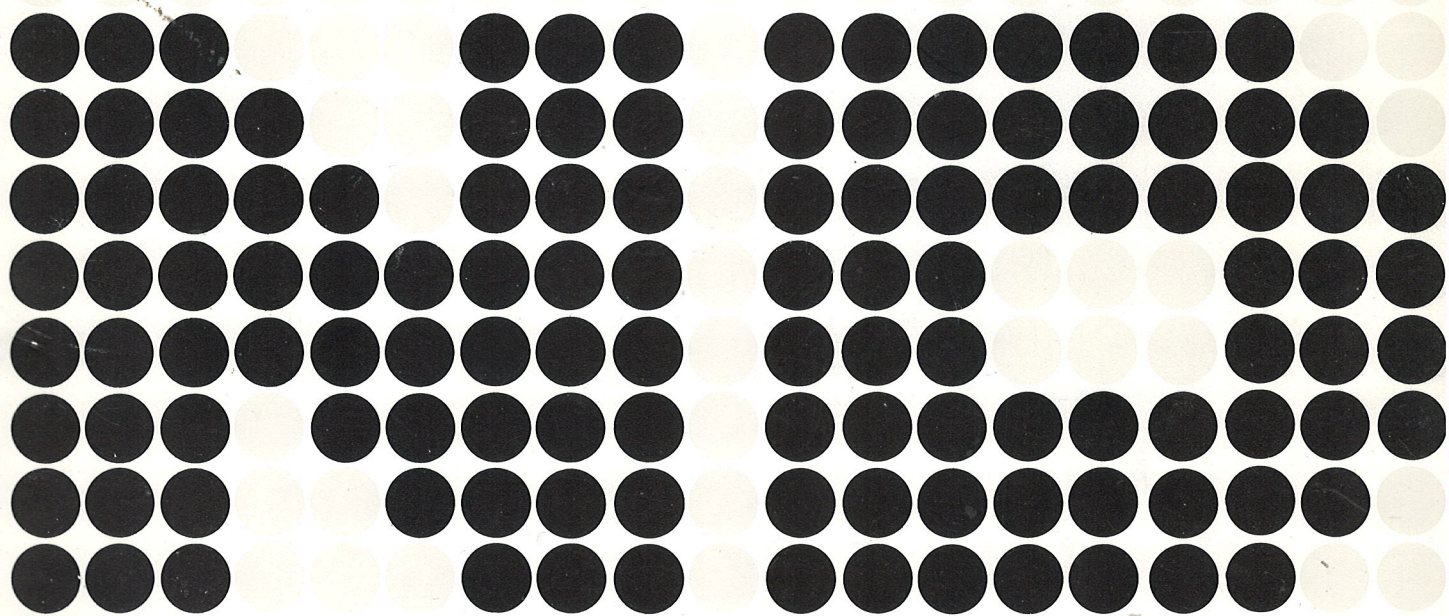


ACCESS DBA

HANDBOK

ND Norsk Data AB



ACCESS DBA

HANDBOK

Skrivdatum: 14 april 1983

Förord:

PRODUKTEN

Denna manual beskriver Data Dictionary-DBA-modulen i fråge- och transaktionssystemet ACCESS 1, version B. Produkten är registrerad med ND-nummer ND-10185B.

LÄSAREN

Avsikten med denna manual är att sätta läsaren i stånd till att utföra underhållsuppgifterna i förbindelse med systemet ACCESS.

FÖRHANDSKUNSKAPER

Manualen förutsätter grundläggande kunskap om ADB. För att förstå denna manual bör läsaren känna till NDs operativsystem SINTRAN med de vanligaste kommandona i timesharing, användningen av ISAM-filer samt själva ACCESS-systemet. Om ACCESS skall användas i ett distribuerat nätverk (NORDNET), bör läsaren också ha kunskap om hur data överförs i ett sådant nät (t ex veta vad en datakanal är, hur de etableras och tas ned, etc.).

ANDRA AKTUELLA MANUALER

ACCESS Användarhandledning	ND-60.152
SINTRAN III Reference Manual	ND-60.128
SINTRAN III Timesharing-Batch Guide	ND-60.132
ISAM Users Guide	ND-60.108
SINTRAN III Communication Guide	ND-60.134
The Data Base System SIBAS II	
ND User Guide	ND-60.127
ND-100 Sort-Merge System	ND-60.146

MANUALEN

Denna manual beskriver DBA-modulen i ACCESS. Den innehåller den information som är nödvändig för att underhålla och ha ansvaret för driften av ACCESS. Manualen är upplagd mer som en uppslagsbok än som en lärobok, som skall läsas från pärm till pärm.

I N N E H Å L L S F Ö R T E C K N I N G

<u>Sektion</u>	<u>Sida</u>
1 ACCESS och DATABASADMINISTRATÖREN	1
1.1 Inledning	1
1.2 Översikt över ACCESS och DBA	1
2 DBA I SINTRAN	3
2.1 Start och stopp av DDI	3
2.2 Användare i systemet	4
2.3 DBA och användarområden i SINTRAN filsystem	4
2.4 Kommandon	5
2.5 Aktivering av DBA.	6
2.6 Körning av DBA utan att avbryta ACCESS-användarna	6
2.7 BACKUP av databasbeskrivningen	8
2.8 BACKUP av databaserna	9
2.9 DRIFTSORGANISATION	10
3 DATABASDEFINITIONSNIVÅ	11
3.1 Tillgängliga kommandon	11
3.2 Definiering av ACCESS-användare	11
3.3 Definition av Databas-Administratör	13
3.4 Definition av Dataelementen	13
3.4.1 Specificering av lagringsformat	14
3.4.2 Specificering av Utskriftsformat (Utformat)	15
2.1 Exempel på utformat	16
2.2 Kommentar till Lagringen av Data	16

<u>Sektion</u>	<u>Sida</u>
3.4.3 Standardvärden i fälten	16
3.5 Sökning efter förekomst av olika typer	17
3.6 Odefinierade Typer	17
3.7 Definition av en ny databas	17
3.8 Översikt över tillgängliga databaser	19
3.9 Strykning av databas	19
3.10 Ändring i existerande databasbeskrivning	19
3.11 Avslutning på definitionsnivån (1)	19
 4 TABELLDEFINITIONSNIVÅ	 20
4.1 Tillgängliga kommandon	20
4.2 Tillgång till tabellerna	20
4.3 Definition av dataelementen i tabellen	21
4.4 Definition och redigering av en tabell	21
4.4.1 Nyckel	22
4.4.2 Kolumnbredd och radlängd	23
4.5 Tabeller och filer i ACCESS	23
4.6 Knytning av DDI's tabellinformation till en fysisk ISAM-fil	24
4.7 Definition av SIBAS-tabeller	25
4.8 Knytning av Attributnamn till SIBAS-itemnamn	26
4.9 Användning av 'sekventiella' filer	26
4.10 Strykning av tabellbeskrivning	27
4.11 Översikt över tillgängliga tabeller i databasen	27
4.12 Avslutning på tabelldefinitionsnivå (nivå 2)	27

<u>Sektion</u>	<u>Sida</u>
5 Allmänt om körning av ACCESS	28
5.1 Terminaltyp	28
5.2 Kommandofilerna	28
5.3 NO SUCH FILE	29
5.4 Fel på användarens temporära arbetsområde	29
5.5 DDI inte aktiv	29
5.6 Kommunikationsproblem mellan DDI och ACCESS	30
5.7 Allmänt om användningen av filer	30
5.8 Förbättringar i sökhastigheten för ACCESS	30
5.9 Andra felmeddelanden	31
5.10 Terminalen 'hänger'	32
6 FULLSTÄNDIGA EXEMPEL	33
6.1 Exempel på registrering av databasadministratör och användare	33
6.2 Exempel på definition av en databas	35
DDI-PROCESSEN SOM BATCH-PROCESS	39
KONTROLLKOMMANDON I DBA-MODULEN	43

1 ACCESS och DATABASADMINISTRATÖREN

1.1 Inledning

ACCESS är ett system för användare, som med en minimal kunskap om ADB, skall kunna behandla stora mängder information. Systemet hämtar själv en beskrivning av den information, som skall behandlas. ACCESS förutsätter, att denna beskrivning finns lagrad på ett sätt, som ACCESS kan använda. Normalt lagras beskrivningen på ISAM-filer under användaren DATA-DICTIONARY. Beskrivningarna läses av en process som kallas Databasregistret eller på engelska - Data-Dictionary - DDI.

DDI är en process*) som går samtidigt med ACCESS, och som är i stånd att ge ACCESS en fullständig och tillräcklig beskrivning av de datastrukturer systemet opererar på.

Vi behöver också hjälpmedel för att anpassa systemet för användarna, t ex lägga in nya beskrivningar och rätta gamla. Detta hjälpmedel har vi kallat DBA - Data Bas Administratör. Detta namn används också på den, som är ansvarig för driften av systemet, och det är endast denna DBA-användare, som bör ha tillgång till DBA-modulen.

DBA-modulen, som beskrivs här, har till funktion att lägga in beskrivningar av (användare och databaser), rätta dem och stryka dem, som inte behövs längre. För att ACCESS skall kunna utföra transaktioner, är systemet beroende av att beskrivningen av datastrukturerna är riktig. ACCESS påverkas också av hur data är strukturerade. Om strukturen är bra, blir systemet snabbare.

*) DDI-systemet är kopplat till en batch-process i SINTRAN, när man då inte behöver belägga någon terminal. Om en terminal kan reserveras för DDI, behöver man inte använda batchprocessen, utan kan starta DDI från terminalen. Se bilaga A.

1.2 Översikt över ACCESS och DBA

En ACCESS-användare kan ha tillgång till en eller flera databaser. En databas är en samling av 'sekventiella' filer, ISAM-filer och SIBAS realmar. I en SIBAS databas kan du läsa, men inte skriva från ACCESS. SIBAS-processen måste vara uppstartad, innan du kan få tillgång till en SIBAS-databas.

Beskrivningen av ACCESS-databaserna ligger på en samling filer, som automatiskt upprättas av DBA. Dessa filer måste läggas på användare DATA-DICTIONARY, dvs samma användare som DDI-processen körs under.

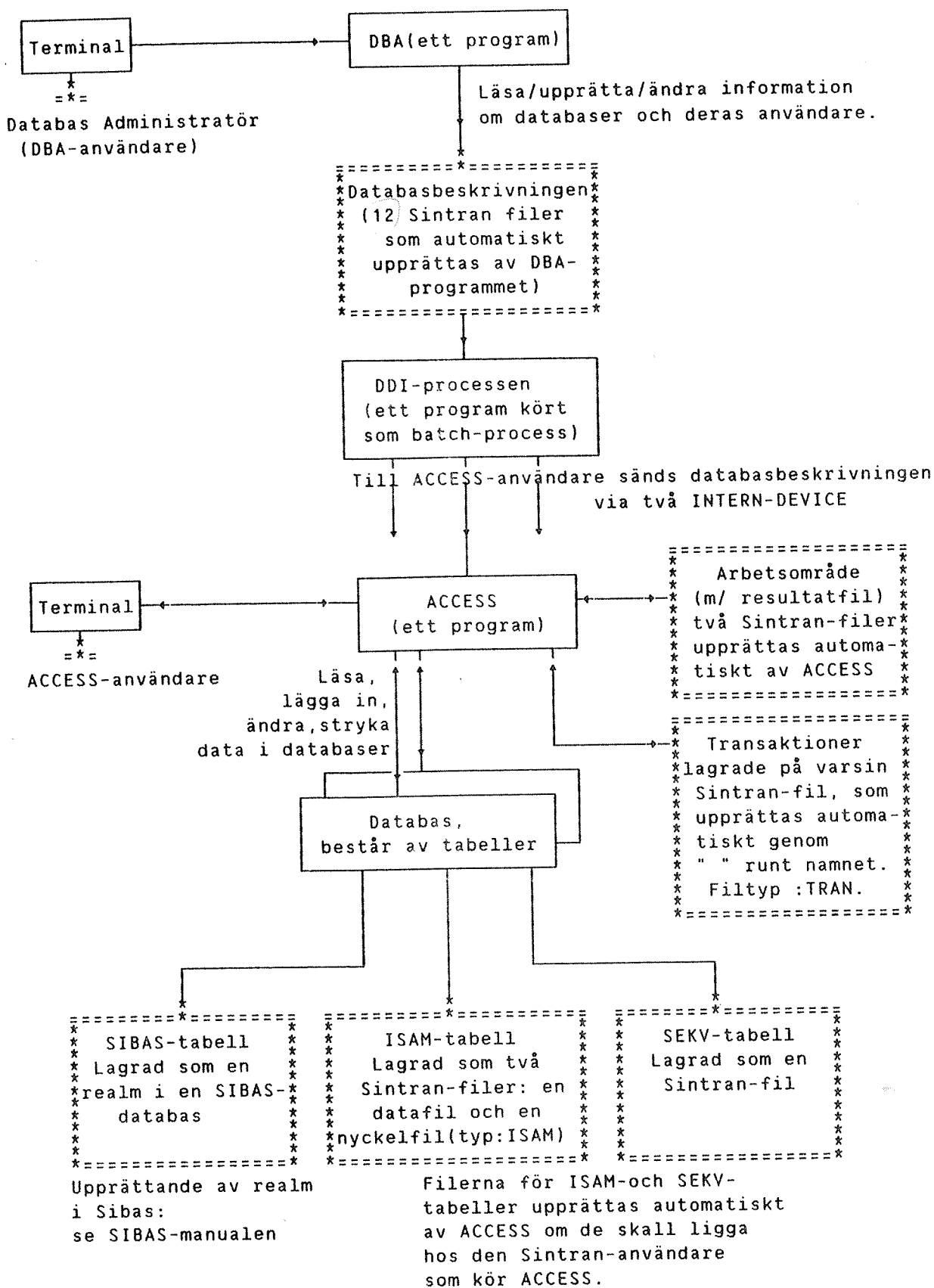


Fig. 1. Översikt över ACCESS och DBA

2 DBA I SINTRAN

2.1 Start och stopp av DDI

DBA är ett självständigt reentrant subsystem i SINTRAN. Om DBA ska köras, kan inte DDI-processen vara aktiv, därför att DBA behöver samma filer och kan ändra på innehållet i dem.

För att använda DBA loggar man in som användare DATA-DICTIONARY, där beskrivningen av databaser normalt skall ligga.

Obs. genom att logga in under ett annat användarnamn får man inte tillgång till den databasbeskrivning ACCESS eventuellt har samlat tidigare under användare DATA-DICTIONARY. Likaså blir den information man lägger in genom DBA-modulen under den nya användaren, normalt ignorerad av ACCESS senare! Se närmare under avsnitt 2.3.

Efter det att du har loggat in, bör du kontrollera att ingen kör ACCESS för ögonblicket, t ex genom ÉTERM-STAT,,, (sök CC ACCESS CC). Om någon kör, be dem sluta, annars får ni problem när du fortsätter enligt nedan.

Stoppa så DDI-processen, om det behövs, genom Sintran-kommandot:

```
ÉABORT-JOB <batchno>,DATA-DICTIONARY
```

Du kan kontrollera, att inte DDI-processen går, genom kommandot:

```
ÉLIST-BATCH-PROCESS
```

och kontrollera, att användare DATA-DICTIONARY inte listas som användare av någon batch-process.

Nu är du klar att köra DBA-modulen: Ge kommandot

```
ÉDBA
```

När du har använt DBA färdigt, kan ACCESS köras igen. Men först måste DDI-processen startas genom kommandot:

```
ÉAPP-BATCH <BATCH-NR>, (DA-DICT)DDI-MACH:BATC,(DA-DICT)DDI-MACH:SYMB
```

De två sista parametrarna i kommandot ovan är batch-processens in-fil och ut-fil.

Man kan också få DDI-processen att starta automatiskt, när man går ut ur DBA-modulen, genom att en gång för alla ersätta rad 3 i filen (DATA-DICT)DBA-HELP:SYMB med kommandot beskrivet ovan, men utan É. (Använd PED eller NOTIS-WP! Se upp, så att du inte tar bort eller lägger till rader i filen. Du skall endast skriva över rad 3.)

Obs.:

APP- skall stå längst till vänster och kommandot måste rymmas på de 59 första positionerna på raden.)

Transaktionsdelen av ACCESS är beroende av svar från DDI-processen för att kunna fungera, så att DDI måste vara aktiv, när transaktions-systemet går.

Om du vill veta lite mer om körning av DDI som batchprocess, kan du läsa om det i SINTRAN-III, Time-sharing/ Batch Guide, ND-60.132.01.

Obs. att alla kommandon till SINTRAN kan förkortas så länge, de är entydiga. Det vill säga att kommandot ovan t ex kan också skrivas:

ÉAB-J <batch no>,D-D

ÉLI-B-P

ÉAP-B <batch no>,(D-D)DDI-M:BA,(D-D)DDI-M:SY

2.2 Användare i systemet

ACCESS har sin egen användaridentifikation som inte är identisk med SINTRANS. En ACCESS-användare definieras i DBA-delen, och ges där tillgång till databasen. Till varje ACCESS-användare tillordnas ACCESS ett arbetsområde, som består av två SINTRAN-filer. Detta arbetsområde måste definieras för alla användare. Se EDITERA-ANVÄNDARE.

Endast en ACCESS-användare kan använda samma arbetsområde (fil) samtidigt. Däremot kan flera personer (terminaler) samtidigt operera under samma SINTRAN-namn. Sammanhanget mellan SINTRAN användarnamn och arbetsområde förklaras närmare i sektion 3.2.

2.3 DBA och användarområden i SINTRAN filsystem

För att kunna ladda ACCESS i SINTRAN med hjälp av medföljande mode-fil, måste användare SYSTEM vara definierad som 'Friend' hos användare DATA-DICTIONARY, eller alla filer av typen :BPUN under DATA-DICTIONARY vara definierade med 'Public' skrivrättighet.

Om du skulle köra DBA från en annan användare än DATA-DICTIONARY, får du en separat databasbeskrivning på detta användarområde. Obs. att ACCESS inte använder sig av en beskrivning på ett sådant användarområde med mindre än, att DDI-processen körs under denna användares namn. Dvs., önskar du att använda en sådan separat beskrivning, måste du, innan du startar DDI-processen, ändra innehållet i batch-processens in-fil (DATA-DICT)DDI-MACHINE:BATC så att namnet DATA-DICTIONARY på första raden byts ut mot namnet på den SINTRAN-användare du loggat in på, när du kör DBA-modulen.

Skall ditt system använda en databasbeskrivning byggd av DBA på ett annat SINTRAN-användarområde förutom den "normala" beskrivningen på DATA-DICTIONARY, måste du starta två DDI-processer på varsin batch-processor. De måste använda olika par av "intern-device" och ha varsin kopia av in- och ut-filer nämnda i APP-BATCH kommandot i avsnitt 2.1 ovan.

(Se också beskrivningen av hur ACCESS laddas in i SINTRAN från FLOPPY.) Kopia av in-filen måste (på första raden) få varsitt av

de två SINTRAN användarnamnen.

Det är filen ACCESS-HELP:SYMB, som bestämmer vilken av de två DDI-processerna, som ACCESS skall kommunicera med. När du startar ACCESS-modulen från ett SINTRAN-användarområde, letar ACCESS först på detta område efter filen ACCESS-HELP:SYMB. Om den inte finns där, används den utgåva av ACCESS-HELP, som ligger på det område, som tillhör DATA-DICTIONARY. I den fil som påträffas, måste det vara uppgivet vilka två intern-device, som den önskade DDI-processen använder för kommunikation.

Alltså: för varje SINTRAN-användare där ACCESS skall använda en databasbeskrivning, som inte ligger hos DATA-DICTIONARY, måste du kopiera filen (DA-DICT)ACCESS-HELP:SYMB över till denna SINTRAN-användare och på rad 2 och 3 ändra numren på intern-device, så att de stämmer med motsvarande nummer i in-filen till den önskade DDI-processen.

Alternativet med separata databasbeskrivningar kan vara användbart t ex för att skilja drift av databasen från utveckling. Det kan också utnyttjas genom att nybörjare får experimentera med sin egen databasbeskrivning utan att förstöra den "verkliga" databaskörning, som använder DATA-DICTIONARY.

2.4 Kommandon

Kommandon ges i kommandoposition, markerad med * eller ** under strecken nederst på skärmen.

DBA-modulen har två nivåer: Nivå 1 kallas Databasdefinitionsnivå. Det är här databasen definieras, medan nivå 2 kallas Tabelldefinitions-nivån, då det är här alla tabeller definieras.

Antalet stjärnor visar, vilken nivå man är på:

- * → Databasdefinitionsnivå
- ** → Tabelldefinitionsnivå

Alla kommandon i DBA kan förkortas på samma sätt som i SINTRAN, d.v.s. så länge de är entydiga. Det vill säga, att *E-TA är tillräckligt för att ange *EDITERA-TABELL, eftersom det är skilt från kommandot *EDITERA-TYPER.

Blanka före kommandonamn är inte tillåtna!

Ett kommando avbryts genom tryckning på 'HEMMA'-tangenten:



2.5 Aktivering av DBA.

DBA startas med kommandot:

ÉDBA

Kom ihåg att du normalt måste vara inloggad som användare DATA-DICTIONARY, om du vill rätta de beskrivningar, som DDI-processen skall ge till ACCESS.

DBA frågar efter användareidentifikation. Användareidentifikationen är inte SINTRANs användareidentifikation, som här är 'DATA-DICTIONARY', utan ACCESS's egen användareidentifikation. För att en ACCESS-användare skall få tillgång till DBA-modulen, måste han/hon vara definierad som DBA-användare. Är det första gången DBA-modulen används, står ACCESS både som användare och DBA-användare, utan lösenord. Om du önskar lägga upp en separat databeskrivning, som beskrivs i 2.3, frågar DBA inte om användareidentifikation. Du kommer då rätt in i användarbeskrivningen, där SYSTEM står uppfört i förväg. Ändra SYSTEM till önskat användarnamn, och ge arbetsområde och ev. lösenord. (Se avsnitt 3.2). Det nya namnet förs automatiskt upp som DBA-användare.

2.6 Körning av DBA utan att avbryta ACCESS-användarna

När du startar DBA-modulen, som beskrevs i avsnitt 2.1, måste alla ACCESS-användare sluta arbeta, medan DBA-modulen används. Pausen kan göras minimal genom att DBA-modulen i stället körs på en annan SINTRAN-användare än DATA-DICTIONARY - t ex på användare ACCESS - och att databasbeskrivningen kopieras till användare DATA-DICTIONARY efter varje körning av DBA. På detta sätt behöver ACCESS-användarna bara vänta under själva kopieringen av filerna.

Nedan är det satt upp ett förslag till mode-fil, som kan användas för kopieringen. Genom att mode-jobben 'lånar' intern-devicen under kopieringen, behöver ingen ACCESS-användare avbryta sin körning under kopieringen! De får bara lite extra väntetid om de skall ha något gjort, som de behöver DDI-processen till. Obs. användningen av BASIC som utvidgning av SINTRAN som 'JOBBS-KONTROLL-SPRÅK', maskinen måste ha BASIC installerad, för att mode-filen skall kunna användas.

ÉCC DETTA MODE-JOB KOPIERAR DATABASBESKRIVNINGEN
ÉCC FRÅN SINTRAN-ANVÄNDARE ACCESS TILL DATA-DICTIONARY,
ÉCC UTAN ATT ABORTERA EVENTUELLA ACCESS-ANVÄNDARE.
ÉCC DU BÖR LIKVÄL SÄNDA EN 'BROADCAST' OM VILKA ÄNDRINGAR
ÉCC DU HAR GJORT PÅ DATABASBESKRIVNINGEN, OCH SÄGA ATT
ÉCC ACCESS-ANVÄNDARE KAN BLI TVUNGNA ATT VÄNTA ETT PAR
ÉCC MINUTER UNDER ETT AV KOMMANDONA DE GER.

ÉCC DU MÅSTE GÖRA FÖLJANDE ÄNDRINGAR I MODEFILEN INNAN DU STARTAR:
ÉCC 1.INTERNDEVICE-NUMRET 129 NEDAN MÅSTE PÅ 6 STÄLLEN
ÉCC ÄNDRAS TILL DET DEVNO2 SOM DU FINNER I FILEN
ÉCC (DATA-DICT)ACCESS-HELP:SYMB, RAD 3.
ÉCC 2.I KOMMANDOT ÉABORT-JOB 2 ... OCH

CC ÉAPP-BATCH 2 MÅSTE TALET 2 ÄNDRAS TILL NUMRET
ÉCC PÅ DEN BATCH-PROCESSORN SOM DDI-PROCESSEN ANVÄNDER.
ÉCC OBS. ATT MODE-FILEN MÅSTE HA STORA BOKSTÄVER!
ÉCC DU MÅSTE VARA INLOGGAD SOM ANVÄNDARE DATA-DICTIONARY
ÉCC UNDER KÖRNINGEN.

ÉBASIC

10 REMARK. VI SKALL ANVÄNDA MONITORCALLET RESRV SOM
11 ' EN FUNKTION:
20 EXTERNAL RESRV
21 INTEGER RESRV
25 REMARK. RESRV ÄR MONITORCALLET FÖR ATT RESERVERA
26 ' INTERN-DEVICE. RELES FRIGER DEM IGEN.

50 REMARK. KOLLA OM DDI-PROCESSEN ÄR I GÅNG:
60 A% = RESRV(129%,1%,1%)
66 REMARK. OM A% >< 0, GÅR DDI-PROCESSEN.
70 IF A% >< 0% GO TO 100
80 CALL RELES(129%,1%)
90 GO TO 200

100 REMARK. TVINGA ACCESS-ANVÄNDARNA ATT VÄNTA I KÖ:
110 B% = RESRV(129%,0%,0%)
115 PRINT "ACCESS-ANVÄNDARNAS INTERN-DEVICE RESERVERAT"

150 REMARK. STOPPA DDI-PROCESSEN:
160 ÉABORT-JOB 2 DATA-DICTIONARY
165 PRINT "DDI-PROCESSEN STOPPAD"

170 REMARK. VI LURAR NYA ACCESS-KÖRARE
171 ' ATT TRO ATT DDI-PROCESSEN ÄR OK I FRAMTIDEN:
180 B% = RESRV(129%,1%,0%)
185 PRINT "DDI-PROCESSENS INTERN-DEVICE ÖVERTAGNA"

200 REMARK. KOPIERINGEN STARTAR:
210 ÉCOPY-FILE (DATA-DICT)DATABASE:DATA (ACCESS)DATABASE:DATA
215 PRINT "FIL 1"
220 ÉCOPY-FILE (DATA-DICT)DATABASE:ISAM (ACCESS)DATABASE:ISAM
225 PRINT "FIL 2"
230 ÉCOPY-FILE (DATA-DICT)ACCESS:DATA (ACCESS)ACCESS:DATA
235 PRINT "FIL 3"
240 ÉCOPY-FILE (DATA-DICT)ACCESS:ISAM (ACCESS)ACCESS:ISAM
245 PRINT "FIL 4"
250 ÉCOPY-FILE (DATA-DICT)TABLE:DATA (ACCESS)TABLE:DATA
255 PRINT "FIL 5"
260 ÉCOPY-FILE (DATA-DICT)TABLE:ISAM (ACCESS)TABLE:ISAM
265 PRINT "FIL 6"
270 ÉCOPY-FILE (DATA-DICT)ATTRIBUTE:DATA (ACCESS)ATTRIBUTE:DATA
275 PRINT "FIL 7"
280 ÉCOPY-FILE (DATA-DICT)ATTRIBUTE:ISAM (ACCESS)ATTRIBUTE:ISAM
285 PRINT "FIL 8"
290 ÉCOPY-FILE (DATA-DICT)USER:DATA (ACCESS)USER:DATA
295 PRINT "FIL 9"
300 ÉCOPY-FILE (DATA-DICT)USER:ISAM (ACCESS)USER:ISAM
305 PRINT "FIL 10"
310 ÉCOPY-FILE (DATA-DICT)TYPE:DATA (ACCESS)TYPE:DATA

```
315 PRINT "FIL 11"
320 ÉCOPY-FILE (DATA-DICT)TYPE:ISAM      (ACCESS)TYPE:ISAM
325 PRINT "FIL 12 (SISTA FIL)"

400 REMARK. OM DDI-PROCESSEN MÅSTE STOPPAS, DVS. A% >< 0%,
401 '      SÅ MÅSTE DESS INTERN-DEVICE FRIGES:
410 IF A% = 0% THEN GO TO 500
420 CALL RELES(129%,1%)
425 PRINT "DDI-PROCESSENS INTERNDEVICE ÄR FRIGIVET"

500 REMARK. VI STARTAR DDI-PROCESSEN:
510 ÉAPP-BATCH 2 (DATA-DICT)DDI-MACHINE:BATC (DA-DICT)DDI-M:SYMB
515 PRINT "DDI-PROCESSEN UPPSTARTAD"

550 REMARK. OM DDI-PROCESSEN GICK TILL ATT BÖRJA MED,
551 '      MÅSTE ACCESS-ANVÄNDARNA FÅ KOMMA UT UR KÖN:
560 IF A% = 0% GO TO 600
570 CALL RELES(129%,0%)
575 PRINT "ACCESS-ANVÄNDARNAS INTERNDEVICE FRIGIVET"

600 END
LOAD FTNLIBR:BRF
RUN
EXIT
ÉCC DETTA ÄR SISTA RADEN I MODE-FILEN.
```

2.7 BACKUP av databasbeskrivningen

När DBA-modulen körs, bygger du en beskrivning av databaser och deras användare. Denna databasbeskrivning ligger på 12 filer, normalt hos användare DATA-DICTIONARY. Varje gång du startar DBA bör du ta 'backup' av dessa filer, dvs ta säkerhetskopior, som du kan kopiera tillbaka, om något går snett med originalfilerna. Kopiorna bör inte ligga på den diskpacke, där originalen ligger. Nedan antar vi att kopiorna skall ligga hos SINTRAN-användare BACKUP-USER, där alla de 12 filerna måste finnas före körningen. Användare DATA-DICTIONARY måste vara 'friend' hos BACKUP-USER.

För kopieringarna kan MODE-filen i föregående avsnitt användas - medan du är inloggad som DATA-DICTIONARY. Detta förutsätter att BASIC är installerat på maskinen. DDI-processen behöver inte stoppas för kopieringen (men ACCESS-användarna märker en paus medan kopieringen sker). Innan mode-filen körs, måste du göra följande förändringar (förutom de som nämns i mode-filen själv):

1. För att ta säkerhetskopior:

Ersätt användarnamnet (DATA-DICT) med (BACKUP-USER) i alla COPY-FILE-kommandon i mode-filen. Ersätt så användarnamnet (ACCESS) med (DATA-DICT) i samma kommandon.

2. För att ersätta originalen med kopiorna:

Ersätt användarnamnet (ACCESS) med (BACKUP-USER) i alla COPY-FILE-kommandon.

2.8 BACKUP av databaserna

Förutom 'backup' av databasbeskrivningen, bör du ta säkerhetskopia av alla databaserna som du beskriver där. Detta görs genom att ta kopia av varje enskild tabell i databaserna.

För tabeller av typ SIBAS utförs en vanlig 'backup' av SIBAS-databasen, se SIBAS-manualen.

För de andra tabellerna kan du kopiera de filer, tabellerna är lagrade på. Använd kommandot **ISAM-FILER eller **SEKV-FILER i DBA-modulen för att se namnen på de SINTRAN-filer, som hör till tabellen. Glöm inte att kopiera filerna av typ :ISAM för tabeller av typ ISAM! Skulle något gå galet, kopierar du bara tillbaka filerna igen.

För tabeller av typ ISAM eller SEKV kan du alternativt bygga transaktioner i ACCESS för att ta säkerhetskopia:

Gå till transaktionsbeskrivningsnivån i ACCESS och ta fram en arbetstabell för den önskade tabellen. Fyll i arbetstabellen med endast SKRIV. i tabellkolumnen. Ta fram en kommandobox och skriv i denna:

```
DATA-LISTNING (BACKUP-USER)TABELLNAMN:BACK
```

där parametern ändras till namnet på den SINTRAN-fil, där du önskar lagra säkerhetskopian av denna tabell.

Du lagrar så denna transaktion genom kommandot:

```
**LAGRA-TRANSAKTION "BACKUP-TABELLNAMN"
```

där TABELLNAMN är namnet på den aktuella tabellen.

Var gång du vill ta en ny säkerhetskopia av tabellen, ger du bara detta kommando i ACCESS:

```
*UTFÖR BACKUP-TABELLNAMN
```

Obs. att du måste ha 'write-access' till BACKUP-USER, när du gör detta. Filen (BACKUP-USER)TABELLNAMN:BACK måste finnas sen förut.

Behöver du ersätta originaltabellen med säkerhetskopian, går du in i ACCESS, ger kommandot *LADDA-DATABAS, uppger namn på databas och tabell, och ger så kommandot:

```
**LADDA-DATA-FIL (BACKUP-USER)TABELLNAMN:BACK
```

Svara JA på frågan om standardvärden önskas, tryck på HEMMA-tangenten och svara JA på frågan om du vill lägga in data, och svara JA på frågan om gamla data skall strykas. (Se också manualen för ACCESS-delen av systemet.) Vid ISAM-filer får du en nyskriven nyckelfil på detta sätt så att eventuella fel i denna, vid den tidpunkt du tog

säkerhetskopian, rättas.

2.9 DRIFTSORGANISATION

Om det är önskvärt att skilja drift och utveckling av användandet av ACCESS mot en databas, kan arbetet t ex organiseras som så:

Databasbeskrivningarna kopieras från användare DATA-DICTIONARY till ACCESS. Till detta kan du använda motsvarande mode-fil som i avsnitt 2.6, men med SINTRAN-användarnamnen (DATA-DICT) och (ACCESS) ombytta. Användare DATA-DICTIONARY måste då vara 'friend' hos SINTRAN-användare ACCESS.

Beskrivningen av den nya databasen byggs så med DBA-modulen, medan du är inloggad som SINTRAN-användare ACCESS.

Vid eventuella SIBAS-tabeller kopieras en test-version av den verkliga SIBAS-databasen till användarområdet ACCESS och startas genom SIBAS-SERVICE på eget systemnr med 'owner' ACCESS.

Vid tabeller av typ ISAM eller SEKV kopieras de filer, tabellerna ligger på till användare ACCESS. Använd samma filnamn! Genom kommandot **ISAM-FILER eller **SEKV-FILER i DBA-modulen skall filnamnen vara listade utan SINTRAN-användare angiven på förhand.

En egen 'test'-DDI-process för databasbeskrivningr på användare ACCESS startas upp, se avsnitt 2.3.

Nu kan ACCESS köras mot kopiorna av databasen för att testa ut, att de fungerar som det ska, eventuellt för träning/utbildning. Användaren måste vara inloggad som ACCESS. Obs. att en kopia av filen ACCESS-HELP:SYMB måste ligga hos SINTRAN-användare ACCESS, med device-numren till test-DDI-processen inskrivna. (Se avsnitt 2.3).

Efter det att databasbeskrivningen är testad och funnen i ordning, måste man stoppa test-DDI-processen och köra DBA-modulen en gång till för att skriva in SINTRAN-användare på filnamnen till de 'egentliga' SEKV- och ISAM-tabellerna (kommando **ISAM-FILER och **SEKV-FILER). Så kopieras databasbeskrivningen till DATA-DICTIONARY med mode-filen i avsnitt 2.6, och det verkliga arbetet med ACCESS mot denna databas kan börja. 'Backup' kan tas samtidigt av databasbeskrivningarna på användare ACCESS.

3 DATABASDEFINITIONSNIVÅ

3.1 Tillgängliga kommandon

Trycker man "HJÄLP" i kommandoposition, får man en översikt på skärmen över alla tillåtna kommandon på databasdefinitions nivå:

Kommando	Betydelse
HJÄLP-INFORMATION	Ge information om tillåtna kommandon.
EXIT	Lämna DBA-modulen.
EDITERA-TYP	Editering av variabeltyper.
VAR-ANVÄNDS-TYPEN <typnamn>	Ge information om var datatyp används.
ODEFINIERADE-TYPER	Lista datatyper som används, men är odefinierade.
DBA-ANVÄNDARE	Definiera dem som har tillgång till DBA
EDITERA-ANVÄNDARE	Editera användarbeskrivning.
NY-DATABAS <dbnamn>	Definition av ny databas.
EDITERA-DATABAS <dbnamn>	Editering av databasbeskrivning.
STRYK-DATABAS <dbnamn>	Strykning av databas.
LISTA-DATABASER	Lista ut alla definierade databaser.

Du kan använda pil-ned (↓) och pil-upp (↑) tangenterna för att bläddra i hjälpinformationen. Alla andra tryckningar ger kommandoposition. Dit kommer du också med pil-upp eller pil-ned, om ny sida inte finns.

Om du är tveksam om vilka kommandon du kan använda, kan du alltid trycka hjälp i kommandoposition, och alla tillgängliga kommandon visas då på skärmen. Alternativt kan du använda CTRL/G, eller HJÄLP-tangenten på NOTIS-terminalen (TDV-2200/9).- Använd HEMMA-tangenten för att komma tillbaka till utgångspunkten.

3.2 Definiering av ACCESS-användare

Tryck in kommandot :

*EDITERA-ANVÄNDARE

och följande skärmbild kommer då fram:

ACCESS System användare		
ACCESS användarnamn	Lösenord	Arbetsområde (:SORT och :DATA)

Användarnamn

Existerande ACCESS-användarnamn listas i kolumnen för detta, och nya användare kan definieras genom att fylla i fältet under sista användare. Om det inte finns någon användare av ACCESS, står användare SYSTEM, och detta namn ändras till ditt eget.

Användare kan strykas med SKIFT/STRYK-tangenten på Notisternalen (eller CTRL/D CTRL/L). Man kan hoppa mellan fält med hjälp av piltangenterna (pil-upp eller pil-ned), CR eller tabulatorfunktionerna. Enstaka tecken kan strykas med DEL-tangenten eller CTRL/A. Andra användare, som är DBA, får du inte lov att ändra på. Detta är reglerat, genom att att du inte kan hoppa till en sådan rad med piltangenterna.

LÖSENORD

För att skydda användningen av ACCESS kan användarna tilldelas ett lösenord. Lösenordet för en användare skrivs ut i fältet under Lösenord, och DBA kan ändra på lösenordet, stryka det, och tilldela nya lösenord. En DBA kan inte få se och kan inte ändra på lösenordet för andra DBA. Dessa är överskrivna med stjärnor.

ARBETSSOMRÅDE

För varje användare skall det definieras två filer som ACCESS använder som temporärt arbetsområde, en som buffer och den andra för sortering. Dessa är två SINTRAN-filer som skall ha samma namn men olika filtyp, vilka skall vara ":DATA" och ":SORT". ACCESS måste kunna skriva på dessa filer när en användare kör transaktioner med ACCESS. Filerna bör följaktligen ligga på det användarområde, ACCESS senare skall köras under.

Flera användare kan tilldelas samma arbetsområde, men då kan bara en åt gången utföra ACCESS-transaktioner. Om två försöker använda samma arbetsområde, resulterar detta i att den andra får ett felmeddelande.

Arbetsområdena identifieras med namnet på SINTRAN-filerna.

SINTRANS format för angivande av filnamn är:

<DIRECTORY>:<SINTRAN-ANVÄNDAR-NAMN><FILNAMN>:<FILTYP>;<VERSION>

DBA kontrollerar, att det uppgivna filnamnet är tillåtet. Men om filerna inte finns, och ACCESS inte har rätt att bygga dem, eller om directory-namnet är felstavat, kommer ett felmeddelande först, när ACCESS startas upp.

Directory och SINTRAN-användarnamn består av upp till 16 alfanumeriska tecken. Typ är 4 tecken, medan version är en siffra. Version används endast i mycket speciella fall. Om du arbetar på ditt eget SINTRAN-definierade användarområde, är det tillräckligt med <filnamn:typ>.

En mer fullständig definition av SINTRANS filnamn finns i SINTRAN-manualerna.

När arbetsområdet skall uppges, skall inte typ och version ges; SINTRAN användarnamn måste däremot specificeras, om arbetsområdet skall associeras med en bestämd SINTRAN-användare.

Om Sintran användarnamn inte uppges, bygger ACCESS arbetsfiler hos alla användare som ACCESS körs från. Om man således av misstag loggar in på fel Sintrananvändare, får man oönskade filer i detta användarområde.

Var uppmärksam på att man normalt kan läsa (kopiera) andra användares filer men inte skriva på dem. I så fall måste du vara definierad av berörd användare som 'FRIEND'.

När du senare uppger filnamnet för arbetsområdet kan du förkorta namnet så länge det är entydigt.

3.3 Definition av Databas-Administratör

DBA är en speciell användare av ACCESS systemet som har lov att bygga nya databaser, stryka gamla data, ändra existerande filer om det är nödvändigt, samt att se till att systemet används av alla på avsett sätt.

För att definiera en användare som en DBA i ACCESS, ges kommandot *DBA-ANVÄNDARE. Detta leder till att följande bild kommer upp på skärmen:

Databas-administratörer
Defin. DBA-användare i ACCESS

Fälten fylls i med ACCESS's användaridentifikation för de användare som skall utföra DBA-funktion.

Obs. att man får inte lov att redigera/stryka information om andra DBA-användare än en själv. Information om andra DBA "hoppas" över, som i EO-ANVÄNDARE kommandot.

3.4 Definition av Dataelementen

För att användarna av transaktionsdelen i ACCESS skall slippa att hålla reda på detaljer om lagring och framhämtning av data, görs detta i DBA-modulen en gång för alla.

Till alla dataelement knyts en typ. Den anger hur ett dataelement lagras, men begränsar också användandet. Dataelement av samma typ har det gemensamt att de kan representera varandra ("byta ut varandra").

För att definiera sådana typer av dataelement har vi en speciell kommandomodul. Vi kommer in i den genom att skriva:

*EDITERA-TYPER

Vi får då upp följande bild:

Datatyper		
Datatypnamn	Utskriftsformat	Lagringsformat

Vi ger alla datatyper vi definierar ett namn, som vi senare kan använda som referens i typbeskrivningen t ex för definition av dataelementen i en tabell.

I tabellen för Datatyper är det tillräckligt att fylla i ett av fälten plus namnet på typen. Genom att t ex fylla i fältet 'Utskriftsformat' låter man systemet själv finna ut riktigt lagringsformat, - och omvänt.

3.4.1 Specificering av lagringsformat

Lagringsformat, eller 'Internt format', definierar hur datatypens skall lagras. Men man begränsar då också användningssättet.

Här är en översikt över de tillåtna lagringsformat, som kan användas, och vilken lagring de motsvarar.

Lagringsformat	Möjlig aritmetik	Storlek på dataelementet
INTEGER	JA (Min/max värde: -32768 till 32767)	2 bytes/ 1 ord / 16 bits
DOUBLE INTEGER	JA (Min/max värde: -2 147 483 648 till + 2 147 483 647)	4 bytes/ 2 ord / 32 bits
REAL	JA - Upplösning 6 till 8 siffror	6 eller 4 bytes/ 3 eller 2 ord / 48 eller 32 bits *)
DOUBLE REAL	JA - Upplösning 14 till 20 siffror	8 bytes / 4 ord
DOUBLE PRECISION REAL		8 bytes/ 64 bits
CHARACTER(< n >)	NEJ	'n' bytes
TEXT(< n >)	NEJ	'n' bytes

*)Avhängigt av vilken CPU man kör på.

Om datatypen skall användas i SIBAS item-definition, måste storleken på dataelementet vara identiskt med LENGTH till dataelementet, som det definierats i SIBAS-DRL. Obs. att LENGTH anger antalet ord. Vilken SIBAS-typ, som valts i DRL, har ingen betydelse.

Exempel: 'no1' och 'no2' är definierade med lagringsformat REAL, och vi kör på en vanlig 48 bit floating point ND-100 CPU. De kan vara definierade i SIB-DRL som:

```
NEW ITEM <realmnamn> no1 TYPE integer START 1 LENGTH 3
```

```
NEW ITEM <realmnamn> no2 TYPE character START 4 LENGTH 3
```

Obs. att lagringsformatet i ACCESS måste stämma med det, som är definierat för databasen (och som andra användare använder när data läggs in).

De följande avsnitten används av DBA men ännu inte i ACCESS. Dvs. DBA fyller i och kontrollerar sambandet mellan de olika beskrivningarna, men ACCESS företar ingen formatering av utskriften och kontrollerar inte värden vid input, som det beskrivs här. Detta kommer att implementeras senare.

3.4.2 Specificering av Utskriftsformat (Utformat)

Utformat används för att beskriva, hur man önskar ett dataelement presenterat, t ex i resultatsammanhang. Endast en liten del av de möjligheter, man har här, används idag. Resten implementeras senare. Formatsyntaxen motsvarar den, som används i COBOL-program.

Ett 'A' visar, att här skall det vara ett alfanumerisk tecken, medan '9' och 'Z' anger ett numeriskt tecken (= siffra). Skillnaden mellan '9' och 'Z' är, att '9' skriver ledande nollor, medan Z undertrycker dem (skriver blanka).

'Z' och '9' kan blandas som i PIC-klausulen i COBOL, i motsats till fält definierade som alfanumeriska.

Repetition anges genom att placera indikatorerna efter varandra, eller genom att de efterföljs av ett tal inom parentes, som säger hur många gånger, de skall förekomma.

Blanka tecken, och text mellan apostrofer ('), kan förekomma var som helst vid angivande av Utformat.

För att kunna behandla tal lagrade som Real eller Double Precision Real, finns det ett speciellt format, 'E'-formatet. Det är avsett för allt, som man inte känner storleken av på förhand, och där det endast är ett visst antal siffror, som är av intresse. Fälten definieras som vanliga numeriska, men kan dessutom ges ett exponentfält, indikerat genom 'E'.

En fullständig COBOL PIC-sträng kan också användas.

3.4.2.1 Exempel på utformat

Utformat-beskrivning	Exempel på resultat	Lagring
'Lön : NOK ' ZZZ 999.99	Lön : NOK 94 345.00	Real
'Lön : NOK ' 999 999.99	Lön : NOK 094 345.00	Real
'Lön : NOK ' 9(3) 9(3)'. - '	Lön : NOK 094 345.-	Real el. Double Integer.
9.9(5) E 99	0.94350 E+6	Real el. Double Prec. Real
A(22)	Ole Olsen	Character(22)
A(5) A(4)' / 'A(9)	Norsk Data / ND-Comtec	Character(18)

Det måste råda samstämmighet mellan specifikationen av Utformat och det specificerade Lagringsformatet. Datatyper lagrade med format 'Character' (Text) kan endast visas som alfanumeriska fält, medan datatyper av format Integer, Real, Double Integer och Double Precision Real alla måste presenteras med en numerisk fältbeskrivning.

Om talet är för stort i förhållande till fältbeskrivningen, fylls fältet helt med stjärnor (*).

3.4.2.2 Kommentar till Lagringen av Data

Hur data lagras har betydelse för hur de senare kan användas. Varje lagringsformat upptar en viss plats i filerna. För att finna storleken av en fil beräknar vi längden av en post (record) och multiplicerar med antalet poster.

Lagringsformatet bestämmer till en viss grad storleken på posterna och därmed filstorleken, och hur effektivt det går att leta genom filen. Det säger sig självt, att ju mindre det dataelement är, som skall jämföras, ju fortare går sökningen. Men det är först och främst användningen, som bestämmer, hur ett dataelement skall lagras.

3.4.3 Standardvärden i fälten

DBA-modulen kontrollerar det du har fyllt i fälten med mot specifikationen av datatyp. Om du anger felaktiga data, får du en felutskrift.

De fält, som inte blir ifyllda, fyller DBA-modulen själv med standardvärden på basis av uppgivna data. Det är därför tillräckligt att fylla i två fält: namnfältet och ytterligare ett.

3.5 Sökning efter förekomst av olika typer

För att kunna ge DBA en översikt över användningen av de olika typer, som vid varje tidpunkt är definierade, används kommandot:

*VAR-ANVÄNDS-TYPEN < typ-namn >

Det är i många situationer bra att veta, om en datatyp är använd. En datatyp, som inte är använd, kan man då stryka. Det kan också vara bra att veta, var typen används, om man vill lagra en ny typ, och använda denna i redan definierade tabeller, där den nya typen har samma lagringsformat som den gamla. Man kan också tänka sig att använda kommandot för att finna ut, om två typer är nästan lika, om man önskar slå ihop dem.

Man har inte rätt att stryka en typ, som är i bruk.

3.6 Odefinierade Typer

För att hålla reda på alla typer definierade med hjälp av *EDITERA-TABELL-kommandot (se avsnitt 4.3), och som inte är beskrivna med *EDITERA-TYP kommandot, kan du använda kommandot:

*ODEFINIERADE-TYPER

En lista över alla typer som är använda i EDITERA-TABELL, och som inte är beskrivna genom EDITERA-TYP, kommer då fram. Till exempel kommer en datatyp, som du har använt i en tabell men har stavat fel ('ÄPPLER'), att skrivas ut.

Man ska inte använda en tabell, som innehåller odefinierade typer. Resultatet blir i såfall felaktigt.

3.7 Definition av en ny databas

En databas är ett samlingsbegrepp för en grupp filer som kan vara realmer i en SIBAS databas, indexerade ISAM-filer eller s.k. 'sekventiella' filer utan indexering. Posterna (recorden) i filerna måste ha fast längd.

ACCESS kan ge en användare tillgång till de data, som ligger i en sådan databas, genom att den beskrivs av DBA. En sådan beskrivning måste innehålla både information om hur data lagras, vem som skall få lov att använda informationen, samt hur informationen skall presenteras för användarna av databasen.

En ny databasbeskrivning genereras genom att man skriver:

*NY-DATABAS

Ge databasnamn: EXEMPEL-BAS

där namnet på den nya databasen ges som svar på fråga eller:

*NY-DATABAS EXEMPEL-BAS

där namnet på databasen skrivs direkt efter kommandot. Det ACCESS-användarnamn, du gav, då du gick in i DBA, registreras som ägare till databasen.

Detta kommando medför, att det avsätts en plats för en beskrivning av en ny databas med namn Exempel-bas.

För en SIBAS databas måste alla namn, också databasnamnet, vara identiskt med det namn, som är givet i SIBAS-DRL. Om du ger ett databasnamn på mer än 8 tecken, måste de första 8 vara helt lika de, som tidigare definierats av SIBAS, medan de resterande ignoreras ('trunkeras'), när ACCESS kallar på SIBAS.

Samma databas kan ha filer av olika typer (SIBAS, ISAM, och 'SEKV'). Om databasen innehåller SIBAS-realmer, måste det databasnamn, som används i ACCESS, vara överensstämmande med SIBAS-databasens namn. En eventuell omdefiniering av databasen är ett SIBAS Supervisor-jobb, som inte skall behandlas här.

När du refererar till en fil för första gången i ACCESS (genom att använda en tabell för första gången), och filen inte finns, kommer ACCESS, om det är möjligt att bygga filen som en indexerad SINTRAN-fil. Det är vanligtvis inte möjligt i SINTRAN att bygga en fil på en annan SINTRAN-användares område. ACCESS måste därför köras på den SINTRAN-användare, där den nya filen skall ligga, om inte två användare är definierade som 'Friends' med 'directory'-access till varandras filer.

När det gäller SIBAS, måste RT vara definierad som friend till den användare, som har SIBAS-databasen på sitt område. Om det är ett lösenord definierat för SIBAS-databasen, frågar ACCESS efter detta.

Obs. att DBA-modulen i ACCESS inte kommunicerar med filerna i databasen. Det förekommer därför heller ingen kontroll av att de upplysningar, du har gett, är riktiga. Iakttag därför försiktighet!

De upplysningar/data, du ger i DBA-modulen under användare DATA-DICTIONARY, lagras i filer på dennes användarområde och tillförs inte SIBAS-databasen eller filsystemet.

Tillgången till dessa tabeller bör vara begränsad, och vad en användare ser av databasen bör också kunna begränsas. Definition och ändring av tabellerna beskrivs i nästa kapitel.

Kommandot "*NY-DATABAS" medför att systemet avslutar behandlingen på denna nivå (Databasdefinitionsnivån) och går till Tabelldefinitionsnivån (nivå 2) för att definiera tabellerna i databasen.

3.8 Översikt över tillgängliga databaser

En lista över tillgängliga databaser i systemet kan man få genom att ge kommandot :

*LISTA-DATABASER

Namnen på alla databaser i systemet kommer då fram på skärmen.

3.9 Strykning av databas

En databasbeskrivning stryks från ACCESS-systemet genom kommandot:

*STRYK-DATABAS

följt av databasens namn. Om inte namnet ges, frågar systemet efter namnet som vid *NY-DATABAS-kommandot.

För att detta kommando skall kunna utföras, kräver systemet att det inte finns tabeller kvar i databasen, och att DBA, som företar strykningen, också är ansvarig för den aktuella databasen.

De datafiler, som är definierade att tillhöra databasen, berörs inte, men skall ACCESS kunna använda dem igen, måste de definieras på nytt i DBA-modulen.

3.10 Ändring i existerande databasbeskrivning

Oftast vill man ändra i existerande databasbeskrivningar hellre än att stryka och bygga en helt ny. Detta görs med kommandot:

*EDITERA-DATABAS

Namnet på databasen kan antingen ges tillsammans med kommandot eller som svar då systemet frågar.

Beskrivningen av den önskade databasen laddas då, och systemet går ner till Tabellnivån (nivå 2).

3.11 Avslutning på definitionsnivån (1)

Arbetet på Definitionsnivån (nivå 1) avslutas, och man går tillbaka till SINTRAN genom att ge kommandot

*EXIT

eller genom att använda 'EXIT'-tangenten.

4 TABELLDEFINITIONSNIVÅ

Tabelldefinitionsnivån (nivå 2) används för att definiera, stryka och ändra i tabeller eller filer. Du kommer till denna nivå efter kommandot

*NY-DATABAS och *EDITERA-DATABAS.

Allt arbete på denna nivå förutsätter alltså, att användare och databas är definierade.

4.1 Tillgängliga kommandon

En lista över kommandon på denna nivå kommer upp, om man ger kommandot:

**HJÄLP

Kommando	Betydelse
HJÄLP	Ge information om tillåtna kommandon.
EXIT	Lämna tabelldefinitionsnivån.
EDITERA-TYPER	Editera beskrivning av variabeltyper.
VAR-ANVÄNDS-TYPEN	Visa var en typbeskrivning används.
ODEFINIERADE-TYPER	Lista datatyper som är använda, men odefinierade.
EDITERA-TABELL	<tabell> Editera en tabellbeskrivning: ISAM/SIBAS/SEKV
STRYK-TABELL	<tabell> Stryk en tabell.
LISTA-TABELLER	Lista alla tabeller i databasen.
EDITERA-BEHÖRIGHET	Editera de användare som har behörighet.
ISAM-FILER	Editera beskrivningen av alla ISAM-filer.
SIBAS-REALMER	Editera beskrivningen av alla SIBAS-realmer.
SEKV-FILER	Editera beskrivningen av alla SEKV-filer.
SIBAS-ITEMS	<tabell> Editera beskrivningen av en SIBAS-tabell.

4.2 Tillgång till tabellerna

För att ACCESS skall tillåta en användare att köra transaktioner mot en databas, måste vederbörande vara definierad som användare av databasen. Detta görs med kommandot:

**EDITERA-BEHÖRIGHET

I och med att databasen redan är definierad på nivå 1, behöver inte systemet veta namnet. Följande bild kommer då fram på skärmen:

Användarbehörighet				
Användarnamn	Skriv.	Ny.	Uppdatera.	Stryk.

Under användarnamn skriver du namnet på de ACCESS-användare, du vill skall ha tillgång till databasen. I kolumnerna under varje ACCESS-kommando skriver du JA eller NEJ beroende på, om du tillåter att denna användare skall kunna använda detta kommando i ACCESS eller ej. Obs. att blank är lika med 'NEJ', men DBA (ägaren) har likväl obegränsad tillgång.

4.3 Definition av dataelementen i tabellen

För att ett element ska kunna vara en del av en tabell måste dess typ först definieras. Vi använder då kommandot:

**EDITERA-TYPER

Detta kommando är tidigare beskrivet i sektion 3.4.

4.4 Definition och redigering av en tabell

Ge kommandot:

**EDITERA-TABELL

DBA-systemet frågar då efter tabellnamn. Om den tabell du uppger namnet på existerar, får du fram den, och du kan redigera och ändra i den. Finns det ingen tabell med det namn du uppgett, frågar systemet, om du önskar bygga en ny tabell. Svarar du NEJ på detta, ignorerar DBA-modulen kommandot. Svarar du JA, bygger DBA-modulen en ny tabellbeskrivning, dvs. den frågar efter vad slags tabelltyp du vill definiera. Du kan då välja mellan ISAM, SIBAS och SEKV. Se upp, så att du specificerar en tabelltyp, som din version av ACCESS kan använda. En bokstav är tillräcklig.

Följande bild kommer då fram på skärmen med tomma kolumner:

Tabellnamn		
Elementnamn	Datatyper	Nr.

Tabellen skall fyllas i med information om de element, vi önskar använda i vår tabell. Tabellen identifieras sedan med tabellnamnet.

Namnet på elementen kommer senare upp som kolumnnamn, när vi använder ACCESS för att utföra transaktioner på tabellen.

Typnamn fylls i med det namn, som användes i EDITERA-TYP-kommandot.

Nr. fylls i för att indikera vilken ordning dataelementen skall ha i arbetstabell och resultattabell. För ISAM- och SEKV-filer bestämmer de också den fysiska ordningen på filen.

Detta var vänstra sidan av arbetstabellen för definition och ändring av tabeller. Genom att vi fyller i arbetstabellen fält för fält, flyttar den sig åt höger, så att vi till slut får fram följande arbetstabell:

Tabellnamn			
Nr.	Nyckel	Kolumnbredd	Radlängd

Nr. är samma fält som ovan.

4.4.1 Nyckel

Nyckel används för sökning och kan vara antingen PRIMÄR eller SEKUNDÄR. För 'sekventiella' filer är detta fält utan betydelse.

Primär - Indikerar att dataelementet används som primärnyckel. Det vill säga att dataelementet måste vara entydigt (dvs. ha ett unikt värde) för hela filen. Nyckelfilen innehåller adressen till alla nyckelvärden, så att sökningen går snabbare. Primärnyckeln definierar den logiska ordningen i filen, även om posterna rent fysiskt har en annan ordning.

För element i SIBAS-realmer definieras elementet som primär (P), om elementet definierats med 'duplicates not allowed', eller som sekundär (S) om 'duplicates allowed'.

En ISAM-tabell måste ha minst ett dataelement definierat som PRIMÄR. Högst 6 dataelement kan vara definierade som nycklar.

Sekundär - Indikerar att dataelementet är en sekundärnyckel.

I en telefonkatalog kan man säga att efternamn och förnamn är två sekundärnycklar. Det kan finnas flera med samma efternamn, liksom det kan finnas flera med samma sekundärnyckelvärde. Förnamn kan man då använda som nästa nivå att sortera efter, så att man slutligen har identifierat den eller dem man söker.

Det räcker att ange "S" eller "P" i nyckelfältet. Alla andra värden, inklusive felstavningar, indikerar att dataelementet inte skall knytas till en söknyckel. Det blir då mer tidskrävande att leta efter detta element, enär sökningen måste ske sekvenssiellt.

Å andra sidan tar insättning och uppdatering mer tid och större lagringsutrymme, när ett fält är definierat som söknyckel.

4.4.2 Kolumnbredd och radlängd

Kolumnbredd används för att uppge den initiella startbredden för kolumnen i ACCESS' arbetstabell. Kolumnen får den angivna bredden eller den bredd, som behövs, för att elementets namn skall få plats.

Kolumnbredden i ACCESS's arbetstabell kan du ändra i ACCESS med:

< > - tangenten (= CTRL/W) för att öka bredden eller

> < - tangenten (= CTRL/P) för att minska bredden.

efter det att arbetstabellen har kommit upp på skärmen.

Radlängden är längden av kolumnen, när resultatet av en fråga skrivs ut på standard tabellformat. Till exempel:

Ett textfält med längd 80 tecken och radlängd 20 kommer att fylla 4 rader å 20 tecken. Resultatet visas då med minst 4 rader per post. Antalet rader i tabellen bestäms av det fält som upptar flest rader.

Ominget uppgessörjer ACCESS för att resultatet blir förnuftigt på skärmen.

4.5 Tabeller och filer i ACCESS

ACCESS bygger själv alla nödvändiga filer (undantaget de filer som SIBAS-databasen ligger på). Man måste då köra på det användarområde, som senare skall äga filerna. Du kan emellertid också själv bygga dessa filer med kommandot:

Logga in på önskat användarområde.

För ISAM-filer, används kommandot:

ÉCRE-FILE <filnamn>:<typ>,0
ÉCRE-FILE <filnamn>:ISAM,0

För 'sekventiella' filer:

ÉCRE-FILE <filnamn>:<typ>

där <typ> kan vara DATA eller ett annat begrepp om högst fyra bokstäver, men som måste överensstämma med det, som är definierat som tabellens Sintran-filnamn i kommandot **SEKV-FILER eller **ISAM-FILER. Se följande avsnitt.

Om databasen skall uppdateras via ACCESS av en annan användare, måste du ha definierat en sådan access som tillåten.

När det gäller SIBAS-filer, måste RT vara definierad som 'friend' till alla användare, som är ägare av filer, där SIBAS-realmer är utlagda.

Du kan förkorta filnamnet så länge det är entydigt för SINTRAN. Obs. att felstavade filnamn leder till att ACCESS bygger nya filer - med felstavade namn!

4.6 Knytning av DDI's tabellinformation till en fysisk ISAM-fil

Informationen om en tabell är bara en beskrivning, som lagras i DDI för senare användning i ACCESS. För att denna information skall ha något värde, måste den knytas till en fysisk fil som man kan utföra transaktioner på. Detta görs genom två kommandon. Det första är:

**ISAM-FILER

Detta resulterar i att följande skärmbild kommer upp:

ISAM-filnamn		
Tabellnamn	ISAM-id	SINTRAN filnamn

Under tabellnamn kommer namnen på alla definierade tabeller av typ ISAM i den databas, du håller på med för tillfället, fram. Om det är den första tabellen du definierar, kommer endast denna upp.

Under ISAM-id kommer det fram två bokstäver som används av ISAM för att identifiera denna tabell. Inom en databas kan inte två tabeller ha samma ISAM-identifikationskod.

ISAM använder två SINTRAN-filer för att lagra nycklar och data på. Dessa skall ha samma namn men olika filtyp: nyckelfilen har alltid typ ISAM, medan du kan välja filtyp för datafilen.

I fältet under SINTRAN-filnamn skall namnet på datafilen skrivas. Utelämnar du filtyp här, får datafilen :DATA som filtyp. Se annars avsnitt 3.2 under ARBETSOMRÅDE hur SINTRAN filnamn anges. Kom ihåg att ta med SINTRAN-användare i filnamnet, annars kan ACCESS få problem med att lokalisera filerna.

Obs. att ett tabellnamn kan ändras genom en ändring i Tabellnamn-kolumnen. En tabell kan också strykas här genom att använda SKIFT + STRYK-tangenten (eller trycka <D> + <L>). Detta har samma verkan som kommandot **STRYK-TABELL.

4.7 Definition av SIBAS-tabeller

Tabellerna definieras, som alla andra tabeller, med kommandot

****EDITERA-TABELL**

och arbetstabellen som kommer fram fylls i som beskrivs i 4.4

Obs.: Lagringsformatets längd måste överensstämma med den längd, som getts vid definition av SIBAS-databasen i SIBAS definitionsmodul (SIBAS-DRL).

Attributnummer behöver inte definiera samma ordning som är definierad i SIB-DRL.

Man behöver inte använda alla nycklar, som är definierade i SIBAS-databasen. Men endast de element, som är definierade som nycklar, ger optimal sökhastighet.

Man kan i nuvarande version av ACCESS endast referera till enkla datavärden, 'single items', även om SIBAS-databasen gott kan vara definierad med 'group items'.

Filerna i SIBAS (realm-namnen) definieras med kommandot

****SIBAS-REALMER**

och man får upp tabellen:

SIBAS realmnamn	
Tabellnamn	SIBAS-realm
löner	loen

Obs. att både Tabellnamn och SIBAS-realm kan ändras genom skrivning av ett nytt värde i kolumnen. En SIBAS-realm kan strykas med:

SKIFT STRYK-tangenten (= CTRL/D CTRL/L)

(gör det samma som **STRYK-TABELL).

4.8 Knytning av Attributnamn till SIBAS-itemnamn

För att ACCESS skall slippa begränsningen på åtta bokstäver på SIBAS-item namn, ges det en möjlighet att definiera Itemnamn, som skall användas, oberoende av det attributnamn, som använts i tabellbeskrivningen. Detta sker med kommandot

****SIBAS-ITEM**

som ger följande tabell:

ANSTÄLLDA	
Alementnamn	SIBAS-item
fullt namn	namn
chef	chef
lön	lön
avdelningsnamn	avdelning

Både Attributnamn och SIBAS-item kan ändras, genom att nytt värde skrivs i kolumnerna. Attribut kan strykas endast med ****EDITERA-TABELL-**kommandot.

Obs. att namn på realmer och items i SIBAS består av högst 8 tecken (bokstäver och/eller siffror) och måste vara identiska med de, som definierades då databasen byggdes med SIBAS-DRL.

4.9 Användning av 'sekventiella' filer

En 'sekventiell' fil är en fil med fast postlängd och utan CR/LF mellan posterna. Det kan inte ges några söknycklar till en sådan fil, utan varje gång ACCESS skall finna något i filen, måste hela filen genomsökas från början. Sådana filer används likväl ofta, därför att de är enkla att definiera och använda. En tabell av 'sekventiell' typ definieras som tidigare beskrivet med den skillnaden, att inga nycklar ska definieras i nyckelkolumnen. Tabellen för att definiera innehållet är annars helt lik den, som används för de andra tabelltyperna.

Filnamnet (i SINTRAN) kan omdefinieras med hjälp av kommandot ****SEKV-FILER**. Följande tabell kommer upp:

SEKV filnamn	
Tabellnamn	SINTRAN-fil

Tabellnamn och SINTRAN-fil kan förändras genom insättning av ett nytt värde i kolumnerna. Obs. att en tabell kan strykas med

SKIFT STRYK-tangenten (= CTRL/D CTRL/L)

eller det samma som **STRYK-TABELL. Kom ihåg att ta med Sintrananvändaren i filnamnet!

4.10 Strykning av tabellbeskrivning

Genom att ge kommandot:

****STRYK-TABELL**

med tabellnamn som parameter stryks tillhörande tabellbeskrivning. Själva filen berörs inte. Tabellen kan förekomma i andra beskrivningar, och alla beskrivningar, där tabellen ingår, stryks. T ex kommer en struken tabell inte fram i bilden efter kommandot **ISAM-FILER, även om ISAM-filen fortfarande existerar.

4.11 Översikt över tillgängliga tabeller i databasen

En lista över tabellerna i databasen kan man få genom att ge kommandot:

****LISTA-TABELLER**

4.12 Avslutning på tabelldefinitionsnivå (nivå 2)

Nivå 2 - tabelldefinitionsnivån avslutas, och man kommer upp till Databasdefinitionsnivån (nivå 1) genom att trycka:

****EXIT**

Från nivå 2 måste man alltså trycka 'EXIT' två gånger för att komma tillbaka till SINTRAN.

5 Allmänt om körning av ACCESS

5.1 Terminaltyp

Om terminaltypen av en eller annan orsak är fel angiven, blir texten på skärmen mer eller mindre förvanskad.

Gå tillväga på följande sätt för att sätta terminaltyp:

- 1) Ge SINTRAN-kommandot i ACCESS eller DBA:

```
ÉSET-TERM-TYPE ,0
```

- 2) En meny över tillgängliga terminaltyper skrivs då ut. Uppge rätt typ! För terminaltyp 52 måste terminalen också ha Extended Control Mode ON (ta fram meny 2). Se handledningen för terminalen!

5.2 Kommandofilerna

Systemet använder kommandofilerna ACCESS-HELP:SYMB och DBA-HELP:SYMB. Innan systemet tas i bruk, måste önskad språkversion kopieras till dessa filer från filerna ACCESS-SWEDE:SYMB eller ACCESS-ENGL:SYMB, och DBA-SWEDE:SYMB eller DBA-ENGL:SYMB - som levereras med systemet.

Varje SINTRAN-användare kan ha sin personliga version av ACCESS-HELP:SYMB på sitt område, så att han kan köra ACCESS på det språk han önskar. Någon vill kanske ha svenska, andra engelska. Det är också möjligt att använda PED eller NOTIS-WP på HELP-filerna för att förändra ordalydelsen på kommandon, både i ACCESS och DBA. Du kan också själv översätta hela HELP-filerna till ett annat språk (t ex finska eller samiska), så att kommandon, meddelande och hjälp-information blir på detta språk. Det är då viktigt att inga rader tas bort eller läggs till i HELP-filerna. Du får bara skriva över de gamla raderna.

Du bör för övrigt hindra otillåten ändring av HELP-filerna genom att ge dem 'read-only'-access. Om kommandona ständigt får ny lydelse, blir det problem för ACCESS-användarna att följa med!

Om ACCESS-HELP inte finns på det SINTRAN användarområde, som du är inloggad på, användas alltid den version av ACCESS-HELP, som ligger hos användare DATA-DICTIONARY.

5.3 NO SUCH FILE

Detta meddelande ges, om ACCESS inte finner de filer, det behöver för att systemet skall fungera.

En av dessa filer är den fil, där terminalen du arbetar med, är beskriven (DDBnnn:VTM filerna som ligger under användare SYSTEM). Terminaltypen motsvarar <nnn> i DDBnnn:VTM.

5.4 Fel på användarens temporära arbetsområde

Detta meddelande ges om användarens arbetsområde inte är tillgängligt för skrivning från ACCESS. Du kan enkelt ändra på detta genom att gå in hos den SINTRAN-användare, där arbetsområdet ligger, och ge ett av kommandona:

```
ÉSET-FILE-ACCESS <filnamn>  
public:RWA  
friend:RWA  
own:RWACD
```

eller:

```
ÉCREATE-FRIEND < namn på SINTRAN-användare >
```

Har du till exempel definierat användare 'HANS' i EDITERA-ANVÄNDARE, med arbetsområde i filerna (HANS-DALE)ARBETSOMRÅDE:DATA och :SORT, måste filerna under SINTRAN- användare 'HANS-DALE' vara tillgängliga när HANS kör ACCESS. Skall användare OLE också använda HANS' filer för skrivning, måste HANS ha gjort 'OLE' till 'FRIEND' (obs inte motsatt).

Meddelandet kan också komma om det används flera SINTRAN MAIN-DIRECTORIES. Se SINTRAN Timesharing/Batch Guide, ND-60.132 kapitel 3.

För att finna ut vilka "FRIENDS" som redan är definierade finns kommandot:

```
ÉLIST-FRIENDS...
```

De användare som är definierade har då, om inget annat specificerats, RWA-access till denna användares filer.

5.5 DDI inte aktiv

Detta meddelande fås om en användare försöker att använda ACCESS, utan att DDI körs som en batchprocess. Det kan tänkas att man har glömt att starta batchprocessen DDI-MACHINE:BATC (se kapitel 2). Men det kan också bero på att ACCESS inte får någon kontakt med DDI. T ex kan de nummer som uppgetts för kommunikations-devicen i ACCESS-HELP och DDI-MACHINE:BATC-filerna inte stämma överens, eller inte finnas i operativsystemet (se nästa avsnitt).

5.6 Kommunikationsproblem mellan DDI och ACCESS

ACCESS kommunicerar med DDI genom att använda en funktion i SINTRAN som kallas INTERN-DEVICE. Ett Intern-device är ett bufferområde i SINTRAN som används för att överföra data mellan processer. En process reserverar buffern för läsning, en annan för skrivning. För att ACCESS skall kunna kommunicera med DDI, behövs alltså två sådana "DEVICE".

Alla device är tilldelade nummer vid generering av SINTRAN (Se längst bak i SINTRANmanualerna). ACCESS använder vanligtvis device 128 och 129 för kommunikation med DDI, men dessa kan vara upptagna av andra system, och då måste ACCESS använda andra device.

Vilka device som används, står först i filen "ACCESS-HELP" och i filen "DDI-MACHINE:BATC". Numren här måste stämma överens!

Om dessa device är upptagna av andra system, kan lediga SIBAS-device användas (från 160 till 166 ev. 172 (240B till 246B ev. 254B)). Kom ihåg att ändra i båda de filer, där numren på intern-devicen är definierade.

5.7 Allmänt om användningen av filer

I SINTRAN har varje användare sitt tilldelade dataområde på massminnet (disken). Alla användare är definierade i ett 'Main Directory', eventuellt i andra directories. En katalog motsvarar ofta en fysisk diskenhet. Om en användare har tilldelats plats i mer än en katalog, måste namnet på katalogen (directory name) vara med vid specificering av filnamn.

Normalt kan man läsa och kopiera andras filer. Däremot kan man inte skriva eller förändra på en annan användares filer. Om du skall skriva på en annan användares filer, kan vederbörande definiera dig som 'Friend' med kommandot ÉCREATE-FRIEND, och du kommer att få standard tillgänglighet för 'friend': RWA (read-write-append).

Tillgången till en användares filer kan ändras med två kommandon :

ÉSET-FRIEND-ACCESS

och

ÉSET-FILE-ACCESS

I SINTRAN är (som regel) en katalog huvudkatalog ("main directory"). Vänner kan då definieras mellan alla användare i huvudkatalogen, även om inte alla vännerna har sina filer fysiskt på denna katalog. Det är tillräckligt, att användarna är definierade i huvudkatalogen.

5.8 Förbättringar i sökhastigheten för ACCESS

En SINTRAN-fil är ett dataområde på disken som kan vara sammanhängande ('contiguous') eller indexerat ('indexed'). Sökning sker snabbare i en sammanhängande fil. Nackdelen är att storleken på en sammanhängande

fil måste bestämmas på förhand, och att filen inte utan vidare kan utvidgas. En indexerad fil utvidgas efter behov (dynamiskt). ~~men~~ sökningen baseras här på en indextabell över sidorna. Indextabellen underhålls av Sintran filsystem.

Bägge slagen av filer (som inte har något med filtyp att göra) skapas med kommandot:

ÉCREATE-FILE <filnamn>,<antal sidor>

om <antal sidor> sätts=0, blir filen indexerad. Annars erhålls en sammanhängande fil.

ISAM och SIBAS är skräddarsydda för snabb hantering av sammanhängande filer, enär adresserna till posterna ligger i indextabellen för filerna, och man slipper således uppslag först i filsystemets indextabell och sedan i SINTRANS indextabell.

När en sammanhängande (contiguous) fil är full, måste den utvidgas om detta är möjligt. Om inte måste filen omdefinieras genom en s.k. 'unload' och 'load' operation. Så det man vinner i tid genom snabbare exekvering av transaktionerna, måste vägas mot behovet av flexibilitet.

Obs. att en indexerad fil gott kan kopieras till en sammanhängande fil. Likaså kan en sammanhängande fil kopieras till en indexerad fil.

Det har utvecklats hjälpmedel för underhåll av SIBAS- och ISAM-filer. De är inte integrerade i denna omgång av DBA-modulen. Om fel uppstår i filerna, t ex att poster försvinner, eller om det kommer underliga felmeddelanden, kan man för ISAM-filer använda programmet 'ISAM-SERVICE' för att kontrollera om filerna är i ordning ('konsistenta'). Man kan också företa s.k. 'unload'/'load'-operationer från detta subsystem.

Den generella dokumentationen av operativsystemet SINTRAN är omfattande. Det är därför bra att veta lite mer om filsystemet än det som beskrivs här, om du skall kunna utföra ett gott DBA-jobb. Time-sharing/ Batch Guide (ND-60.132) beskriver de flesta funktionerna i filsystemet.

5.9 Andra felmeddelanden

När ett felmeddelande kommer från ACCESS, är det en klartext till användaren, som kan vara efterföljd av en felkod. Denna felkod är ett decimalt tal, inte oktalt.

Felkoden motsvarar SINTRANS felkoder, vilka finns beskrivna i ACCESS användarhandledning.

SIBAS och ISAM kan också generera felkoder (beskrivet i SIBAS- och ISAM-manualer). Speciellt med SIBAS är att systemet skriver ut felstatus på konsolen ('error device'), om något allvarligt sker. ISAM försöker alltid återvända till ACCESS med felstatus, men det kan förekomma situationer, där ISAM 'ger upp'. Ett meddelande motsvarande

detta skrivs då ut:

ISAM FATAL ERROR IN ROUTINE 'ZABUF'/1

och kontrollen överlämnas till SINTRAN. I så fall bör du kontrollera, att filen inte omstrukturerats sedan sist, att det är rätt fil ACCESS kör mot, osv.

5.10 Terminalen 'hänger'

Skulle systemet 'hänga', dvs. ingenting sker, kan du prova följande åtgärder, tills du får 'liv' igen:

- Kontrollera lamporna på terminalen. Om alla är mörka, kan strömmen ha gått. (Kontrollera att kontakten inte har ramlat ut. Byt säkring om nödvändigt. Kontakta annars systemansvarig.)

- Tryck på HEMMA-tangenten.

- Tryck på ^{SLUT}EXIT-tangenten eller två gånger på ESC-tangenten.

- Tryck <Q>. Om din terminal kör med 'handshake', (se avsnitt om detta i beskrivningen för laddning av ACCESS från FLOPPY) kommer <S> att stoppa all utskrift till skärmen. Den ser då död ut, men utskriften kommer igen, när du trycker <Q>.

- Tryck på LINE-tangenten (eller LOKAL-tangenten), tryck så <C> och så LINE (eller LOKAL) en gång till. Om skärmen har fått ett styrtecken, kan det hända att den har fått , dvs Video OFF, som slår av bilden på terminalen. I lokal-mode kommer <C> dvs. Video ON, att slå på bilden igen.

- Slå terminalen av och på och tryck på HEMMA-tangenten och hjälptangenten (eller <G>). Om din terminal borde ha använt 'handshake' men inte gör det, kan den 'hänga' sig, och denna procedur kan få den att fungera synligt igen.

- Kontrollera med andra terminaler om hela maskinen har stannat. Be den som är ansvarig för maskinen att starta upp den om nödvändigt.

- LOGGA IN på en annan terminal som användare RT, om du kan lösenordet. Finn ut numret genom ÉTERM-STAT,,, på den terminal som hänger och ge kommandot ÉSTOP-TERMINAL <nummer>. Då bör ACCESS-användaren kunna LOGGA IN på nytt och starta om från början.

6 FULLSTÄNDIGA EXEMPEL

6.1 Exempel på registrering av databasadministratör och användare

Först loggar vi in som användare DATA-DICTIONARY. Kontrollera i förväg, om någon använder ACCESS, med hjälp av kommandot ÉTERMINAL-STATISTICS, och ge ev. användare besked med MAIL-systemet. Skulle likväl någon användare bli 'hängande', kan han frigöras med kommandot ÉSTOP-TERMINAL (under användare SYSTEM).

Kontrollera att DDI är passiv med kommandot:

ÉLIST-BATCH-PRO,...

Om DATA-DICTIONARY visar sig vara användare av en batch-process, stryks denna genom kommandot:

ÉABORT-JOB < Batch-nummer >, DATA-DICTIONARY

Om någon vid denna tidpunkt använder ACCESS, kommer vederbörande att 'hänga'.

DBA-modulen startas så genom att man skriver:

ÉDBA

Vi förutsätter nu att systemet är levererat med den test-databas, som används i manualen. Då är redan en DBA registrerad med namnet ACCESS, utan lösenord.

Det frågas nu efter användaridentifikation, och vi svarar A, som är en förkortning för användaren ACCESS nämnt i föregående avsnitt. (Om ACCESS hade varit registrerad med lösenord, hade detta också efterfrågats.)

Vi får nu (efter en väntelid) upp en hjälp-bild, med översikt över de tillgängliga kommandon, vi kan starta med.

Först måste vi specificera identifikation för alla, som skall ha rätt att köra ACCESS-delen av systemet. Därefter måste vi ange vilka av dessa, som skall tillåtas att köra DBA-modulen, dvs. DBA-användarna.

Vi ger kommandot

*ED-A

som är en förkortning av *EDITERA-ANVÄNDARE. Vi får upp en tabell över användare där ACCESS finns med. Vi registrerar MIG-SJÄLV och ALBERT-VON-NEUMANN som nya användare, varefter tabellen innehåller:

ACCESS System användare		
ACCESS användarnamn:	Passord:	Arbetsområde(:DATA och :SORT)
ACCESS		(ACCESS)WORKAREA
MIG-SJÄLV	MI	(ACCESS)OMRAADE-1
ALBERT-VON-NEUMANN		OMRAADE-2

Användaren MIG-SJÄLV måste hädanefter uppge lösenordet MI, när han vill in i ACCESS, medan ALBERT-VON-NEUMANN inte frågas om lösenord.

Dessutom måste MIG-SJÄLV alltid logga in sig under SINTRAN-användaren ACCESS (eller en användare som är registrerad som 'friend' hos denne), annars kommer han inte in på sitt arbetsområde. ALBERT-VON-NEUMANN däremot kan logga in sig, var han önskar, emedan någon SINTRAN-användare inte specificerats för hans arbetsområde. I gengäld kommer det att upprättas två filer på varje SINTRAN-användarområde han loggar in sig på för att köra ACCESS! Så det kan bli många filer av hans arbetsområde.

Vi går ut ur tabellen genom att använda HEMMA-tangenten.

Vi önskar nu, att MIG-SJÄLV skall registreras som DBA, dvs. få lov att använda DBA-modulen och ha 'egna' databaser i ACCESS, medan ALBERT-VON-NEUMANN enbart skall köra ACCESS mot databaser, som andra har registrerat.

Kommandot

*DBA

som är en förkortning för kommandot *DBA-ANVÄNDARE, ger en tabell över databas-administratörer. Användaren ACCESS är registrerad sen tidigare, och vi för upp MIG-SJÄLV i förkortad version:

Databas Administratörer	
Defin. DBA-användare i ACCESS	
ACCESS	
M	

När du nu går ut ur tabellen genom HEMMA, skrivs det fullständiga namnet MIG-SJÄLV in automatiskt i stället för M.

Nu måste vi gå ut ur DBA-modulen, genom att trycka EXIT. När vi definerar en databas, kommer nämligen det användarnamn, vi uppgav, då vi startade DBA-modulen, att användas som namn på 'ägaren' av databas-beskrivningen senare. Vi var nu inne som DBA-användare ACCESS, och om vi vill att MIG-SJÄLV skall vara ansvarig för den databas vi skall definiera, måste vi gå ut ur DBA-modulen och in igen (genom ÉDBA), så att MIG-SJÄLV är som den som startar DBA-modulen.

Obs. att om inte test-databasen levereras med systemet, eller du är inloggad på ett annat användarområde än DATA-DICTIONARY (jmf. avsnitt 2.3), kommer du första gången direkt in i tabellen över ACCESS-användare, när du skriver ÉDBA. I denna tabell står SYSTEM som enda

användare. Du måste skriva över ordet SYSTEM med användarnamnet MIG-SJÄLV och så fylla i resten av tabellen, som du gjorde ovan.

Eftersom SYSTEM ursprungligen stod som DBA-användare också, kommer en ändring av namnet till MIG-SJÄLV i användartabellen automatiskt att leda till motsvarande ändring i tabellen över DBA-användare. M.a.o behöver du inte att anropa *DBA-ANVÄNDARE för att sätta upp MIG-SJÄLV som DBA i detta fall.

6.2 Exempel på definition av en databas

Vi önskar definiera databasen EXEMPEL-BAS med MIG-SJÄLV som 'ägare', dvs. som den DBA-användare, som får lov att använda DBA-modulen på denna databas.

Från SINTRAN kommer vi in i DBA-modulen genom att skriva

ÉDBA

och vi uppger användarnamn M-S (förkortning för MIG-SJÄLV) och lösenord MI. Vi kommer in på databasdefinitionsnivån (med en * längst ner), och ger kommandot:

*NY-DATABAS EXEMPEL-BAS

Genom detta lämnar vi databasdefinitionsnivån, och kommer ned till tabelldefinitionsnivån - nivå 2.

Här måste vi definiera några datatyper för dataelementen. Vi ger då kommandot:

**EDITERA-TYPER

och får upp en arbetstabell på skärmen för ifyllnad:

Datatyper		
Datatyper :	Utskriftsformat	Lagringsformat
PERSON-NAMN	A(32)	TEXT(32)
PENGAR	ZZZ ZZ9.00	LONG INTEGER
AVD-NAMN	A(20)	TEXT(20)

Vi fyller bara i fältet för typnamn och ett fält till, t ex lagringsformat, och låter systemet självt välja värden för de resterande två fälten.

Vi kan nu definiera en tabell, som vi senare kan fråga på. Denna tabell vill vi skall vara lik exempeltabellen i avsnitt 4.4.

För att definiera tabellen, ger vi följande kommando :

*NY-TABELL ANSTÄLLDA

ED-TAB ?!

och vi får då upp skärmbilden, som tidigare beskrivits, och fyller i den som så:

ANSTÄLLDA		
Elementnamn :	Datatyper :	Nr.
NAMN	PERSON-NAMN	1
CHEF	PERSON-NAMN	2
LÖN	PENGAR	3
AVDELNING	AVD-NAMN	4

ANSTÄLLDA			
Nr.	Nyckel	Kolumnbredd	Radlängd
1	P		
2	S		
3			
4	S		

För Kolumnbredd och Radlängd låter vi systemet välja värden.

Så måste ISAM-filerna definieras. Detta görs genom att använda kommandot :

**ISAM-FIL

Och vi fyller i den tillhörande bilden:

ISAM-filer		
Tabellnamn	ISAM-id	SINTRAN filnamn
ANSTÄLLDA	AN	(ACCESS)ANSTALLD

Nu är tabellen "ANSTÄLLDA" knuten till ISAM-filerna "ANSTALLD:DATA" och "ANSTALLD:ISAM" hos Sintran-användare ACCESS. Den sistnämnda filen innehåller alla nycklar, medan den första innehåller de data, som vi lägger ut på filen.

Nu är filen definierad, men vi måste också definiera, vem som skall få rätt att använda ACCESS på filen. Vi använder då kommandot **EDITERA-BEHÖRIGHET och fyller i följande arbetstabell:

Användarbehörighet				
Användarnamn	Skriv.	Ny.	Uppdatera.	Stryk.
MIG-SJÄLV	JA	JA	JA	JA
ALBERT-VON-NEUMANN	JA	JA	JA	NEJ

Användaren "MIG-SJÄLV" ger vi fullständig behörighet till användning av ACCESS genom transaktionshantering, medan användaren "ALBERT-VON-NEUMANN" inte får lov att stryka data, som är inlagda i databasen.

Obs. att nu kan kommandona *LISTA-DATABASER och *EDITERA-DATABAS användas eftersom det nu existerar en databas och en tabell.

Vi är färdiga med att definiera en tabell, och går tillbaka till SINTRAN genom att ge kommandot EXIT två gånger.

För att ACCESS skall kunna få tag i de beskrivningar, vi har byggt, aktiverar vi DDI-processen genom att ge följande kommando:

ÉAPPEND-BATCH 1,DDI-MACHINE:BATC,DDI-MACHINE:SYMB

Om detta är inlagt som termineringskommando till SINTRAN i filen DBA-HELP, slipper vi utföra detta kommando varje gång vi går ut ur DBA-modulen, se avsnitt 2.1.

Nu är definitionen av databasen färdig, och vi kan köra ACCESS mot tabellen ANSTÄLLDA som i exemplen i 'ACCESS Användarhandledning'.

Filerna,arbetsområdet och ISAM-filerna överläter vi nu till ACCESS självt att upprätta. Vi avslutar vår DBA-session med att logga ut från användare DATA-DICTIONARY och kan ev. börja arbeta med ACCESS genom att logga in igen med vårt vanliga användarnamn.

LYCKA TILL MED ACCESS !

B I L A G A

DDI-PROCESSEN SOM BATCH-PROCESS

Bilaga
DDI-PROCESSEN SOM BATCH-PROCESS

Under körningen av ACCESS använder ACCESS ofta uppgifter om databasen och dina behörigheter. Denna databasbeskrivning, som byggts genom körning av DBA-modulen, ligger normalt på filer hos användare DATA-DICTIONARY och är gemensam för alla dem som kör ACCESS.

För att ordna kön av användare, hålls filerna öppna av en självständig process i SINTRAN, DDI-processen. De olika ACCESS-användarna måste sända sina frågor om databasen till denna process via ett intern-device, och DDI-processen sänder så upplysningar tillbaka via ett annat intern-device.

DDI-processen är inte något annat än ett vanligt program, som aldrig stannar av sig självt, när det en gång startats. Programmet finns på filen (DATA-DICT)DDI-MACHINE:PROG. Programmet kan startas från en terminal, men belägger då denna så länge det går. Vanligtvis bör det därför startas från en batch-processor, för inte att binda så dyrbara resurser. Men lägg märke till att inga andra kan använda denna batchprocess när DDI-processen är igång! Informera de andra användarna av maskinen om att de för all del inte får lägga sig i batch-kö på den batchprocessorn som användare DATA-DICTIONARY är aktiv på, dvs. den processorn som DDI-processen är startad på. Då kommer de att få vänta 'oändligt' länge!

En batchprocessor får upplysningar, om vad som skall göras från en fil, som vi kallar processens in-fil. Dessutom måste den ha en ut-fil för att skriva på, t ex felmeddelanden. För att starta DDI-processen används

(DATA-DICT)DDI-MACHINE:BATC

som in-fil och

(DATA-DICT)DDI-MACHINE:SYMB

som ut-fil. Den sistnämnda bör vara tom när vi startar DDI-processen. Kontrollera med ÉLI-FI att den är upprättad!

Systemet levereras med följande in-fil DDI-MACHINE:BATC :

```
ÉENTER DATA-DICTIONARY,,,32000
É(DATA-DICTIONARY)DDI-MACHINE:PROG
128
129
CC <DDI PROCESS> CC
É
```

Första raden i denna fil talar om för batchprocessen, att DDI-processen skall använda databasbeskrivningen på användarområdet DATA-DICTIONARY och att den skall gå så länge som möjligt, dvs. 32000 minuter.

Vi måste kanske förändra denna rad. Generellt har den innehållet:

ÉENTER <SINTRAN-användare>,<lösenord>,<ev. projektnr.>, <maxtid för processen>

När accounting-systemet inte används på maskinen, utelämnas projektnr, dvs. det skall inte stå något mellan de två sista kommana. Se filen ovan, där man också har förutsatt, att det inte är lösenord för Sintran-användaren.

Om accountingsystem används på maskinen, kan det vara flera parametrar i ENTER kommandot än ovan nämnt. Ändra filen på motsvarande sätt, så att den passar ert system!

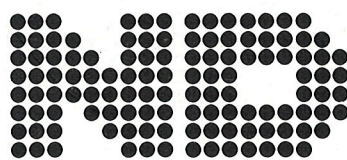
På rad 2 och 3 skall numren på de intern-device som DDI-processen använder stå. Dessa måste anpassas till de nummer, som är tillgängliga lokalt. Se beskrivningen av hur ACCESS laddas från FLOPPY.

B I L A G A

KONTROLLKOMMANDON I DBA-MODULEN

Bilaga
KONTROLLKOMMANDON I DBA-MODULEN

STRYK	alt. <D> <D>	Stryk-tangenten alt. två gånger <D> stryker det fält där markören står.
SKIFT STRYK	alt.<D> <L>	SKIFT STRYK-tangenten alt. <D> <L> stryker hela den linje där markören står.
HJÄLP	alt. <G>	Ger den hjälpinformation som är aktuell i den situation man befinner sig.
SKRIV	alt. <O>	Skriver det som står på skärmen ut på fil. (Filnamnet efterfrågas).
a	alt. <A>	Stryker det tecken där markören är.
EXP	alt. <E>	Ger möjlighet för insättning av tecken framför det tecken där markören står.



Norsk Data

STOCKHOLM:
(KANALVÄGEN 3),
BOX 721,
194 27 UPPLANDS VÄSBY
TEL: 0760-92000

GÖTEBORG:
(FRÖFÄSTEGATAN 22),
BOX 258,
421 23 VÄSTRA FRÖLUNDA
TEL: 031-4967 60

MALMÖ:
S. TULLGATAN 3, 5 tr.,
211 40 MALMÖ
TEL: 040-70510

TELEX: 15255 NORDATA S
POSTGIRO: 30121-8
BANKGIRO: 217-9752
BANK: SKANDINAVISKA ENSKILDA BANKEN