univ.stud.) 7 = Startdato er uoppgitt, mens elevens stoppdato er brukt. 8 = Startdato er uoppgitt, mens kursets stoppdato er brukt. 9 = Startdato >= stoppdato, mens elevens stoppdato er brukt. A = Startdato >= stoppdato, mens kursets stoppdato er brukt. NULL = Hvis MRK_DL1 er blank, er alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende, eller avsluttet i databasen.

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Miniregisterkontroll - startdatokontroll på avslutta MRK_DL2 VARCHAR2 1	<p>På avsluttafilene (år t) rettes ugyldige startdatoer ved kobling mot igangværende fil (år t-1) eller vårsemesterfila (år t). Kodene viser hvilke regler som er brukt:</p> <p>For universitetsstudenter:</p> <p>1 = Ligger på igangværendefila med samme kurs, bruker startdato fra igangværende.</p> <p>2 = Ligger på vårsemesterfila med samme kurs, startdato settes til start i vårsemesteret.</p> <p>3 = Ligger på igangværendefila med ett annet kurs, setter startdato '19(t-1)09'.</p> <p>4 = For resten settes startdato til start i det semesteret kurset er avsluttet.</p> <p>For andre som har ugyldige startdatoer:</p> <p>7 = Ligger på igangværendefila med samme kurs, bruker startdato fra igangværende.</p> <p>8 = Ligger på igangværendefila med ett annet kurs, setter startdato '19(t-1)08'.</p> <p>9 = Personen ligger ikke på igangværendefila, setter startdato '19(t-1)08'.</p> <p>A = Avslutning av ordinær grunnskole; startdato settes lik stoppdato.</p> <p>0 = Har ikke rettet startdatoer</p> <p>NULL = Hvis MRK_DL2 er blank, er alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende, eller avsluttet i databasen.</p>
Miniregisterkontroll - dublettkontroll MRK_DL3 VARCHAR2 1	<p>Dersom like kurs er overlappende i tid på avsluttafilene, slår vi records sammen for å unngå slike dubletter. Vi merker hvor vi henter ulike variable fra etter følgende regler:</p> <p>0 = Alle variable er hentet fra samme record (ikke dubletter, dublettene forlenger ikke kursperioden eller dubletter har ikke overlappende perioder).</p> <p>1 = Startdato, men ikke stoppdato, hentes fra den recorden hvor andre opplysninger hentes fra.</p> <p>2 = Stoppdato, men ikke startdato, hentes fra den recorden hvor andre opplysninger hentes fra.</p> <p>3 = Start- og stoppdato hentes fra en record mens andre opplysninger hentes fra en annen record.</p> <p>4 = Startdato, stoppdato og andre opplysninger hentes fra tre ulike records.</p> <p>NULL = Hvis MRK_DL3 er blank, er alle opplysninger om kurset hentet fra igangværendefiler. Kurset kan være løpende, eller avsluttet i databasen.</p>
Miniregisterkontroll - fødselsnummer MRK_FNR VARCHAR2 1	<p>Merking av fødselsnummer.</p> <p>Gyldige koder:</p> <p>0 = Gyldig fødselsnummer</p> <p>1 = Ugyldig fødselsnummer, men gyldig d-nummer</p> <p>2 = Ugyldig fødselsnummer bestående av blankt personnummer</p> <p>3 = Ugyldig fødselsnummer som ikke omfattes av kode 1 eller 2</p>
Årgang AARGANG NUMBER 4	<p>Viser hvilken årsfil record kommer fra.</p> <p>Format: YYYY</p>

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Hjelpevariabel K_STAT1 NUMBER 1	Statusvariabel som ble satt under danning av forløp på kursnivå (STATUS1) og som er til hjelp ved kontroll av ilegg i databasen viktigste kursnivå: Ved kobling mellom igangværende-/avsluttafila og hovedtabell (løpende record)/avgangstabell (tidligere avsluttede tilfeller) er det seks forskjellige kombinasjoner for match på FNR x NUS som kan oppstå. K_STAT1 viser hvilken kombinasjon som oppsto. Gyldige koder: 1 = igangværende fil matcher hverken mot hoved- eller avgangstabell 2 = avslutta fil matcher hverken mot hoved- eller avgangstabell 3 = avslutta fil matcher mot hovedtabell 4 = igangværende fil matcher mot hovedtabell 5 = igangværende fil matcher ikke mot hovedtab., men mot avgangstabell 6 = avslutta fil matcher ikke mot hovedtabell, men mot avgangstabell
Hjelpevariabel K_STAT2 VARCHAR2 3	Statusvariabel som ble satt under danning av forløp på kursnivå og som er til hjelp ved kontroll av ilegg i databasen for viktigste kursnivå. Gir info. om hvordan recorden ble behandlet i rutine for kursnivå.
Hjelpevariabel K_EKODE NUMBER 6	Statusvariabel som ble satt under danning av forløp på kursnivå (ENDRKODE) og som er til hjelp ved kontroll av ilegg i databasen viktigste kursnivå: Viser hvilke internvariable som evt. har endret verdi siden forrige record i forløpstabellen for kurs. Hver internvariabel har en unik endrekode (verdi) og disse verdiene er summert dersom endringer har funnet sted på flere internvariable samtidig. Summen angir en unik kombinasjon av endringer. De enkelte endringene har følgende verdier: 0 = Ingen endring i noen internvariable 1 = KLTRINN er endret 2 = HELDEL er endret 4 = VOKSEN er endret 8 = STUDRET er endret 16 = LAERLING er endret 32 = AMO er endret Hvis en record har f.eks. endrkode = 3 betyr det at både KLTRINN og HELDEL er endret.
Hjelpevariabel STATUS6 NUMBER 1	STATUS6 forteller om kurset har fått sine avgangsinformasjoner, og hvordan disse er kodet: Gyldige koder: 0 = Record inneholder ingen avgangsinformasjoner for kurset 1 = Avgangsinformasjonene er kodet ved normal avgang 2 = Record i forløpstabellen er oppdatert med avgangsinformasjoner i ettertid (for kurs som har vært viktigste kurs i en periode, men blitt avsluttet før avgangsinformasjonene var tilgjengelige for kurset). 3 = Avgang med lik dato som tilgangen. Tilganger med denne koden har fått overført avgangsinformasjonene fra avgangsrecorden som er slettet (vi har hatt en tilgang og endring med lik dato). 4 = Avgang med lik dato som gammel endring. Dersom avgangen senere gjøres om til en ny endring, så forkastes denne samtidig som avgangsinformasjonene blir overført til gammel endring.
Hjelpevariabel STATUS7 VARCHAR2 3	Verdien sier noe om hvordan recorden er behandlet i oppdateringsrutinen (se kapittel 5.6 for utfyllende informasjon). Er til hjelp under danning av forløp og til kontroll senere.

Variabelnavn Kortnavn Datatype Lengde	Koder og kommentarer
Hjelpevariabel STATUS8 NUMBER 1	Her er STATUS7 gruppert etter tilfeller som kan behandles noenlunde likt (for å lette kodingen i andre rutiner).
Hjelpevariabel STATUS9 NUMBER 1	STATUS9 forteller om ny record som blir lest inn fra F_UTD_KURS-tabellen er viktigere enn siste record som er registrert i hovedtabell for tilfellet: 0 = Det er ikke match mellom ny record og record i hovedtabell 1 = Den nye recorden er viktigere 2 = Den nye recorden er en "viktigere" avgang el. endring (samme NUS) 3 = Den nye recorden er en "viktigere" tilgang (samme NUS) 4 = Den nye recorden er en IKKE-viktigere endring, men kodes som viktigere likevel (samme NUS) 9 = Den nye recorden er IKKE viktigere
Endrekode for internvariablene ENDRKODE NUMBER 6	Statusvariabel som settes under danning av forløp: Viser hvilke endringer som evt. har skjedd i internvariablene siden forrige record ble registrert i forløpsdatabasen. Hver internvariabel får en unik endrekode (verdi) og disse verdiene summeres dersom endringer har funnet sted på flere internvariable samtidig. Summen angir en unik kombinasjon av endringer. De enkelte endringene har følgende verdier: 0 = Ingen endring i noen internvariabler 1 = KLTRINN er endret 2 = HELDEL er endret 4 = VOKSEN er endret 8 = STUDRET er endret 16 = LAERLING er endret 32 = AMO er endret 64 = NUS er endret 128 = NUS1 er endret 256 = NUS2 er endret 512 = NUS3 er endret 1024 = NUS4 er endret 2048 = UTFALL, UTFDATO og KURSAVG er satt 4096 = UTFALL, UTFDATO og KURSAVG er NULL'et ENDRKODE = 13 betyr f.eks. KLTRINN, VOKSEN og STUDRET (1 + 4 + 8 = 13) har endret verdi siden siste registrering i databasen.

De sist utgitte publikasjonene i serien Notater

- 2000/3 A.K. Johnsen og Ø. Hokstad: FoB2001: Kvalitativ testing av boligskjema - prøveundersøkelse 1999: Dokumentasjonsnotat. 32s.
- 2000/4 C. Hendriks, Ø. Hokstad og R. Sønsterudbråten: FoB2001: Boligtelling - prøveundersøkelse 1999: Dokumentasjonsnotat. 60s.
- 2000/5 K. Bjønnes, G. Dahl og B.R. Joneid: FD - Trygd: Dokumentasjonsrapport: Økonomisk sosialhjelp 1992-1997. 31s.
- 2000/6 B.R. Joneid og J. Lajord: FD - Trygd: Dokumentasjonsrapport: Demografi 1992-1997. 117s.
- 2000/7 J. Heldal: Kalibrering av AKU: Dokumentasjon av metode og program. 28s.
- 2000/8 H. Hågård og L. Rogstad: FoB2001: Adresser i folkeregisteret og GAB: Rapport fra en arbeidsgruppe for adresse-samordning og utredning av elektronisk datautveksling mellom DSF og GAB. 51s.
- 2000/9 B. Sundby: Rutiner for produksjon av statistikk over pleie- og omsorgstjenestene i kommunene 1997. 84s.
- 2000/10 E. Aas: På leting etter målefeil - en studie av pleie- og omsorgssektoren. 31s.
- 2000/11 I. Øyangen: Lokalvalgsundersøkelsen 1999: Dokumentasjonsrapport. 36s.
- 2000/12 E. Engelen: Arealbruksstatistikk for tettsteder: Dokumentasjon av arbeid med metodeutvikling 1999. 50s.
- 2000/13 F. Gundersen og A.E. Hustad: Statistikk over anmeldte lovbrudd og registrerte ofre: Dokumentasjon. 51s.
- 2000/14 T. Martinsen: Prosjekt over industriens energibruk. 58s.
- 2000/15 R. Ragnarson: Harmonisert produksjonsstatistikk for industrien. 39s.
- 2000/16 B. Halvorsen og R. Nesbakken: Fordelingseffekter av økt elektrisitetsavgift for husholdningene. 74s.
- 2000/17 J. Fosen og L. Solheim: Avledede variable i registerstatistikk: To metoder for klassifikasjon av sysselsettingsstatus. 43s.
- 2000/18 K. Myklebust: Rapport fra seminar om stedfesting av bedrifter. Oslo 1. desember 1999. 73s.
- 2000/20 K. Bjønnes, G. Dahl, J. Johansen og Ø. Sivertstøl: FD - Trygd: Dokumentasjonsrapport: Fødsels- og sykepenger, 1992-1997. 109s.
- 2000/21 A. Vedø, J-A. Sigstad Lie og J. Bjørnstad: Statistisk modellering i AKU: Modellstudier og modellestimering. 113s.
- 2000/22 B. Strøm: MSG-6 Utslippsmodellens ligningsstruktur: Teknisk dokumentasjon. 49s.
- 2000/23 T. Risberg, G. Rogdaberg og R.M. Rosvold: Sykepleiernes tilpasning i arbeidsmarkedet: En kort beskrivelse av teorier og dataregistre. 46s.
- 2000/24 A.S. Brørs, K. Dybendal, A.H. Foss og T. Jakobsen: Dokumentasjon av BESYS - befolkningsstatistikksystemet: Befolkningsendringer i 1998 og befolkningsbasen (BEBAS) 1. januar 2000. 43s.
- 2000/25 E. Høydahl: FoB2001: Kommunenes innspill om kommunehefter. 18s.
- 2000/26 T. Kalve og J. Sørøy: Revisjon av barnevernsdata. 30s.
- 2000/27 A. Skoglund: Publikasjoner fra forskningsvirksomheten 1991-1999. 72s.
- 2000/28 H. Hungnes: Omregning av KVARTS-relasjoner til MODAG-relasjoner. 12s.
- 2000/29 R.N. Johnsen: Undersøking om foreldrebetaling i barnehagar, januar 2000. 36s.
- 2000/30 O. Rognstad: Plan for landbruksstatistikken etter 1999. 23s.
- 2000/31 Ø. Kleven: Levekårsundersøkelsen i Longyearbyen 2000: Dokumentasjon og tabellrapport. 188s.