

Norsk Data



ACCESS DBA **Användarhandbok**

ND-30.020.3 SW

A decorative graphic consisting of a grid of yellow and black dots. The dots are arranged in a pattern that is roughly rectangular, with some dots missing or colored differently to create a textured effect. The dots are arranged in a grid that is approximately 15 columns wide and 10 rows high, with some dots missing or colored differently to create a textured effect.

ACCESS DBA

Användarhandbok

ND-30.020.3 SW

Kunngjøring

Opplysningene i dette dokumentet kan endres uten varsel. Norsk Data er ikke ansvarlig for feil som måtte forekomme i dette dokumentet. Norsk Data er ikke ansvarlig for sine bruker-systemers anvendelse eller pålitelighet på maskiner som ikke er levert eller anbefalt av Norsk Data.

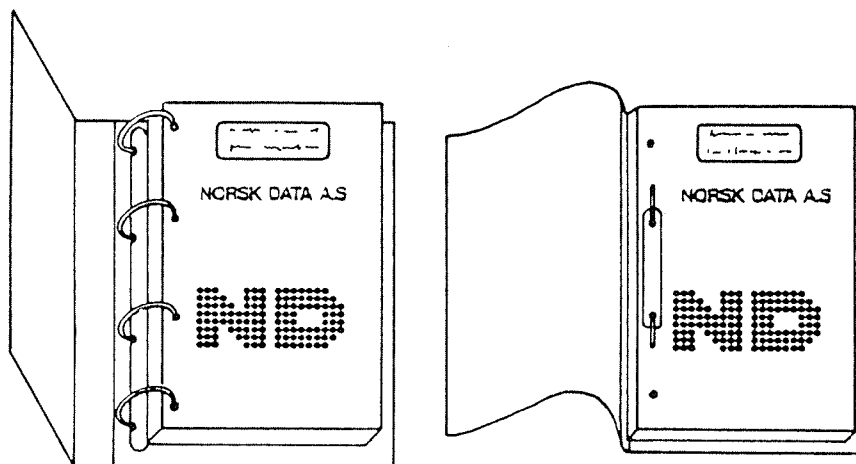
De opplysninger som er gitt i dette dokumentet er beskyttet av opphavsrett. De må ikke fotostat-kopieres, reproduseres eller oversettes uten at Norsk Data's tillatelse først er innhentet.

Copyright©1985 — Norsk Data A.S.

Denne håndboken er i løsbladsystem for å forenkle oppdatering. Gamle sider kan fjernes og nye sider settes inn på en enkel måte hvis håndboken er revidert.

Løsbladsystemet gjør det også mulig å plassere håndboken i en ringperm (A) for å beskytte den og for å gjøre det lett å slå opp i den. Ringpermer med 4 ringer tilsvarende hullene i håndboken kan bestilles i to bredder, 30 mm og 40 mm. Bruk bestillingsskjema nederst på siden.

Håndboken kan også plasseres i plastomslag (B). Dette omslaget passer bedre for håndbøker på 100 sider eller mindre enn for større håndbøker. Plastomslag kan også bestilles nederst på siden.



A: Ringperm

B: Plastomslag

Vennligst send bestillingen til det lokale ND kontoret eller (i Norge) til:

Norsk Data A.S
Grafisk Senter
Postboks 25, Bogerud
0621 Oslo 6

BESTILLING

Jeg ønsker å bestille:

..... Ringpermer, 30 mm, nkr 20,- pr. stk.

..... Ringpermer, 40 mm, nkr 25,- pr. stk.

..... Plastomslag, nkr 10,- pr. stk.

Navn

Firma

Adresse

By

Håndbøker kan oppdateres på to måter: nye utgaver og revisjoner. Nye utgaver består av en helt ny håndbok som erstatter den gamle håndboken. Nye utgaver inneholder alle revisjoner som er foretatt siden den forrige utgaven ble sendt ut. Revisjoner består av en eller flere enkeltsider som brukeren selv skal sette inn i håndboken. Alle reviderte sider oppgis på en ny trykkingsfortegnelse som sendes ut sammen med revisjonen. Denne nye trykkingsfortegnelsen skal erstatte den gamle.

Nye utgaver og revisjoner annonseres i Customer Support Information (CSI), og kan bestilles som forklart nedenfor.

Leserens kommentar-ark bakerst i denne håndboken kan brukes både til å informere om feil i håndboken, og til å gi en bedømmelse av denne. Både detaljerte og generelle kommentarer er velkommen.

Arkene, samt forespørsler om dokumentasjon sendes til:

Dokumentasjonsavdelingen
Norsk Data A.S
Boks 25, Bogerud
0621 Oslo 6

Bestillinger av dokumentasjon sendes til det lokale ND-kontoret eller (i Norge) til:

Grafisk Senter
Norsk Data A.S
Boks 25, Bogerud
0621 Oslo 6

Förord:

PRODUKTEN

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

LÄSAREN

Manualen är skriven för personer som skall vara ansvariga för systemfunktionerna i ACCESS fråge-system.

FÖRHANDSKUNSKAPER

Läsaren bör ha baskunskaper om ADB. Hon eller han skall ha kunskap om ND operativsystemet SINTRAN III, ISAM-filer och ACCESS-systemet. För att skapa SIBAS-tabeller, är det också bra att ha en grundläggande kunskap om strukturen på SIBAS-databas.

MANUALEN

Manualen beskriver DBA-programmet i ACCESS. Den innehåller information som behövs för att underhålla ACCESS-systemet. Det är mera en referens manual än en handbok som skall läsas från början till slut.

FÖRÄNDRING I PRODUKTEN OCH MANUALEN

- Namnen på lagringsformaten har ändrats och en del nya har tillkommit (NUMERIC TEXT, BCD, UNPACKED DECIMAL), se sida 20.
- SIBAS-realmbeskrivning kan kopieras automatiskt till ACCESS tabell-beskrivning. Group items, sets och CALC-nycklar kan definieras, sidan 34.
- Tabeller, databaser, användare och användarrättigheter till tabeller kan definieras från ACCESS (sidorna 37, 25, 18 och 29).
- D-versionen av ACCESS lagrar flera transaktioner på en fil (ett transaktionsbibliotek). Ett kapitel om fleranvändartransaktionsbibliotek har också lagts till (sidan 55).
- DDI-processen skall numera stoppas med en "mode-fil", ACC-DDI-STOP:MODE, inte med SINTRAN-kommando ABORT-JOB.

Se också förordet i ACCESS Användarhandbok.

ANDRA AKTUELLA HANDBÖCKER

| | |
|-------------------------------------|----------------|
| ACCESS Användarhandbok | ND-60.153.3 SW |
| SINTRAN III Reference Manual | ND-60.128.4 EN |
| SINTRAN III Timesharing/Batch Guide | ND-60.132.3 EN |
| ISAM Reference Manual | ND-60.108.5 EN |
| SIBAS II User Manual | ND-60.127.4 EN |

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 DBA-PROGRAMMET | 1 |
| 1.1 | Introduktion | 3 |
| 1.2 | En översikt över ACCESS och DBA | 3 |
| 2 | DBA I SINTRAN. | 7 |
| 2.1 | Start och stopp av DDI | 9 |
| 2.2 | Användare i systemet | 10 |
| 2.3 | Att ge flera användare tillgång till samma databas | 10 |
| 2.4 | DBA och användarområden i SINTRAN fil-systemet | 11 |
| 2.5 | Kommandon | 12 |
| 2.6 | Start av DBA | 12 |
| 2.7 | Organisation av den dagliga driften av programmet | 13 |
| 3 | STARTNIVÅN | 15 |
| 3.1 | Hur Du definierar ACCESS-användare | 17 |
| 3.1.1 | Från DBA | 17 |
| 3.1.2 | Från ACCESS | 18 |
| 3.2 | Hur Du definierar DBA-användare | 18 |
| 3.3 | Hur Du skapar dataelementen | 19 |
| 3.3.1 | Hur Du specificerar lagringsformatet | 19 |
| 3.3.2 | Motsvarighet med COBOL-data-beskrivning | 22 |
| 3.3.3 | Reserverade datatypsnamn | 22 |
| 3.4 | Hur Du tar reda på var en datatyp används | 23 |
| 3.5 | Odefinierade-datatypeer | 23 |
| 3.6 | Hur du definierar en ny databas | 23 |
| 3.6.1 | Från DBA | 23 |
| 3.6.2 | Från ACCESS | 25 |
| 3.7 | Hur Du kan få en lista på tillgängliga databaser | 25 |
| 3.8 | Hur Du stryker en databas | 25 |
| 3.9 | Ändringar i en databas-beskrivning | 26 |
| 3.10 | Avslutning av DBA startnivå | 26 |
| 4 | TABELLDEFINITIONSNIVÅN | 27 |
| 4.1 | Hur Du specificerar användares tillgång till databaser | 29 |
| 4.1.1 | Hur Du specificerar behörighet från ACCESS | 29 |
| 4.2 | Hur Du definierar dataelement i tabeller | 30 |
| 4.3 | Hur Du definierar och redigerar tabeller | 30 |
| 4.3.1 | Söknycklar i ISAM-tabeller | 31 |
| 4.3.2 | Kolumnbredd och resultatlängd | 32 |
| 4.3.3 | Användning av FLATA-filer | 32 |

| Sektion | Sida |
|-------------------|---|
| 4.4 | Tabeller och filer i ACCESS 33 |
| 4.5 | Hur Du definierar SIBAS-tabeller 34 |
| 4.6 | Hur Du stryker en tabell-beskrivning 36 |
| 4.7 | Hur Du får en lista på tillgängliga tabeller i databasen . 36 |
| 4.8 | Avslutning på tabelldefinitionsnivån 37 |
| 4.9 | Att skapa och redigera en tabell från ACCESS 37 |
| 4.9.1 | Skapa en tabell från början 37 |
| 4.9.2 | Hur Du ändrar en tabell-beskrivning 41 |
| 4.10 | Hur Du skapar olika synliga områden 42 |
| 5 | KOMMANDOT LADDA-DATABAS I ACCESS 43 |
| 6 | ALLMÄNT OM KÖRNING AV ACCESS-SYSTEMET 51 |
| 6.1 | Kommandofiler 53 |
| 6.1.1 | Kommando DUMP-HELP-FIL i ACCESS 54 |
| 6.1.2 | Standard transaktionsbibliotek och standardtransaktion . 54 |
| 6.2 | Fleranvändartransaktionsbibliotek 55 |
| 6.3 | Någonting är fel i arbetsområdet 55 |
| 6.4 | DDI inte aktiv 55 |
| 6.5 | Kommunikationsproblem mellan DDI och ACCESS 55 |
| 6.6 | Förbättringar i ACCESS söktid 56 |
| 6.7 | Andra felmeddelanden 57 |
| 7 | FULLSTÄNDIGA EXEMPEL 59 |
| 7.1 | Exempel på hur Du definierar användare 61 |
| 7.2 | Exempel på Hur Du definierar en databas 62 |
| <u>BILAGA</u> | |
| A | DDI-PROCESSEN SOM EN BATCH-PROCESS 67 |
| B | INFORMATION FÖR SYSTEMANSVARIG 71 |
| C | KONTROLLKOMMANDON I DBA MODULEN 77 |
| | Nyckelord 81 |

KAPITEL 1

ACCESS- OCH DBA-PROGRAMMET

1 ACCESS- OCH DBA-PROGRAMMET

1.1 INTRODUKTION

ACCESS är ett system för användare med liten eller ingen erfarenhet av datorer som vill kunna behandla stora mängder information. ACCESS hämtar själv en beskrivning av den information, som skall behandlas. Detta görs med hjälp av en process kallad Data Dictionary (DDI). Beskrivningarna lagras på ISAM-filer (DDI-filer), se sida 13. Dessa filer finns normalt under användare DATA-DICTIONARY, men kan också finnas under en annan användare, se kapitel 2.4. I följande fall, förutsätts att processen DDI körs under användare DATA-DICTIONARY, vilket betyder att filerna också måste finnas under denna användare.

DDI-processen är implementerad som en batch-process <1> Processen måste vara aktiv när ACCESS körs. Den ger ACCESS en fullständig beskrivning av data-strukturen systemet arbetar med.

Vi behöver också ett hjälpmedel för att skapa systemet med för användaren, dvs skapa nya beskrivningar och ändra gamla. Detta är DBA, Data Bas Administratören, som beskrivs i denna manual.

Bara en person som definierats som DBA-användare har tillgång till DBA-programmet.

DBA-programmet som beskrivs i denna handbok är skapat för att till ACCESS ge information som behövs under det dagliga arbetet. DBA-programmet används för att skapa databaser och användare, och för att ändra dem och stryka dem när de inte längre behövs. ACCESS är beroende av en korrekt beskrivning på datastrukturen och svarstiden är beroende av hur dina data är strukturerade.

1.2 EN ÖVERSIKT ÖVER ACCESS OCH DBA

En ACCESS-användare kan ha tillgång till en eller flera databaser. En databas är en samling tabeller som kan vara FLATA-filer, ISAM-filer eller SIBAS-realmer. En SIBAS-process måste köras om Du vill ha tillgång till en SIBAS-databas.

Beskrivningen av ACCESS-databasen finns i en samling filer som automatiskt skapas av DBA-programmet. Dessa filer tillhör användaren DATA-DICTIONARY, samma användare som DDI-processen körs under.

<1> .DDI systemet är kopplat till en batch-process i SINTRAN. Detta betyder att den inte upptar en terminal. Om en terminal kan reserveras för DDI-processen, behöver Du inte använda batch-process, se bilaga A.

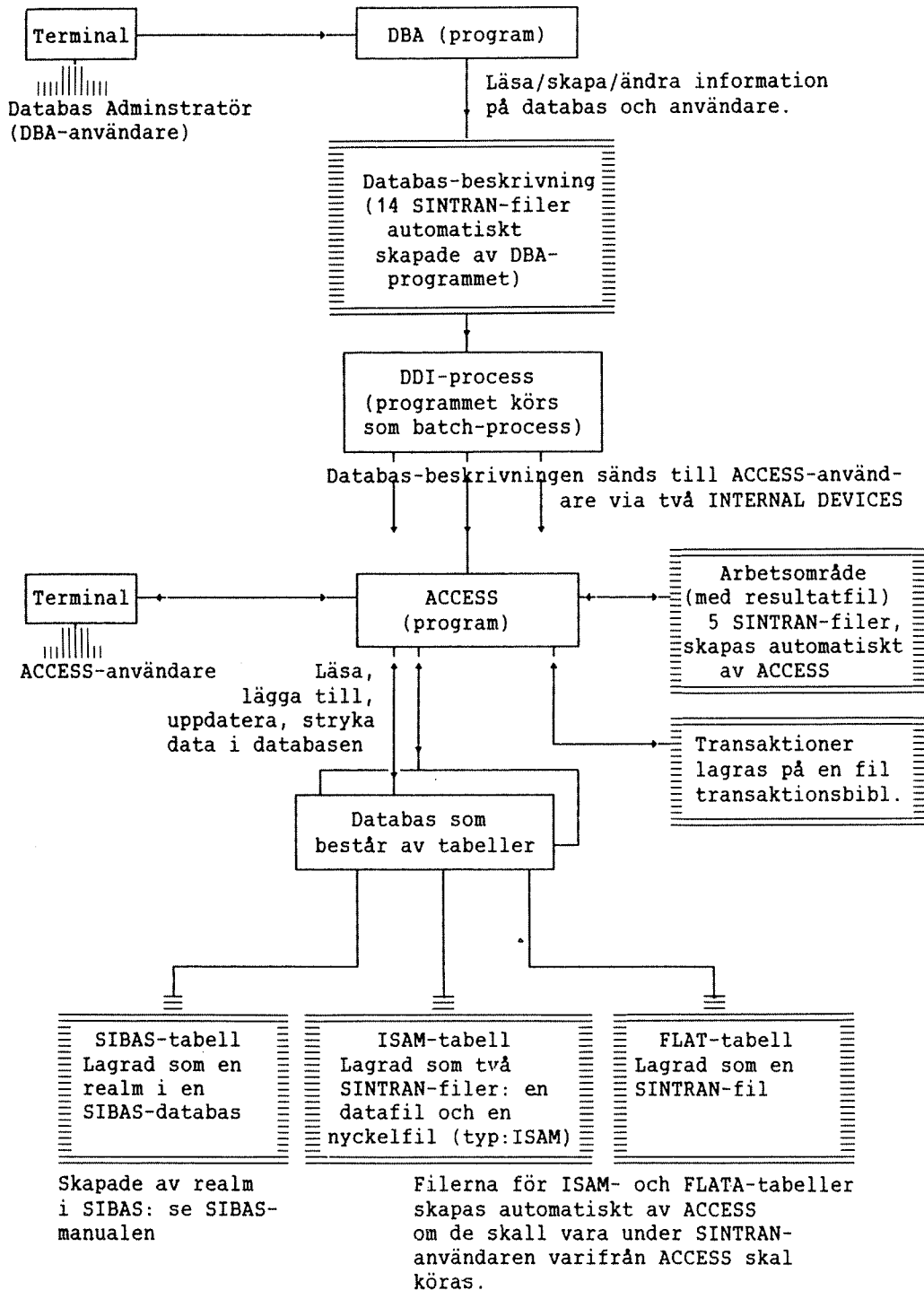


Fig. 1. En översikt över ACCESS och DBA

K A P I T E L 2

DBA I SINTRAN.

2 DBA I SINTRAN.

2.1 START OCH STOPP AV DDI

För att starta DBA-programmet, logga in som användare DATA-DICTIONARY, där beskrivningen av databasen finns.

Om Du loggar in som en annan användare, kommer Du inte att ha tillgång till databas-beskrivningen under användare DATA-DICTIONARY. Dessutom kommer informationen du skriver in under den andra användaren, att ignoreras av ACCESS senare. Mera information om detta i kapitel 2.4.

Se till att ingen använder ACCESS för tillfället genom att använda SINTRAN-kommandot TERM-STAT,,, (se efter förkortning av "ACCESS"). Om någon använder ACCESS, måste de logga ut, annars får de problem när Du fortsätter enligt beskrivningen nedan. Se till att DDI-processen är passiv genom att använda SINTRAN-kommandot:

```
@LIST-BATCH-PROCESS←↓
```

En lista innehållande status på varje batch-process visas. Om användare DATA-DICTIONARY använder en batch-process, måste denna avbrytas enligt nedan:

```
@MODE ACC-DDI-STOP:MODE←↓
```

Du är nu klar för att köra DBA-programmet. Ge kommandot:

```
@DBA←↓
```

Efter att Du är klar med ditt arbete med DBA-programmet, kan ACCESS köras igen. Före en ny uppstart av ACCESS, starta DDI-processen med SINTRAN-kommandot:

```
@APPEND-BATCH,<batchno>,(DA-DIC)ACCESS-DDI:BATC,(DA-DIC)ACC-DDI:LOG
```

De två sista parametrarna är in- och ut-filerna för batch-processen.

Du kan även få DDI-processen att automatiskt startas när Du går ut ur DBA-programmet. Du gör detta genom att ersätta rad 3 i filen (DATA-DIC)DBA-HELP:TEXT med kommandot beskrivet ovan, men utan "@". (Du kan använda NOTIS-WP, men se till att Du inte lägger till eller stryker några rader i filen, Du skall bara skriva över rad 3.)

OBS: "APP-..." måste placeras längst till vänster, och kommandot måste rymmas på de 59 första positionerna på raden.

Transaktionsdelen av ACCESS är beroende av DDI-processen för att kunna fungera, så DDI måste vara aktiv medan Du använder ACCESS.

Ytterligare information om batch-systemet finns i SINTRAN III Timesharing/Batch Guide. Se också bilaga A.

Alla kommandon i SINTRAN kan förkortas så länge som de förblir entydiga. Det vill säga att kommandona på föregående sida kan också skrivas:

@LI-B-P,,,↵

@MODE -DDI:MODE↵

@AP-B <batch no>,-DDI:BATC,-DDI:LOG↵

Filen ACCESS-DDI:BATC är en fil som beskriver körningen av DDI som en batch-process. ACCESS-DDI:LOG är ut-filen för batch-processen.

Se till att filerna finns under användaren DATA-DICTIONARY med kommandot:

@LIST-FILES↵

2.2 ANVÄNDARE I SYSTEMET

ACCESS har sin egen användaridentifikation som inte är identisk med SINTRANS.

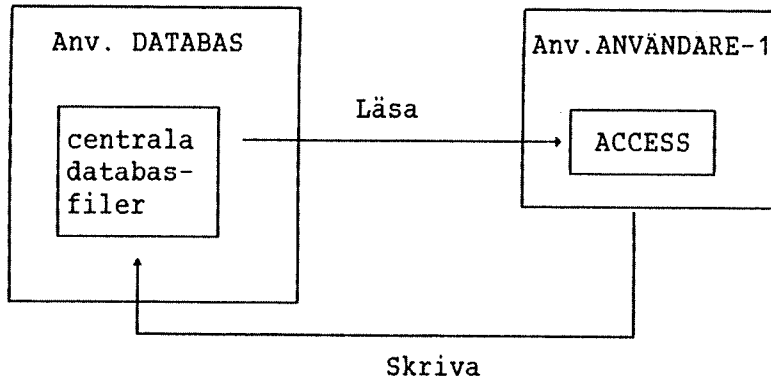
ACCESS-användare skapas i DBA-programmet och får där rättigheter till databaser och filer. Varje ACCESS-användare har ett arbetsområde, som består av fem SINTRAN-filer. Detta arbetsområde måste definieras för varje användare.

Endast en ACCESS-användare kan använda samma arbetsområde (filer) samtidigt, men flera personer (terminaler) kan vara inloggade i SINTRAN under samma SINTRAN-användarnamn. Förhållandet mellan SINTRAN-användare och arbetsområden förklaras på sidan 18.

2.3 ATT GE FLERA ANVÄNDARE TILLGÅNG TILL SAMMA DATABAS

Alla användare som skall lägga till, ändra eller stryka information i en databas (ISAM- eller FLAT-fil) måste ha skrivrättighet till filerna där informationen skall lagras. Eftersom SINTRAN-användare normalt inte har rätt att skriva på andra användares filer, kan detta bli problem när användare-1 som kör ACCESS i sitt eget arbetsområde, försöker att skriva på en databas-fil som tillhör användaren där den centrala databasen är placerad. Låt oss säga att användaren som äger databasen är en användare som skapats speciellt för detta ändamål och som kallas DATABAS.

Situationen kan beskrivas enligt nedan:



Detta problem kan kringgås på olika sätt:

1. Alla användare av databasen kan definieras som "friends" till användaren som äger databasen. Detta är dock begränsat till åtta användare, eftersom detta är max antalet av "friends" som tillåts.
2. Andras rättigheter (public access) för databas-filerna kan sättas till RWA. För att göra det enklare att skapa nya tabeller, kan "default public access" för användaren som äger databasen också sättas till RWA.

2.4 DBA OCH ANVÄNDAROMRÅDEN I SINTRAN FIL-SYSTEMET

Om Du kör DBA från en annan användare än DATA-DICTIONARY, kommer Du att få en oberonde databas-beskrivning. Observera att ACCESS inte kommer att använda denna beskrivning om inte DDI-processen körs under denna speciella användare. Om Du vill använda en oberoende beskrivning, måste Du göra följande före Du startar DDI-processen: ändra innehållet i filerna (DATA-DIC)ACCESS-DDI:BATC och (DATA-DIC)ACC-DDI-STOP:MODE, så att namnet DATA-DICTIONARY på första raden ersätts med namnet på den SINTRAN-användaren Du är inloggad som när Du kör DBA-programmet.

Om Du vill att din installation skall ha en databas-beskrivning under ett annat SINTRAN-användareområde som tillägg till den "normala" beskrivningen under DATA-DICTIONARY, måste Du starta två DDI-processer som två olika batch-processorer. De två processerna måste använda olika "internal devices", och ha sina egna kopior på in- och ut-filer i APPEND-BATCH-kommandot som beskrivs i sektion 2.1 (Jämför beskrivning på hur ACCESS laddas i SINTRAN från floppy-disk). Varje kopia på in-filen måste ha sitt eget SINTRAN-användarnamn på första raden.

Eftersom filen ACCESS-HELP:TEXT innehåller namnet på "internal devices", är det den som bestämmer vilken av de två DDI-processerna som ACCESS kommer att kommunicera med. När Du startar ACCESS-programmet från ett SINTRAN-användareområde, kommer ACCESS först att söka i detta område efter filen ACCESS-HELP:TEXT. Om den inte finns här eller under användare SYSTEM, kommer ACCESS-HELP under användare DATA-DICTIONARY att användas.

I filen som påträffas, måste de två "internal devices" till den önskade DDI-processen ha specificerats.

Detta betyder: för varje SINTRAN-användare som skall använda sin egen individuella databas-beskrivning som inte ligger under användare DATA-DICTIONARY, så måste Du kopiera filen (DA-DICT)ACCESS-HELP:TEXT till denna SINTRAN-användare, och ändra numret på den "internal device" på rad 2 och 3 så att de stämmer överens med den "interna device" i in-filen till den önskade DDI-processen.

En oberoende databas-beskrivning tillåter Dig att utveckla databasen utan att hindra den dagliga driften av ACCESS. Det ger också nybörjarna ett tillfälle att testa och experimentera utan att hindra den "egentliga" körningen av programmet under användare DATA-DICTIONARY.

2.5 KOMMANDON

Kommandon ges i kommando-läge, markerat med * eller ** under strecken nederst på skärmen.

DBA-programmet har två nivåer. Nivå 1 kallas Startnivå. Alla tabeller definieras på nivå 2, Tabelldefinitionsnivå. Antalet asterisker visar vilken nivå Du befinner Dig på:

- * → Startnivå
- ** → Tabelldefinitionsnivå

Alla kommandon i DBA kan förkortas på samma sätt som i SINTRAN. Detta betyder att **R-TA är tillräckligt för att skilja kommandot **REDIGERA-TABELL från kommandot **REDIGERA-DATATYPER. Blanka före ett kommandonamn är inte tillåtet.

Ett kommando avbryts genom att trycka på "HEMMA"-tangenten:



2.6 START AV DBA

DBA-programmet startas med kommandot:

@DBA

Kom ihåg att Du måste vara inloggad som användare DATA-DICTIONARY om Du vill redigera beskrivningarna som DDI-processen kommer att ge ACCESS.

DBA frågar efter användar-identifikation. Användar-identifikationen är inte SINTRAN-användar-identifikationen, som här är DATA-DICTIONARY, utan ACCESS' egen användar-identifikation. Alla ACCESS-användare som skall använda DBA-programmet, måste ha definierats som DBA-användare.

Om Du vill skapa en obereonde databas som beskrivits i kapitel 2.4, kommer DBA inte att fråga efter användar-identifikationen den första gången. Du kommer direkt in i användarbeskrivningen, där bara användare SYSTEM är definierad. Ändra SYSTEM till det önskade användarnamnet och ge arbetsområde och eventuellt lösenord (se side 18.) Det nya användarnamnet kommer automatiskt att registreras som DBA-användare.

2.7 ORGANISATION AV DEN DAGLIGA DRIFTEN AV PROGRAMMET

Om Du vill arbeta med att utveckla en databas och samtidigt undvika att störa den dagliga driften av ACCESS, bör arbetet organiseras på följande sätt:

Kopiera databasbeskrivningen (DDI-filerna) från användare DATA-DICTIONARY till användare ACCESS. Användare DATA-DICTIONARY måste vara "friend" till användare ACCESS. Det finns följande DDI- filer:

```
DATABASE (:DATA och :ISAM)
TABLE      (:DATA och :ISAM)
ATTRIBUTE (:DATA och :ISAM)
GROUP-SET (:DATA och :ISAM)
TYPE       (:DATA och :ISAM)
USER       (:DATA och :ISAM)
ACCESS     (:DATA och :ISAM)
```

Starta DBA-programmet när Du är inloggad som SINTRAN-användare ACCESS, och definiera den nya databasen.

Om det finns tabeller av typen ISAM eller FLAT, kopiera filerna till användarområde ACCESS. Använd samma filnamn! Om Du ger kommandot **ISAM-FILER eller **FLATA-FILER i DBA-programmet, kommer filnamnen vid detta tillfälle att visas utan SINTRAN-användarnamn.

Starta en individuell "test-DDI-process" för databas-beskrivningen på användare ACCESS, se kapitel 2.4.

Nu kan ACCESS köras mot kopian av databasen för att testa att allting fungerar korrekt, eller den kan användas för utbildning av nya ACCESS-operatörer. Användarna skall logga in som ACCESS. Observera att det måste finnas en kopia av filen ACCESS-HELP:TEXT under SINTRAN-användare ACCESS, med det korrekta "internal device" numret för "test-DDI-processen".

Efter att databas-beskrivningen har testats och Du konstaterat att den är ok, måste DDI-processen stoppas. Sedan måste DBA-programmet köras än en gång under SINTRAN-användare ACCESS för att sätta in SINTRAN-användarnamnen i de verkliga filnamnen FLAT- och ISAM-tabeller (kommandon **ISAM-FILER och **FLATA-FILER). Sedan måste databas-beskrivningen kopieras till DATA-DICTIONARY, och det riktiga arbetet med ACCESS mot denna databas kan börja. En "backup" (säkerhetskopia) på databas-beskrivningen på användaren ACCESS bör tas vid samma tidpunkt.

K A P I T E L 3

STARTNIVÅN

3 STARTNIVÅN

3.1 HUR DU DEFINIERAR ACCESS-ANVÄNDARE

3.1.1 FRÅN DBA

Ge kommandot:

*ACCESS-ANVÄNDARE←

Följande bild kommer att visas på skärmen:

| ACCESS Systemanvändare | | |
|------------------------|----------|--------------|
| ACCESS användarnamn | Lösenord | Arbetsområde |
| | | |

ACCESS-användarnamn

ACCESS-användarnamnet visas i den första kolumnen, och nya användare skrivs in i slutet på listan. Om det inte finns några användare i ACCESS, kommer kolumnen att vara tom och den första användaren kan skrivas in på första raden. Användare kan strykas med tangenterna SKIFT + STRYK på en NOTIS-terminal (eller CTRL+D CTRL+L).

Lösenord

För att få ett säkert ACCESS-system, bör varje användare ha ett hemligt lösenord, vilket visas i denna kolumn. DBA-användare har tillåtelse att se, ändra, stryka och skapa nya lösenord. En DBA-användare är inte tillåten att se eller ändra en annan DBA-användares lösenord. Dessa lösenord visas överskrivna med asterisker.

Arbetsområde

Varje användare måste ha fem filer som ACCESS kan använda som temporärt arbetsområde.

ACCESS skapar dessa filer automatiskt, men deras namn måste specificeras i kolumnen "arbetsområde". Om arbetsområdet för ANVÄNDARE-1 är specificerad som (ANVÄNDARE-1)WORKAREA", och ANVÄNDARE-1 kör ACCESS när han/hon är inloggad som SINTRAN-användare ANVÄNDARE-1, kommer ACCESS att skapa följande fem filer:

```
(ANVÄNDARE-1)WORKAREA:DATA
(ANVÄNDARE-1)WORKAREA:SORT
(ANVÄNDARE-1)WORKAREA:SYS1
(ANVÄNDARE-1)WORKAREA:SYS2
(ANVÄNDARE-1)WORKAREA:SYS3
```

Om ANVÄNDARE-1 försöker köra ACCESS när han/hon är inloggad som en annan SINTRAN-användare, kommer ACCESS inte att lyckas skapa arbetsområdes filerna, och kommer att visa felmeddelandet "Någonting är fel i arbetsområdet': "(ANVÄNDARE-1)WORKAREA:DATA"/75'.

3.1.2 FRÅN ACCESS

Du kan skapa en ny användare från ACCESS med kommandot *SKAPA-ANVÄNDARE (på Startnivån).

Anropssekvensen är:

```
*SKAPA-ANVÄNDARE <användarnamn> <ACCESS arbetsområde>+↵
```

Om användaren redan finns, kan hans/hennes arbetsområde ändras.

3.2 HUR DU DEFINIERAR DBA-ANVÄNDARE

DBA-användare är speciella användare i ACCESS-systemet, som tillåts skapa nya databaser, stryka gamla data och ändra befintliga beskrivningar på filer som tillhör hans/hennes databas.

För att definiera en ACCESS-användare som en DBA-användare, ge kommandot:

```
*DBA-ANVÄNDARE+↵
```

Följande bild visas på skärmen:

| |
|-----------------------|
| Databas-administratör |
| DBA-användare |
| |

Kolumnen innehåller de ACCESS-användares namn som skall bli DBA-användare. Tryck på HJÄLP-tangenten för att få en lista på alla definierade användare.

Observera att Du inte kan redigera/stryka information om andra DBA-användare, enbart ditt eget.

3.3 HUR DU SKAPAR DATAELEMENTEN

För att den vanliga ACCESS-användaren inte skall behöva bekymra sig om detaljer vid lagring och läsning, skapas dessa en gång för alla i DBA-programmet.

Alla dataelement har en typ. Denna beskriver hur dataelementen skall lagras, men den begränsar också användandet av elementen. Dataelement av samma typ kan representera varandra.

Det finns ett speciellt kommando för att definiera typen på ett dataelement. Ge kommandot:

*REDIGERA-DATATYPER←↓

Följande bild visas på skärmen:

| Datatyper | | |
|--------------|-----------------|----------------|
| Datatypsnamn | Utskriftsformat | Lagringsformat |
| | | |

Varje datatyp måste få ett namn som senare kommer att refereras till i tabell-beskrivningen.

Det är tillräckligt att fylla i fälten "Datatypsnamn" och "Lagringsformat". Utskriftsformat, som visar hur data kommer att skrivas ut på skärmen i frågeresultatet kommer att fyllas i automatiskt.

3.3.1 HUR DU SPECIFICERAR LAGRINGSFORMATET

Lagringsformat, eller "Internt format", definierar på vilket sätt data skall lagras. Det begränsar också användningssättet.

Följande tabell visar alla tillåtna lagringsformat.

| Lagringsformat | Möjlig aritmetik | Storlek på dataelementet |
|---|--|---|
| TEXT (n) | NEJ | n bytes (max 2048) |
| NUMERIC TEXT (n) | JA - Precision 18 siffror | n bytes |
| INTEGER2 | JA (Min/max värde: -32 768 to 32 767) | 2 bytes/16 bits |
| INTEGER4 | JA (Min/max värde: -2 147 483 648 till +2 147 483 647) | 4 bytes/32 bits |
| REAL4 * | JA - Precision 6 siffror | 4 bytes/32 bits |
| REAL6 * | JA - Precision 8 siffror | 6 bytes/48 bits |
| REAL8 | JA - Precision 16 siffror | 8 bytes/64 bits |
| BCD (n,m) eller PACKED DECIMAL (n,m) | JA - Precision 18 siffror | (n + m + 2)/2 bytes |
| { SIGNED } † { UNSIGNED } | | |
| UNPACKED DECIMAL (n,m) | JA - Precision upp till 18 siffror | n + m, utom SIGNED SEPARATE som ger n + m + 1 |
| { SIGNED } † { UNSIGNED } | | |
| { EMBEDDED } † { SEPARATE } | | |
| { TRAILING } † { LEADING } | | |

* Valet mellan REAL4 och REAL6 beror på den typ av ND-100 CPU som används. Om Du skriver bara REAL, kommer DBA-programmet att välja korrekt format.

† Standard är SIGNED, EMBEDDED, TRAILING

Relativt oerfarna DBA-användare som vill definiera sina egna tabeller behöver vanligtvis bara lagringstyperna TEXT och NUMERISK TEXT. Se beskrivning på ACCESS-kommandot *SKAPA-TABELL (sidan 37) för mer detaljerad beskrivning av dessa typer (i ACCESS kallade CHARACTER och NUMERICK).

Om datatypen skall användas i SIBAS-item-definition, måste storleken på dataelementet vara identiskt med längden (LENGTH), som denna är definierad i SIBAS-DRL. Observera att LENGTH i SIBAS-DRL är specificerad som antal 2-bytes ord. SIBAS-typ som valts i DRL har ingen betydelse.

Exempel: vi har en 48-bit floating point ND-100 CPU, och "no1" och "no2" är definierade här med lagringsformat REAL. Dessa kan vara definierade i SIB-DRL som:

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

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

Observera att lagringsformatet i ACCESS måste vara identiskt med det som är definierat för databasen (och som andra användare använder när data lagras). ACCESS sätter inte paritetsbitar i typerna TEXT och CHARACTER.

Om Du bara skriver REAL, kommer DBA att välja mellan REAL4 och REAL6, beroende på CPU.

Om Du vill använda samma databas på både en ND-100 och en ND-500, skall REAL8 användas.

3.3.2 MOTSVARIGHET MED COBOL-DATA-BESKRIVNING

Följande tabell visar lagringsformatet som används i DBA-programmet och motsvarande COBOL-data-beskrivning. Den är upprättad för att förenkla definieringen av tabeller mot data som genererats av ett COBOL-program.

| Lagringsformat | COBOL 100 | COBOL 500 |
|---|---------------------------------|--|
| TEXT (n) | PIC X(n) | PIC X(n) |
| NUMERIC TEXT (n) | | |
| INTEGER2 | COMP-1 | |
| INTEGER4 | COMP-1 PIC 9(n) där n >= 5 | COMP-1 |
| REAL4 | COMP-2 på en 32-bit CPU | COMP-2 PIC S9(n)V9(m) där n + m <= 6 |
| REAL6 | COMP-2 på en 48-bit CPU | COMP-2 PIC S9(n)V9(m) där n + m >= 7 |
| REAL8 | | |
| BCD (n,m) eller PACKED DECIMAL (n,m) | COMP-3 PIC 9(n)V9(m) | COMP-3 PIC 9(n)V9(m) |
| { SIGNED } { UNSIGNED } | PIC S9(n)V9(m) PIC 9(n)V9(m) | PIC S9(n)V9(m) PIC 9(n)V9(m) |
| UNPACKED DECIMAL (n,m) | | |
| { SIGNED } { UNSIGNED } | PIC S9(n)V9(m) PIC 9(n)V9(m) | PIC S9(n)V9(m) PIC 9(n)V9(m) |

3.3.3 RESERVERADE DATATYPSNAMN

I Data Dictionary är vissa datatypnamn reserverade för ACCESS' användning. De skapas om en tabell definieras från ACCESS, eller om kommandot **KOPIERA-REALMS används i DBA-programmet.

Reserverade datatypsnamn är:

INTEGER-<n>
REAL-<n>
BCD-<n>-<m>
NUMERIC-<n>
CHARACTER-<n>
FILLER-<n>

3.4 HUR DU TAR REDA PÅ VAR EN DATATYP ANVÄNDS

För att få information om användandet av datatyper som är definierade, ge kommandot:

*VAR-ANVÄNDS-DATATYP <typnamn>↵

Det är ofta bra att veta om en datatyp används överhuvudtaget; om inte, då kan datatyp-beskrivningen strykas. Det kan också vara bra att veta var en datatyp används om Du vill definiera en ny typ, och använda den i en definierad tabell där den nya datatypen har samma lagringsformat som den gamla. Det kan också vara användbart att jämföra datatyper som har nästan lika lagringsformat, för att se om någon av dem är överflödig.

Du kan inte stryka en datatyp som används.

3.5 ODEFINIERADE-DATATYPER

För att få en lista på alla datatyper som har använts i "*REDIGERA-TABELL" kommandot, men ännu inte definierats med kommandot "*REDIGERA-DATATYPER", använd kommandot:

*ODEFINIERADE-DATATYPER↵

Om Du, till exempel, har stavat fel på en datatyp i en tabell ("APPLES"), kommer detta att visas. Du bör inte använda en tabell som innehåller odefinierade datatyper, eftersom resultatet kommer att bli felaktigt.

3.6 HUR DU DEFINIERAR EN NY DATABAS

3.6.1 FRÅN DBA

En databas är ett samlingsbegrepp för en grupp tabeller som kan vara realmer i en SIBAS-databas, indexerade ISAM-filer eller så kallade FLATA-filer utan söknycklar. Posterna i filerna måste ha en fast längd. Varje dataelement måste ha samma längd och placering i varje post i tabellen.

Vid användandet av DBA-programmet kan varje användare ha tillgång till en sådan databas. Denna beskrivning måste innehålla information om hur dina data är lagrade, vem som har tillgång till databasen och hur informationen i databasen presenteras för användaren.

En ny ACCESS-databas kan skapas genom att ge följande kommando (om databasen skall innehålla SIBAS-tabeller, är det enklare att använda kommandot *VISA-DATABASER (detta är beskrivet nedan):

***SKAPA-DATABAS+↵**

DBA frågar efter namnet på den nya databasen:

Databas namn:SVENSK-DATABAS

Alternativt, kan Du ge databasnamnet tillsammans med kommandot:

***SKAPA-DATABAS SVENSK-DATABAS**

Det ACCESS-användarnamn Du gav när Du började arbeta med DBA, kommer att registreras som ägare till denna databas. Detta namn kan sedan förändras med kommandot *VISA-DATABASER (beskrivs nedan).

Denna databas kan ha tabeller av olika slag: SIBAS, ISAM och FLAT.

Informationen/data Du specificerade i DBA-programmet under användare DATA-DICTIONARY lagras i filer i detta användarområde, och flyttas inte till SIBAS-databasen eller till filsystemet.

Kommandot *SKAPA-DATABAS gör att systemet går från startnivå till tabelldefinitionsnivå.

Om databasen innehåller SIBAS-tabeller, är det bättre att skapa den med kommandot *VISA-DATABASER. En tabell enligt nedan visas då på skärmen:

| Skapa databaser | | | | |
|-----------------|------------|----------|----------------|-----------|
| Databasnamn | SIBAS namn | SIBAS nr | SIBAS lösenord | Skapad av |
| SVENSK-DATABAS | | 0 | | ACCESS |

Nya ACCESS-databaser kan skapas genom att de skrivs till i denna tabell.

Databasnamn är namnet på ACCESS-databasen.

SIBAS namn är namnet på SIBAS-databasen som kan användas i denna ACCESS-databas. Realmer från denna SIBAS-databas kan definieras som tabeller i ACCESS-databasen. Bara en SIBAS-databas kan användas i varje ACCESS-databas.

SIBAS_nr är numret på den SIBAS-process som kontrollerar SIBAS-databasen.

SIBAS lösenord är lösenordet på databasnivå i SIBAS-databasen.

Skapad av är den DBA-användare som skapat ACCESS-databasen. Med andra ord, skriv ditt DBA-användarnamn.

3.6.2 FRÅN ACCESS

Kommandot *SKAPA-DATABAS (beskrivet i föregående sektion) kan också ges på startnivån i ACCESS. Den används på exakt samma sätt som i DBA.

3.7 HUR DU KAN FÅ EN LISTA PÅ TILLGÄNGLIGA DATABASER

En lista på tillgängliga databaser i systemet kan fås med kommandot:

```
*VISA-DATABASER↵
```

Namnet på alla databaser visas på skärmen, tillsammans med viss information som bara är relevant när SIBAS-tabeller används. Denna information beskrivs ovan (kapitel 3.6.1).

3.8 HUR DU STRYKER EN DATABAS

En databas-beskrivning stryks från ACCESS-systemet genom att ge kommandot:

```
*STRYK-DATABAS↵
```

följt av namnet på databasen. Om namnet inte ges kommer systemet att fråga efter namnet på samma sätt som i kommandot *SKAPA-DATABAS.

För att detta kommando skall fungera kräver systemet att den DBA som ger kommandot också är ansvarig för databasen, alltså "ägare" till databasen.

De data-filer som är definierade som tillhörande denna databas berörs inte av detta kommando, men om de skall användas igen, måste de omdefinieras i DBA-programmet.

3.9 ÄNDRINGAR I EN DATABAS-BESKRIVNING

Oftare vill man ändra i en databas-beskrivning än man vill skapa en ny beskrivning. Detta kan göras med kommandot:

*REDIGERA-DATABAS←↓

Namnet på databasen ges direkt efter kommandot, eller som svar på frågan som ställs av systemet.

Beskrivningen på databasen läses då in och system går ned till tabelldefinitionsnivån (nivå 2).

Bara ägaren till databasen kan redigera databas-beskrivningen.

3.10 AVSLUTNING AV DBA STARTNIVÅ

För att avsluta DBA-programmet från startnivån, tryck på



eller ge kommandot:

*SLUT←↓

K A P I T E L 4

TABELLDEFINITIONSNIVÅN

4 TABELLDEFINITIONSIVÅN

Tabelldefinitionsivån (nivå 2) är till för att skapa, redigera och stryka tabeller eller beskrivningar på filer. Du går till denna nivå genom kommandot

*SKAPA-DATABAS←↓

eller

*REDIGERA-DATABAS←↓

Arbete på denna nivå kräver att både användaren och databasen är definierade.

4.1 HUR DU SPECIFICERAR ANVÄNDARES TILLGÅNG TILL DATABASER

Innan en användare kan ställa frågor till en databas, måste han/hon ges tillgång till databasen. Bara DBA-användaren som skapat databasen (med kommandot *SKAPA-DATABAS eller *VISA-DATABASER) kan ge andra användare tillgång. Detta görs med kommandot:

**REDIGERA-ANVÄNDARE←↓

Databasen har redan definierats på startnivå, så systemet kräver inte databasnamnet här. Följande bild visas på skärmen:

| Användarbehörighet | | | | |
|--------------------|--------|-----|------------|--------|
| Användarnamn | Skriv. | Ny. | Uppdatera. | Stryk. |
| | | | | |

I kolumnen användarnamn skriver Du in namnet på ACCESS-användaren som Du vill ge tillgång till databasen. (Tryck på HJÄLP-tangenten för att få en lista på definierade användare.) I kolumnerna under varje ACCESS-kommando skriver Du Ja eller Nej beroende på om Du vill att denna användare skall ha rätt att använda detta kommando. Observera att blank är identiskt med Nej, men ägaren av databasen har obegränsade rättigheter.

4.1.1 HUR DU SPECIFICERAR BEHÖRIGHET FRÅN ACCESS

Kommandot *REDIGERA-ANVÄNDARE på startnivå i ACCESS fungerar på samma sätt som DBA-kommandot som beskrivits ovan, men det tillåter dig inte att se de tidigare definierade rättigheterna.

Kommandot ges enligt:

****REDIGERA-ANVÄNDARE <datbasnamn><användarnamn><S.><N.><U.><S.><↓**

Om användaren redan har tillgång till databasen, kommer användarbehörigheten att ändras.

Du ger behörighet genom att svara JA eller NEJ, och är:

<S.> - Utskrift tillåtet.
<N.> - Insättning av nya poster är tillåtet.
<U.> - Uppdatering av poster är tillåtet.
<S.> - Strykning av poster är tillåtet.

4.2 HUR DU DEFINIERAR DATAELEMENT I TABELLER

Typen för alla dataelement måste definieras med kommandot

****REDIGERA-DATATYPER<↓**

Detta kommando beskrivs i detalj i kapitel 3.3. Det kan användas antingen före eller efter att tabellen har definierats.

4.3 HUR DU DEFINIERAR OCH REDIGERAR TABELLER

En ny tabell skapas med kommandot:

****SKAPA-TABELL<↓**

DBA-programmet frågar efter tabellnamn, och sedan efter tabelltyp. Följande tabelltyper finns:

ISAM: söknycklar kan specificeras
FLAT: inga söknycklar tillåtna
SIBAS: en SIBAS-DBMS-realm

För att redigera en tabell som redan finns, använd kommandot:

****REDIGERA-TABELL<↓**

I detta fall kommer DBA att enbart fråga efter tabellnamn.

Med båda dessa kommandon, visas följande bild på skärmen med tomma kolumner:

| | | | |
|---------------------|---------|--------|-----|
| <Namn på tabellen> | | | |
| Namn på dataelement | Datatyp | Nummer | Sök |
| | | | |

Du kan nu fylla i informationen och beskriva de dataelement som är använda i tabellen. Tabellen kommer senare att identifieras genom tabellnamnet.

Namnet som ges till varje dataelement i din tabell visas som kolumnnamn när Du senare använder ACCESS för att definiera och utföra frågor mot tabellen.

Fältet "Datatyp" fylls ut med namnen på de datatyper Du har definierat med kommandot **REDIGERA-DATATYPER.

Fältet "Nummer" fylls i med ett tal, för att indikera i vilken ordning som dataelementen skall visas senare i arbetstabellen och resultattabellen. Detta fält kan lämnas blankt; det kommer att fyllas i automatiskt av systemet.

Tabellen ovanför är den vänstra delen av tabellbilden som används för att skapa och redigera tabeller. Efter hand som Du fyller i fälten, flyttar fönstret sig mot höger och visar resten av bilden:

| | | | |
|--------------------|-----------|-------------|----------|
| <Namn på tabellen> | | | |
| Nummer | Söknyckel | Kolumnbredd | Radlängd |
| | | | |

4.3.1 SÖKNYCKLAR I ISAM-TABELLER

Söknycklar i ISAM-tabeller kan vara endera PRIMÄR eller SEKUNDÄRA. För FLATA-filer används inte detta fält.

När skall Du använda söknycklar?

Att använda söknycklar betyder snabbare sökningar. Frågor som använder selekterings- och sorterings-kriterier i en kolumn, kommer att utföras snabbare om denna kolumn är skapad som en söknyckel.

A andra sidan tar nycklar plats. När en kolumn definierats som en nyckel, kommer hela kolumnen att lagras två gånger: en gång i :DATA-filen och en gång i :ISAM-filen. Detta betyder att frågor som använder UPPDATERA., STRYK., eller LÄGGA TILL. kommer att utföras långsammare, eftersom nyckel-fältet måste lagras två gånger.

Som regel, använd som nycklar bara kolumner som Du tror att Du kommer att söka ofta i. Använd inte långa text-fält som nycklar.

Sekundär eller primär?

Primära nycklar ger en snabbare sökning än sekundära nycklar. Å andra sidan, en primärnyckelkolumn kan inte innehålla duplikat, inte heller värdet noll i ett numeriskt fält. Till exempel, om en kolumn skall innehålla efternamn, så skulle det inte vara klokt att definiera det som en primärnyckel, eftersom det kan finnas två eller flera personer med samma namn. Använd PRIMÄR bara när Du är säker på att kolumnen inte innehåller duplikat.

Det räcker med att ange "S" eller "P" i nyckelfält. Alla andra värden, inklusive felstavningar, indikerar att dataelementen inte skall knytas till en söknyckel. Det blir då mer tidskrävande att söka efter detta element, eftersom sökningen måste ske sekvensiellt.

En ISAM-tabell måste ha minst ett dataelement, maximum 64, definierat som söknyckel.

4.3.2 KOLUMNBREDD OCH RESULTATLÄNGD

Kolumnbredden används för att uppge den initiella startbredden för en kolumn i en ACCESS-tabell. Kolumnen får den angivna bredden eller den bredd som behövs, för att elementets namn skall få plats.

Resultatlängden är bredden på kolumnen när resultatet på en fråga visas på ett standardtabellformat. Till exempel:

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

Om inget är definierat, kommer ACCESS att se till att resultatet visas på ett förnuftigt sätt på skärmen.

4.3.3 ANVÄNDNING AV FLATA-FILER

En FLAT-fil är en sekvensiell fil med en fast postlängd. Alla poster i filen förutsätts ha en samma struktur. Det finns inga söknycklar i en sådan fil. Detta betyder att varje gång ACCESS skall hitta något i filen, måste ACCESS söka genom hela filen från början. Sådana filer används trots detta mycket, för de är enkla och pålitliga. En FLAT-tabell skapas som tidigare beskrivits, med skillnaden att inga söknycklar har definierats i söknyckel-kolumnen. Tabellen för definition av innehållet, är identisk med den som används för andra tabelltyper.

Filnamnet (i SINTRAN) kan omdefinieras med kommandot ****FLATA-FILER**.
 Följande tabell visas:

| FLATA-filer | |
|-------------|-----------------|
| Tabellnamn | SINTRAN filnamn |
| | |

Filnamnet omdefinieras genom att redigera den högra kolumnen.

4.4 TABELLER OCH FILER I ACCESS

Om den tabell som Du har definierat skall användas av olika användare som kör ACCESS från olika SINTRAN-användarområden, och som skall ha tillgång till samma databas, måste Du göra följande:

1. För ISAM-tabeller, använd följande kommando: ****ISAM-FILER**↵

Följande bild visas på din skärm (som exempel, har vi förutsatt att Du skapat en tabell kallade "PERSONAL"):

| ISAM filer | |
|------------|-----------------|
| Tabellnamn | SINTRAN Filnamn |
| PERSONAL | PERSONAL |

I kolumnen "SINTRAN Filnamn" måste namnet på den SINTRAN-användare där filen skall placeras fyllas i. Med andra ord, PERSONAL måste ändras till, (DATABAS)PERSONAL.

2. Skapa två filer med namnet från kolumnen "SINTRAN filnamn" med filtyperna :ISAM och :DATA. I ovanstående exempel, skulle Du behöva filerna PERSONAL:DATA och PERSONAL:ISAM under SINTRAN-användare DATABAS.

För FLATA-filer, använd samma procedur, men byt ut kommandot ****ISAM-FILER** med ****FLATA-FILER**, och skapa bara filen :DATA.

Om Du inte fyller i användarnamnet i kolumnen "SINTRAN Filnamn", kommer ACCESS senare att skapa filen för denna tabell för varje SINTRAN-användare som kör ACCESS mot denna tabell. Varje SINTRAN-användare kommer att få olika versioner av tabellen. Data kommer att spridas på olika filer, och en ACCESS-fråga kommer bara att hitta en del data: den del som hör till den SINTRAN-användare som just då kör ACCESS. Resultatet blir alltså olika databaser med samma namn.

Detta kan vara en praktisk lösning när fler användare har sina egna data med en generell beskrivning, t ex privata konton. De måste ha sitt eget SINTRAN-användarområde. Då behöver Du bara skapa en beskrivning i DBA-programmet som täcker flera databaser.

Å andra sidan, om Du vill ha bara en databas, måste Du inkludera SINTRAN-användarnamnet i filnamnet!

4.5 HUR DU DEFINIERAR SIBAS-TABELLER

En SIBAS-realm kan definieras som en tabell i ACCESS. SIBAS- "single items" kommer att vara dataelementen i tabellen.

Den SIBAS-databasen (och systemnummer) som skall användas, specificeras när ACCESS-databasen skapas (se sida 24).

SIBAS-databasen kan definieras manuellt, såsom ISAM- och FLATA-tabeller; eller de kan definieras automatiskt, genom att kopiera informationen direkt från beskrivningen i SIBAS-DRL.

Automatisk tabelldefinition

Följande kommando kan användas:

- **VISA-REALMS Kommer att visa alla realmer i SIBAS-databasen
- **KOPIERA-REALM Kopierar en SIBAS-realmbeskriv. till en ACCESS-tabell-beskrivning inklusive alla SIBAS-"items", alla "group items" <1> , och alla "sets" som använder en given realm.
- **KOPIERA-DATABAS Kopierar alla SIBAS-realm-beskrivn. till ACCESS-tabell-beskrivning, inklusive alla "items", "group items" och "sets".

Om Du har SIBAS, version D, finns det en begränsning i kommandona **KOPIERA-REALM och **KOPIERA-DATABAS: set-definitionerna som skapas kan vara ofullständiga. I detta fall får du följande meddelande:

"Set-definitionen ofullständig. Använd kommandot REDIGERA-SET."

När Du nu använder kommandot **REDIGERA-SET, kommer det att finns några blanka fält som måste fyllas i med information från SIBAS-databas-definitionen.

<1> "group items" definieras inte som ACCESS-dataelement. Om Du vill ha det, måste Du göra det manuellt.

Namnet på tabellen och dataelement som skapats med **KOPIERA-REALM och **KOPIERA-DATABAS, kommer att bli identiska med realm- och item-namnen i SIBAS. Om Du vill att tabell- och dataelementnamnen som ACCESS visar på skärmen skall vara mera förklarande, kan Du ändra dem med kommandona **SIBAS-REALMS och **SIBAS-ITEMS. Kommandot **SIBAS-ITEMS kommer att ge följande bild på skärmen:

| PERSONAL | |
|-----------------|-----------|
| Dataelementnamn | Item-namn |
| NAMN | NAMN |
| CHEF | CHEF |
| LÖN | LÖN |
| AVD | AVD |

Vänstra kolumnen innehåller dataelementnamnen som ACCESS-användaren ser på skärmen. Dessa kan redigeras; t ex, "AVD" kan ändras till "AVDELNING".

Manuell tabelldefinition

Kommandona **SKAPA-TABELL, **REDIGERA-TABELL och **VISA-TABELLER används på samma sätt som när Du definierar ISAM- och FLATA-tabeller (se kapitel 4.3).

Observera att Du kan inte använda kommandon för att ändra strukturen på SIBAS-databasen. Din definition måste stämma överens med den definition som redan finns i SIBAS-DRL. Följande söknyckel-typer kan definieras:

- PRIMÄR-NYCKEL - Index-nyckel, duplikat inte tillåtet
- SEKUNDÄR-NYCKEL - Index-nyckel, duplikat tillåtet
- PRIMÄR-CALC - CALC-nyckel, duplikat inte tillåtet
- SEKUNDÄR-CALC - CALC-nyckel, duplikat tillåtet

Du kan ge tabeller och dataelement vilket namn som helst, men eftersom dessa pekar på realmer och items i SIBAS, måste Du använda kommandona ****SIBAS-ITEMS** och ****SIBAS-REALMS** efteråt. Låt oss igen använda ****SIBAS-ITEMS** som ett exempel:

| PERSONAL | |
|-----------------|-----------|
| Dataelementnamn | Item-namn |
| Namn | Namn |
| Chef | Chef |
| Lön | Lön |
| Avdelning | Avdelning |

I det tidigare exemplet hade båda kolumnerna det fullständiga SIBAS-item-namnet, kopierat direkt från SIBAS, och därför måste vi ändra den vänstra kolumnen för att få mera fullständiga dataelementnamn. Här har båda kolumnerna dataelementnamnen definierade med kommandot ****REDIGERA-TABELL**, och därför måste vi ändra den högra kolumnen för att den skall stämma överens med SIBAS-item-namnen. Eftersom namnen i denna kolumn användas av ACCESS för att nå data från SIBAS-databasen, måste namnen vara identiska med dem i SIBAS-DRL.

"Group items" och "sets" i SIBAS-databasen definieras med kommandona ****REDIGERA-GROUPS** och ****REDIGERA-SETS**. Detta kan göras för att uppnå maximal effektivitet. (****REDIGERA-SETS** är bara relevant om Du har fler än en SIBAS-tabell i databasen och de tillhörande realmerna är förbunda med ett set.)

4.6 HUR DU STRYKER EN TABELL-BESKRIVNING

Ge kommandot:

```
**STRYK-TABELL←↓
```

med tabellnamnet som parameter, så stryks tabellbeskrivningen. Filen kommer inte att beröras av detta. Tabellen kan förekomma i andra beskrivningar och alla beskrivningar där tabellen är inkluderad, kommer att strykas. T ex, en struken tabell kommer inte att visas på skärmen, även om ISAM-filerna fortfarande finns.

4.7 HUR DU FÅR EN LISTA PÅ TILLGÄNGLIGA TABELLER I DATABASEN

Kommandot:

```
**VISA-TABELLER←↓
```

visar en lista på alla tabeller i databasen.

4.8 AVSLUTNING PÅ TABELLDEFINITIONSNIVÅN

För att avsluta nivå 2 - tabelldefinitions-nivån - och gå tillbaka till startnivån, tryck på



eller ge kommandot:

****SLUT**↵

Genom att ge SLUT två gånger på denna nivå, går Du ur DBA-programmet.

4.9 ATT SKAPA OCH REDIGERA EN TABELL FRÅN ACCESS

Kommandot *SKAPA-TABELL kan också användas på startnivån i ACCESS. Här används det till att redigera en tabell eller för att skapa en ny tabell.

Maximalt antal element som detta kommando kan hantera, är 18. För tabeller med fler än 18 element, måste Du använda DBA-programmet.

4.9.1 SKAPA EN TABELL FRÅN BÖRJAN

Ge kommandot:

***SKAPA-TABELL**↵

ACCESS kommer först att visa en meny med de tillgängliga databaserna, och sedan fråga efter följande information:

| | |
|-------------------|---|
| Skriv tabellnamn: | Detta måste vara ett tabellnamn som inte redan finns. |
| Skriv tabelltyp: | Skriv ISAM, SIBAS eller FLAT. |
| Skriv filnamn: | För ISAM-tabeller eller FLATA-tabeller, skriv namnet på filen där data i tabellen skall lagras. För SIBAS-tabeller, skriv realm-namnet. |

Det är, naturligtvis, också möjligt att specificera allt detta tillsammans med kommandot på en rad.

Följande bild visas på skärmen:

| TABELLDEFINITION | | |
|------------------|---------|-----------|
| Dataelementnamn | Datatyp | Nyckeltyp |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Du kan nu fylla i den information Du önskar i bilden på skärmen.
 Varje rad representerar ett dataelement i tabellen, och ordningen på dataelementen är densamma i tabellen som på skärmen.
 Varje dataelement beskrivs genom följande tre fält:

- Dataelementnamn Ett namn på upp till 32 tecken, som är unikt inom tabellen. Att det är unikt, kommer att kontrolleras av ACCESS.

- Nyckeltyp Möjliga nyckeltyper är PRIMÄR, SEKUNDÄR eller ingen. Se sidan 31 för mer information.

 Nycklar används inte tillsammans med FLATA-filer.

Datotyp

Det är möjligt att använda de datatyper som definierats med kommandot REDIGERA-DATATYPER i DBA-programmet. Som tillägg finns det några fördefinierade datatyper tillgängliga. Om Du använder en databas som redan finns (t ex, en SIBAS-databas), behöver Du de datatyper som använts i den databasen. Om Du skapar en ny databas för ACCESS, kommer typerna CHARACTER, NUMERIC och FILLER vanligen att vara tillräckligt:

CHARACTER-n

Använd detta för alla text-element. Det kan innehålla upp till n tecken. Till exempel, ett dataelement som använder datatypen CHARACTER-3 kan innehålla upp till 3 tecken. Det kan också innehålla tal, men Du kan inte använda det i uträkningar. Denna typ kan jämföras med lagringsformatet TEXT i DBA.

NUMERIC-n

Använd detta för alla dataelement som skall innehålla tal som Du vill använda i uträkningar. "N" är max. antal tecken som talet kan bestå av: antalet siffror plus ett extra för decimalpunkt, om Du tänker använda dig av decimaler.

FILLER-n

Om Du definierar ett dataelement som filler, kommer ACCESS bara att ignorera det och därför kommer det inte att bli använt i tabellen. Detta är användbart om du vill skapa en annan tabell som använder samma data och samma fil. Exempel: Du vill definiera en tabell som använder samma data som "employees"-tabellen i DATABAS. Denna tabell har definierats enligt följande:

| | |
|------|--------------|
| NAMN | CHARACTER-32 |
| CHEF | CHARACTER-32 |
| LÖN | INTEGER-4 |
| AVD | CHARACTER-20 |

Om Du vill skapa en tabell som bara innehåller namnen och lönen, kan du göra enligt följande:

| | |
|-------|--------------|
| NAMNE | CHARACTER-32 |
| CHEF | FILLER-32 |
| LÖN | INTEGER-4 |
| AVD | FILLER-20 |

Naturligtvis, måste SINTRAN-filnamnet för den nya tabellen vara definierad som samma som för den gamla tabellen.

De kvarvarande fördefinierade datatyperna kan vara nödvändiga om ACCESS skall använda data skapade av andra program, eller data i en SIBAS-databas. Dessa typer är:

| | |
|-----------|---|
| INTEGER-2 | - En 16-bit heltal |
| INTEGER-4 | - En 32-bit heltal |
| REAL-4 | - En 32-bit flyttal |
| REAL-6 | - En 48-bit flyttal |
| REAL-8 | - En 64-bit flyttal |
| BCD-n-m | - En BCD variabel med n siffror före decimalpunkt och m siffror efter |

Mer information om de lagringsformat som motsvarar dessa datatyper, kan Du hitta på sidan 20.

Du kan flytta mellan fälten med följande tangenter:



Flyttar till fältet ovanför.




Flyttar till fältet nedanför.



Flyttar till nästa fält. Nästa fält kan vara det första fältet på raden nedanför.



Flyttar till föregående fält. Föregående fält kan vara sista fältet på raden ovanför.

SKIFT+ 

Stryker aktuellt element.

CTRL+L

Lägger till ett element före aktuellt element.



el-



Avslutar kommandot utan att lagra tabellbeskrivningen.



Lagrar tabell-beskrivningen (i Data Dictionary)

4.9.2 HUR DU ÄNDRAR EN TABELL-BESKRIVNING

Exemplet tidigare visar hur Du kan definiera en tabell från början.

Vad händer om du använder kommandot *REDIGERA-TABELL för att ändra en tabell som redan existerar?

För att ändra beskrivningen över tabellen "PERSONAL", kan Du starta med kommandot (du måste ha gått in i ACCESS under användarnamnet ACCESS för att göra detta):

*REDIGERA-TABELL DATABAS PERSONAL↵

ACCESS svarar med:

Skriv det nya tabellnamnet: Det gamla namnet visas.

Om Du ändra detta, kommer det att finnas två tabeller, en med det gamla namnet och en med det ändrade namnet. Detta kan vara användbart om Du vill skapa en ny tabell som är mycket lika den gamla.

Skriv tabelltyp: Tabelltypen kan ändras. Du kan, till exempel, ändra en tabell från FLAT till ISAM. Databasen måste då omladdas med kommandot *LADDA-DATABAS.

Skriv filnamn: SINTRAN-filnamnet/SIBAS-realmen kan ändras.

Elementen i tabellen visas på skärmen och Du kan redigera dem. Ett exempel:

| T A B E L L D E F I N I T I O N | | |
|---------------------------------|--------------|-----------|
| Dataelement | Datatyp | Nyckeltyp |
| Kundnamn | character-30 | primär |
| Adress | character-40 | |
| Postadress | Postadress | sekundär |
| Telefonnummer | telefon | sekundär |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

4.10 HUR DU SKAPAR OLIKA SYNLIGA OMRÅDEN

Olika användare kan ha tillgång till olika delar av samma information. Eftersom rättigheter bara kan definieras för en hel databas i ACCESS, måste detta ske genom olika ACCESS-databaser definieras mot samma filer eller samma SIBAS-databas. Ett exempel, låt oss säga att två användare, ANVÄNDARE-A och ANVÄNDARE-B, skall arbeta mot olika delar på samma SIBAS-databas:

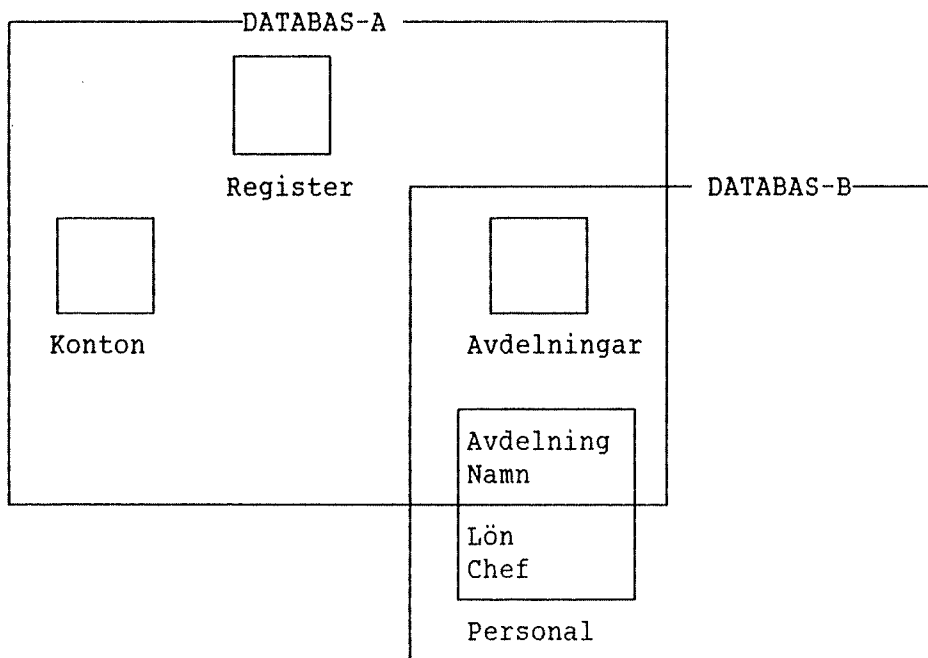
ANVÄNDARE-A

Avdelning
Register
Konton
Personal - bara avdelning och namn

ANVÄNDARE-B

Avdelning
Personal

För att göra detta möjligt, måste Du definiera två överlappande ACCESS databaser. DATABAS-A för ANVÄNDARE-A och DATABAS-B för ANVÄNDARE-B:



Tabellerna Personal och Avdelningar måste definieras i både DATABAS-A och DATABAS-B.

Tabellen Personal måste definieras på två olika sätt. Om det är en SIBAS-realm, betyder detta bara att Du definierar tabellen så att den inte innehåller alla items i realmen. Med ISAM- eller FLAT-fil, måste de fält som inte skall synas definieras som FILLER (se sidan 39).

K A P I T E L 5

KOMMANDOT LADDA-DATABAS I ACCESS

5 KOMMANDOT LADDA-DATABAS I ACCESS

Bara den användare som har definierat databasen har rätt att använda detta kommando, som gör det möjligt att ladda nya data i databasen eller ändra strukturen på filerna. Detta kan vara nödvändigt efter att ett nytt element har lagts till.

Sekvensen startas på startnivå i ACCESS, med kommandot:

*LADDA-DATABAS <databasnamn>←

Data kan laddas från två olika typer av filer.

TEXT-fil Data är lagrat som text och måste konverteras innan det läggs in i en tabell. Posterna är separerade med tecken för CR/LF. En sådan fil skapas i NOTIS-WP eller PED med en post per rad. Kommandot ***LAGRA-UTAN-RAM skapar också filer av denna typ. OBS! 16 bitars format binärt.

DATA-filer Data lagras i ett "binary format". Posterna har en fast längd och är inte separerade med skiljetecken. Kommandot ***LAGRA-DATA skapar också sådana filer.

Vi har följande kommandon:

****LADDA-TEXT-FIL <tabellnamn>,<filnamn>**

Laddar data från en text-fil till en tabell.
(SIBAS, ISAM eller "FLAT".)

****LADDA-DATA-FIL <tabellnamn>,<filnamn>,<postlängd i bytes>**

Laddar data från en data-fil med en given
postlängd till en tabell. (SIBAS, ISAM eller "FLAT".)

****OPTIMERA-TABELL <tabellnamn>**

Sorterar tabellen enligt önskade index-nycklar och
laddar om tabellen. Bara för ISAM-tabeller.

Kommandona **LADDA-TEXT-FIL och **LADDA-DATA-FIL ger följande bild på skärmen för tabellen PERSONAL (efter det att Du fått frågan om Du vill ha standardvärden inlagda):

| PERSONAL | NAMN | CHEF | LÖN | AVDELNING |
|------------|------|------|-----|-----------|
| → Datatyp: | | | | |
| → Start: | | | | |
| → Längd: | | | | |

Om Du svarade "ja" på frågan om standardvärden, kommer ACCESS att slutföra tabellen med värden i fälten. Du kan ändra dessa värden om det är nödvändigt.

Parametrarna Datatyp, Start och Längd har följande mening:

Datatyp indikerar fältets datatyp. Om Du använt text-filer, skall datatypen alltid vara CHARACTER. Tal konverteras automatiskt från text till binärt format.

Om Du använder data-filer, kan Du ha följande datatyper:

| | |
|-----------|----------------|
| CHARACTER | Text fält |
| NUMERIC | Numerisk text |
| INTEGER-2 | 16 bit integer |
| INTEGER-4 | 32 bit integer |
| REAL-4 | 32 bit real |
| REAL-6 | 48 bit real |
| REAL-8 | 64 bit real |
| BCD | BCD fält |

Om Du lämnat datatyps-fältet blankt, kommer ACCESS att ge denna kolumn värdet blank eller noll i alla poster.

Start indikerar var i posten från indata-filen fältet börjar. Första positionen är 1.

Längd talar om hur långt fältet är i indata-filen (antalet bytes). Om längden är mindre än beskriven i DDI (postbeskrivningen), kommer restlängden att fyllas med blanka. Om längden är större, kommer den överskjutande delen att tas bort.

Om Du har ett fält i posten som Du inte vill ha med, kan Du utesluta detta genom att sätta start i fältet efter det fält Du inte vill ha med. Systemet kontrollerar att fälten inte överlappar varandra, till exempel om du ger fel start- och längd-värden. Om start- och längd-värden inte ges, kommer det värde som är definierat i DBA att tas som standardvärde.

Kommandon kan också användas för att omdefiniera databas-tabellerna. Se exempel 3, sidan 49.

Exempel 1: Ladda-text-fil

Antag att posterna i en indatafil har följande struktur:

Start: 1 33 44 76 85

| | NAMN | FÖDD | CHEF | LÖN | AVDELNING |
|--------------------------------|------|------|------|-----|-----------|
| Antal tecken: Postlängd: 96 | 32 | 11 | 32 | 9 | 12 |

Du vill ladda denna fil i tabellen "Personal". Detta betyder att fältet FÖDD måste uteslutas under laddningen. Fyll i tabellen på följande sätt efter att Du givit kommandot **LADDA-TEXT-FIL:

| PERSONAL | NAMN | CHEF | LÖN | AVDELNING |
|------------|-----------|-----------|-----------|-----------|
| → Datatyp: | character | character | character | character |
| → Start: | 1 | 44 | 76 | 85 |
| → Längd: | 32 | 32 | 9 | 12 |

Du avslutar denna sekvens genom att trycka på:



ACCESS börjar sedan läsa data från indata-filen. Dina data presenteras på skärmen i den vanliga resultattabellen och Du kan kontrollera att laddningen gick bra.

Du avslutar denna sekvens med kommandot SLUT, och Du får då frågan:

Vill Du lägga data in i tabellen? J/N.

- a) Om Ja, blir databasen laddad med de registrerade data efter frågan om den gamla informationen skall strykas eller inte.
- b) Om Nej, får Du frågan:

Vill du redigera tabell-beskrivningen? J/N

- i) Om Ja, kan Du ändra post-beskrivningen dvs datatyp, start, längd.
- ii) Om Nej, kommandot avslutas.

Exempel 2: Optimera-tabell

När Du använder en tabell för att stryka, uppdatera och lägga till data, blir innehållet i tabellen dåligt organiserad efter ett tag. Detta betyder att söktiden blir lång.

För att placera data på ett sådant sätt att söktiden blir kort, kan Du använda kommandot:

```
**OPTIMERA-TABELL←↓
```

Detta kommando omstrukturerar innehållet i en tabell för att minimera accesserna till disk. Detta uppnås genom att fysiskt stryka alla poster som strukits och sortera data enligt de mest frekvent använda index-nycklarna.

VARNING: Detta kommando skall aldrig ges utan att man först tagit en backup på datafilerna. Om det uppstår fel, kan all information i tabellerna strykas.

Ett exempel, låt oss optimera tabellen PERSONAL: Skriv **OPTIMERA-TABELL PERSONAL

Följande bild visas på skärmen. Svar skrivna av användaren är understrukna:

| | |
|----------------------|----------|
| Tabellnamn: PERSONAL | |
| Söknyckel | Duplikat |
| NAMN | NEJ |
| CHEF | JA |
| AVDELNING | JA |

Hur många nycklar vill Du sortera på: 1

1. Sorteringsnyckel: AVDELNING

Data i tabellen sorteras nu, och ISAM-filerna regenereras. Tabellen är nu optimerad för sökning i kolumnen AVDELNING.

Observera att kommandot bara är för ISAM-filer, och inte för SIBAS-realmer och FLATA-filer.

Exempel 3: Ändra tabelldefinitionen

Om du vill ändra några av elementen eller lägga till nya i en befintlig tabell, måste detta göras i tre steg.

1. Använd ACCESS för att skriva ut data i orginaltabellen med ***LAGRA-DATA (eller ***LAGRA-TEXT)
2. Gå in i DBA och omdefiniera tabell-beskrivningen.
3. Använd ACCESS för att ladda data till den nya tabellen genom att använda **LADDA-DATA-FIL (eller **LADDA-TEXT-FIL). Svara "ja" på frågan om gammal information skall strykas.

Exempel

Du vill ändra tabellen PERSONAL så att NAMN och CHEF skall ha 40 tecken. LÖN skall ersättas av LÖN.GRUPP, och Du vill utöka posten till att inkludera elementet ANST.DAT (anställningsdatum).

1. Innehållet i den gamla posten lagras. Du utför följande:

| PERSONAL | NAMN | CHEF | LÖN | AVDELNING |
|----------|--------|--------|-----|-----------|
| → | skriv. | skriv. | | skriv. |

Med kommandot *****LAGRA-DATA**, skriver Du resultatet till filen TEMP:DATA. Postlängden är 32+32+20=84 bytes.

2. Gå sedan in i DBA och ändra tabell-beskrivningen för PERSONAL.
3. Gå in i ACCESS och skriv följande:

***LADDA-DATABAS EXEMPEL-BAS←**

****LADDA-DATA-FIL PERSONAL TEMP:DATA←**

Svara "nej" på frågan om standardvärden, och "84" på frågan om postlängden. Följande tabell visas på skärmen och Du kan fylla i den enligt nedan:

| PERSONAL | NAMN | CHEF | LÖN.GRUPP | AVDELNING | ANST.DAT |
|------------|------|-----------|-----------|-----------|----------|
| → Datatyp: | CHAR | CHARACTER | | CHARACTER | |
| → Start: | 1 | 33 | | 65 | |
| → Längd: | 32 | 32 | | 20 | |

Värdet på LÖN-GRUPP och ANST.DAT sätts till 0 eller blank, beroende på om fälten är definierade som numeriska eller alfanumeriska. De nya positionerna i NAMN och CHEF är blanka.

KAPITEL 6

ALLMÄNT OM KÖRNING AV ACCESS-SYSTEMET

6 ALLMÄNT OM KÖRNING AV ACCESS-SYSTEMET

6.1 KOMMANDOFILER

Systemet använder kommandofilerna ACCESS-HELP:TEXT och DBA-HELP:TEXT. Innan systemet startas, måste önskad språkversion kopieras till dessa filer från ACCESS-NO-D:TEXT, ACCESS-SW-D:TEXT eller ACCESS-EN-D:TEXT, som levereras med systemet.

Olika SINTRAN-användare kan ha egna versioner av ACCESS-HELP:TEXT under sina egna användarområden, så att de kan använda den språkversion de föredrar. Det är också möjligt att i NOTIS-WP ändra på HELP-filerna både i ACCESS och i DBA. Du kan översätta HELP-filerna till ett annat språk, så att alla kommandon, meddelanden och hjälpinformationen ges på detta språk. Det är viktigt att ingen rad stryks eller läggs till i HELP-filen. Du skall bara skriva över de befintliga raderna. Om Du önskar stryka ett kommando så att användaren inte kan använda det, så finns det två sätt att göra det på, beroende på typen av kommando:

1. Om kommandot är en operator (ett av tabellkommandona som måste åtföljas av en punkt), ersätt detta med order VOID.

Till exempel, om Du vill förhindra att användaren kan stryka en post i databasen, ersätt order STRYK med VOID på följande rad i användarens ACCESS-HELP-fil:

```
STRYK                % 2 Stryk en post
```

Användaren kan då inte använda operatören "stryk" i tabeller. Orden till höger om "%" -tecken behandlas av ACCESS som kommentar.

2. Om kommandot inte är en operator, ersätt det med blanka.

För att förhindra en användare från att använda RESERVERA-DATABAS, blanka ut "RESERVERA-DATABAS" på följande rad i användarens ACCESS-HELP-fil.

```
RESERVERA-DATABAS   % Låsa tabeller
```

OBS! Om användaren har en fil kallad ACCESS-HELP:INIT, kommer ACCESS att läsa från denna fil istället för från ACCESS-HELP:TEXT. Detta betyder att för att få den ovannämnda processen att fungera, måste Du stryka filen ACCESS-HELP:INIT eller "dumpa" kommandofilerna (detta beskrivs i följande kapitel).

Vi föreslår att Du skyddar HELP-filen genom att ge den enbart READ (R) tillgång. Det kan bli förvirrande för ACCESS-användaren om kommandonamnen ändras för ofta.

Om ACCESS-HELP inte finns under användarområdet från vilken ACCESS körs, och inte heller under användaren SYSTEM, används alltid versionen under DATA-DICTIONARY.

6.1.1 KOMMANDO DUMP-HELP-FIL I ACCESS

Kommandot DUMP-HELP-FIL, på startnivån i ACCESS, konverterar kommandofilen till ett format som kan läsas snabbare av ACCESS, och placerar resultatet i en fil kallad ACCESS-HELP:INIT. Den används på följande sätt:

1. Kontrollera att det finns en fil kallad ACCESS-HELP:SYMB under användare DATA-DICTIONARY, eller, om Du gör detta med en speciell version av kommandofilen för en enskild användare, under den användaren.
2. Skapa en kontinuerlig fil (storlek 60 sidor) kallad ACCESS-HELP:INIT under samma användare. Om användaren redan har denna fil, måste den strykas och skapas på nytt. (Med andra ord, denna fil måste vara tom.)
3. Gå in i ACCESS och ge kommandot DUMP-HELP-FIL. ACCESS kommer att svara med meddelandet: "Kommandofile har dumpats".

6.1.2 STANDARD TRANSAKTIONSIBLIOTEK OCH STANDARDTRANSAKTION

Standard transaktionsbiblioteket är ett bibliotek som öppnas automatiskt när användaren går in i ACCESS. Namnet på filen har givits på rad 5 på filen ACCESS-HELP:TEXT, och kan ändras genom att filen redigeras i NOTIS-WP.

En standardtransaktion är en transaktion som utförs automatiskt när en användare går in i ACCESS. Tillsammans med MENY-kommandon i kommandoboxen i ACCESS, görs det möjligt att ge användaren en meny när han/hon går in i ACCESS. (se ACCESS Användarhandbok).

Namnet på standardtransaktionen skrivs in med början i position ett på rad 6 i filen ACCESS-HELP:TEXT.

6.2 FLERANVÄNDARTRANSAKTIONSBIBLIOTEK

Det är möjligt att ha ett transaktionsbibliotek som kan användas samtidigt av många användare. Filens SINTRAN-rättighet måste vara begränsad till READ (R).

Exempel: Du har ett transaktionsbibliotek kallad COMMON-LIB:TRAN under användare ACCESS. På denna fil har Du en "standard"-transaktion som skall användas av flera olika användare. För att göra detta möjligt, logga in som användare ACCESS och ge kommandot:

```
@SET-FILE-ACCESS COMMON-LIB:TRAN R R R
```

6.3 NÅGONTING ÄR FEL I ARBETSOMRÅDET

Möjlig orsak till detta fel är:

1. Arbetsområdets filer finns inte eller är placerade på ett annat användarområde, till vilket Du inte har läs- och skriv-rättigheter.
2. Arbetsområdets filer används av någon annan användare. Kontrollera användar-definitionen för att se om två användare delar på samma arbetsområde. Om det är så, ändra namnet på ett av arbetsområdena.

(Se sida 18 hur Du definierar användare.)

6.4 DDI INTE AKTIV

Detta meddelande visas om någon försöker köra ACCESS-huvudsystem utan att DDI-processen är aktiv i en batch-process. Det kan vara att Du har glömt att starta batch-processen "ACCESS-DDI-D:BATC" (se kapitel 2). Det kan också bero på att ACCESS-processen inte kan kommunicera med DDI-processen för att numret i ACCESS-HELP och ACCESS-DDI-D:BATC filerna för kommunikations-"devices" inte överensstämmer, eller inte finns i operativsystemet (se sektion 6.5, samt installationsövning och PD-ark).

6.5 KOMMUNIKATIONSPROBLEM MELLAN DDI OCH ACCESS

Kommunikationen mellan ACCESS-huvudsystem och den aktiva DDI-processen använder en mekanism i SINTRAN kallad "internal devices". Ett "internal device" är ett buffer-område i SINTRAN som överför data mellan processer. En process reserverar bufferten för läsning, den andra för skrivning. Två INTERNAL DEVICES behövs för att ACCESS skall kunna kommunicera med DDI.

Alla "internal devices" får ett nummer när SINTRAN-systemet genereras (se SINTRAN-manualen). ACCESS använder vanligtvis nummer 128 och 129 för kommunikation med DDI, men andra nummer kan väljas om dessa "devices" redan används av andra system.

Numret på "internal devices" som används finns i början av filerna ACCESS-HELP:TEXT och ACCESS-DDI-D:BATC.

Numren i de två filerna måste överensstämma!

Om "internal devices" är upptagna av andra system, kan eventuellt lediga SIBAS "internal devices" användas (från 160 till 166 eller 172). Kom ihåg att skriva in numren på "internal devices" i både ACCESS-DDI-D:BATC och i ACCESS-HELP-filerna.

6.6 FÖRBÄTTRINGAR I ACCESS SÖKTID

En SINTRAN-fil är ett dataområde på en disk som kan vara kontinuerlig eller indexerad. Sökning är vanligtvis snabbare om det är en kontinuerlig fil, men problem kan uppstå när filen är full. Det kan hända att filen inte kan expanderas på ett kontinuerligt diskområde.

En indexerad fil expanderar dynamiskt, men sökningen är baserad på en indextabell på sidorna. Indextabellen är upprättad av SINTRAN-filsystem.

Båda filtyperna (vilket inte har något att göra med :TYPE) skapas genom kommandot:

```
@CREATE-FILE < SINTRAN filnamn >,< antal sidor >←↓
```

Om <antal sidor> är lika med 0, kommer filen att bli indexerad. Varje annat nummer ger en kontinuerlig fil.

När en kontinuerlig fil är full, måste den expanderas om möjligt. Om detta inte är möjligt, måste du redefiniera filen med en såkallad "unload"- och "load"-operation (se kapitel 5). Transaktioner på en kontinuerlig fil går snabbare, men Du tappar flexibilitet.

Observera att en indexerad fil kan kopieras in i en kontinuerlig fil och vice versa.

Speciella verktyg har utvecklats för att underhålla SIBAS- och ISAM-filer. De är inte integrerade i denna release av DBA. Om fel uppstår i en fil, om poster försvinner eller konstiga felmeddelande visas, är det möjligt att använda programmet ISAM-SERVICE för ISAM-filer för att kontrollera filernas innehåll. Du kan också utföra såkallade "unload/load"-operationer från detta subsystem.

Den generelle dokumentationen på SINTRAN, operativsystemet, är omfattande. Vi föreslår att Du läser SINTRAN III Time-sharing/Batch Guide som beskriver det viktigaste om filsystemet.

6.7 ANDRA FELMEDDELANDEN

Vanligtvis är ett felmeddelande från ACCESS en klar text som kan efterföljas av ett felmeddelandenummer.

Felmeddelandet överensstämmer med SINTRAN-felmeddelanden, som också beskrivs i ACCESS Användarhandbok.

SIBAS och ISAM kan också ge felmeddelanden, (beskrivna i SIBAS och ISAM-manualerna). SIBAS kommer att skriva ut felmeddelandet på konsolen ("error device") om någonting allvarligt inträffar. ISAM försöker alltid återvända till ACCESS med en felstatus, men situationer kan uppstå när ISAM "kolapsar". Ett meddelande som stämmer överens med detta kan skrivas ut:

ISAM FATAL ERROR IN ROUTINE 'ZABUF'/1

och kontrollen går till SINTRAN. I detta fall skall du kontrollera om filen har restrukturerats, om ACCESS kör mot korrekt fil, etc.

K A P I T E L 7

FULLSTÄNDIGA EXEMPEL

7 FULLSTÄNDIGA EXEMPEL

7.1 EXEMPEL PÅ HUR DU DEFINIERAR ANVÄNDARE

Logga in som SINTRAN-användare DATA-DICTIONARY. Stoppa DDI-processen som beskrivits i kapitel 2.1, och gå in i DBA-programmet genom att skriva:

```
@DBA←↵
```

Vi förutsätter att systemet har levererats med den testdatabas som används i manualen. Användare ACCESS är därför redan definierad som en DBA-användare, utan lösenord.

Systemet frågar efter användar-identifikation, och Du svarar A, som är en förkortning för "ACCESS".

Då visas HJÄLP-bilden på skärmen, som visar de kommandon som finns på nivå 1.

Först måste du definiera de användare som skall tillåtas köra ACCESS-delen av systemet. Sedan vem av dessa användare som tillåts använda DBA-programmet. Starta genom att ge kommandot:

```
*RED-ANV←↵
```

som är en förkortning av *REDIGERA-ANVÄNDARE. Du får upp en tabell över ACCESS-användare som redan är definierade på skärmen. Definiera MIG och ALBERT-NEUMANN som användare:

| ACCESS systemanvändare | | |
|------------------------|----------|----------------------------|
| ACCESS användarnamn | Lösenord | Arbetsområde |
| ACCESS | | (ACCESS)WORKAREA |
| MIG | MI | (ACCESS)WORKAREA-1 |
| ALBERT-NEUMANN | | (ALBERT-NEUMANN)WORKAREA-2 |

Användaren MIG måste från och med nu ge lösenordet MI när den går in i ACCESS, men ALBERT-NEUMANN behöver inte ge något lösenord.

MIG måste alltid logga in som SINTRAN-användare ACCESS (eller en användare definierade som "friend" till denna användare). Annars kan inte MIG använda arbetsområdet (ACCESS)WORKAREA-1.

Gå ut ur tabellen genom att trycka på HEMMA-tangenten.

Du vill definiera MIG som en DBA-användare. MIG skall tillåtas att gå in i DBA-programmet och skapa databaser; ALBERT-NEUMANN får bara tillåtelse att arbeta mot ACCESS' databaser som är definierade av andra.

Kommando

*DBA-ANVÄ↵

som är en förkortning av *DBA-ANVÄNDARE, ger en tabell på DBA-användare på skärmen. Användare ACCESS är redan definierad och Du skall skriva in MIG:

| |
|-----------------------|
| Databas Administratör |
| DBA-användare |
| ACCESS |
| MIG |

Nu skall Du gå ur DBA-programmet, så tryck på SLUT-tangenten (eller ge kommandot *SLUT). När Du skapar en databas, kommer användarnamnet Du gav när Du gick in i DBA att registreras som ägare till databasbeskrivningen. I detta fall har Du loggat in som DBA-användare ACCESS, och eftersom Du vill att MIG skall vara ägare till databasen Du skall skapa skall Du gå ur DBA och gå in igen men denna gång som användare MIG.

7.2 EXEMPEL PÅ HUR DU DEFINIERAR EN DATABAS

Du vill definiera en databas EXEMPEL-BAS med MIG som ägare. MIG kommer då att vara den enda användaren som tillåts ändra beskrivningen på denna databas.

Gå in i DBA-programmet genom att skriva

@DBA↵

och ge MIG och lösenordet MI. Du kommer in på startnivån (markerad med * i kommandopositionen) och ge här kommandot:

*SKAPA-DATABAS EXEMPEL-BAS↵

Du har nu kommit in i tabelldefinitionsnivån nivå 2.

Här skall Du definiera några datatyper för dataelementen. Ge kommandot

****REDIGERA-DATATYPER**↵

Följande tabell visas på din skärm och Du skall fylla i den som visas nedan:

| Datatyper | | |
|--------------|-----------------|----------------|
| Datatypsnamn | Utskriftsformat | Lagringsformat |
| CH-20 | X(20) | TEXT(20) |
| CH-32 | X(32) | TEXT(32) |
| I-4 | -Z ZZZ ZZZ ZZ9 | INTEGER4 |
| N-3 | ZZ9 | NUM(3) |

Det är tillräckligt att fylla i namn-fältet och lagringsformat, och lämna till systemet att välja värden för utskriftsformatet.

Du har nu definierat en tabell som Du senare kan definiera och ställa frågor mot.

Ge följande kommando:

****SKAPA-TABELL PERSONAL**↵

DBA frågar efter tabelltyp:

Skriv in tabelltyp (ISAM, SIBAS eller FLAT): ISAM

och då får Du följande bild på skärmen som skall fyllas i enligt nedan:

| PERSONAL | | | |
|---------------------|-----------------------|--------|-----|
| Namn på dataelement | Datatyper som använts | Nummer | Sök |
| NAMN | CH-32 | 1 | |
| CHEF | CH-32 | 2 | |
| LÖN | I-4 | 3 | |
| AVDELNING | CH-20 | 4 | |

| PERSONAL | | | |
|----------|-----------|-------------|----------|
| Nummer | Söknyckel | Kolumnbredd | Radlängd |
| 1 | P | | |
| 2 | S | | |
| 3 | S | | |
| 4 | S | | |

För kolumnbredd och radlängd, låter vi systemet välja värde.

Sedan måste ISAM-filern definieras. Ge kommandot:

****ISAM-FILER←↓**

och tabellen fylls i enligt nedan:

| ISAM-Filer | |
|------------|------------------|
| Tablenamn | SINTRAN filnamn |
| PERSONAL | (ACCESS)PERSONAL |

Tabellen "PERSONAL" är nu kopplad till ISAM-filen PERSONAL:DATA och PERSONAL:ISAM under SINTRAN-användare ACCESS. Första filen skall innehålla data, den andra skall innehålla söknyckeltabellen.

När du har definierat tabellen måste Du definiera vem som skall få arbeta på tabellen med ACCESS. Använd kommandot REDIGERA-BEHÖRIGHET och tabellen nedan visas:

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

Användaren "MIG" tillåts att manipulera data i tabellen, medan användaren "ALBERT-NEUMANN" inte får stryka poster i databasen.

Observera att kommandot *VISA-DATABASER och *REDIGERA-DATABAS nu kan användas eftersom databas och en tabell finns.

Detta göra att definition av en tabell är slutförd, och Du kan gå tillbaka till SINTRAN genom att trycka på SLUT-tangenten två gånger.

För att ge ACCESS informationen som definierats i DBA-programmet, måste DDI-processen vara aktiv. Detta görs genom att koppla den till en batch-process med SINTRAN-kommandot:

@APPEND-BATCH 1,ACCESS-DDI:BATC,ACCESS-DDI:LOG+J

Om det har inkluderats som ett termineringskommando i filen DBA-HELP, kommer det att utföras automatiskt varje gång Du går ut ur DBA-programmet, se kapitel 2.1.

Detta gör att databas-definitionen är slutförd, DDI är aktiv och ACCESS kan användas som det beskrivs i ACCESS Användarhandbok.

B I L A G A A

DDI-PROCESSEN SOM EN BATCH-PROCESS

ACCESS kräver information om databaser, rättigheter, etc. Denna databas-beskrivning, som definieras i DBA-programmet, som normalt lagras på filer under användare DATA-DICTIONARY, är densamma för alla användare som kör ACCESS.

En aktiv DDI-process innehåller alla beskrivningar som skrivs in i DBA-programmet och ger denna information till ACCESS, när ACCESS behöver den. Två "internal devices" används för att överföra information mellan ACCESS och DBA.

DDI-processen är ett vanligt program som inte stannar av sig själv när det en gång har startats. Programmet finns på filen (DATA-DIC)ACCESS-DDI-D:PROG. Programmet kan startas från en terminal, men uppehåller denna terminal så länge det körs. Därför är det bättre att starta upp programmet som en batch-process, för att undvika detta. Men observera att denna batch-processor kommer att vara upptagen så länge som DDI-processen körs.

Informera de andra användare på systemet att de inte får använda den batch-processor som upptas av DDI-processen.

En batch-processor måste få information om vad den skall göra från en fil kallad in-datafilen för processen. Den behöver också en ut-datafil för till exempel felmeddelanden. För att starta batch-processen, använd filen

```
(DATA-DIC)ACCESS-DDI-D:BATC
```

som in-datafil och

```
(DATA-DIC)ACCESS-DDI-D:LOG
```

som ut-datafil. Den senare måste vara tom när Du startar DDI-processen. Ge kommandot @LI-FI för att kontrollera att den har skapats.

Systemet levereras med följande in-datafil ACCESS-DDI-D:BATC:

```
@ENTER DATA-DICTIONARY,,,32000  
@(DATA-DICTIONARY)ACCESS-DDI-D:PROG  
128  
129  
CC <DDI PROCESS> CC  
@
```

Den första raden i filen berättar för batch-processen att DDI-processen skall använda databas-beskrivningen på användarområdet DATA-DICTIONARY, och att den skall köra så länge som möjligt (32000 minuter).

Du måste kanske ändra denna rad. Det generella formatet är:

```
@ENTER,<SINTRAN användare>,<lösenord>,<project lösenord>,<maximal  
tid för processen>
```

Numren för "internal devices" skall skrivas in på rad 2 och 3. Dessa nummer finns också i filen ACCESS-HELP:TEXT, och i "mode-filen" kallad ACCESS-DDI-STOP:MODE som används för att stoppa DDI-processen. Den filen innehåller följande:

```
@(DATA-DICTIONARY)ACC-DDI-STOP  
128  
129
```


B I L A G A B

INFORMATION FÖR SYSTEMANSVARIG

I denna bilaga finns det i exemplen ett tecken "%", allt efter detta tecken är kommentarer.

HUR DU SÄTTER XON/XOFF

Om ACCESS skall arbeta bra på en NOTIS-terminal TDV-2200/9-ND-NOTIS, skall "communication handshake" alltid ha satts till xon/xoff. Detta skall göras under installationen, men det kan fortfarande finnas terminaler där detta inte är gjort. Detta resulterar i att bilderna på skärmen blir "kladdiga". Proceduren för att sätta xon/xoff visas därför här.

Du måste gå in i menyn för "communication switches" på din terminal (för detaljer se, Tandberg-manualen).

Tryck på CTRL + HJÄLP två gånger och Du kommer in i CONFIGURATION MENU. Använd ned-pil tilldessa Du har placerat markören på räden "Communication switches". Tryck på ENTER-tangenten. Då får Du följande bild på skärmen:

```
C o m m u n i c a t i o n   S w i t c h e s

Send Receive Mode      Simultaneous
Echo                   External
Online                  Toggle
Communication Clock    ASY
Communication Handshake . . . .   %Måste vara XON/XOFF
Modem                   Inhibit
.
.
.
.
.
```

Flytta markören med pil-tangenterna och välj med ENTER-tangenten.

Logga in som användare SYSTEM och sök terminalnumret med @WHO. Här kommer vi att använda terminalnummer 53 som exempel. Utför följande procedur (det understrukna texten måste skrivas av dig):

```

@SINTRAN-SERVICE+↵
*CH-DATA+↵
LOG.UNIT NO.: 53D+↵          % Terminal 53 används som ett exempel.
INPUT/OUTPUT: I+↵
MEMORY? Y+↵
IMAGE? Y+↵
SAVE-AREA? Y+↵

MEMORY    IMAGE    SAVE-AREA

DFLAG/ <value> <value> <value> 101000+↵
+
*EXIT+↵
@

```

SÄTTA PRIORITET PÅ EN SATSVIS PROCESS

Batch-processen eller processer, som skall användas av ACCESS, skall köras med en högre prioritet än vanliga batch-processer. Annars kommer svarstiden för ACCESS-användarna bli onödigt lång. Som med xon/xoff, skall detta tas omhand under installationen. Men, om det blir nödvändigt att börja använda en ny batch-process för ACCESS, skall följande utföras (här är batch nr 3 använt som exempel).

Logga in som användare SYSTEM.

Sök det "logical unit number" på batch-processen. Dessa nummer ges i följande ordning:

| Batch no | Logical unit no |
|----------|-----------------|
| 1 | 670 |
| 2 | 672 |
| 3 | 674 |
| 4 | 676 |
| . | . |
| . | . |
| . | . |

Till exempel: "logical unit number" för batch nr 3 är 674. Fortsätt enligt följande:

```
@SINTRAN-SERVICE↵  
*REMOVE-FROM-TIME-SLICE 674D↵ % Logical unit number  
MEMORY? Y↵  
IMAGE? Y↵  
SAVE-AREA? Y↵  
*EXIT↵  
@PRIOR BCH03 42D↵ % Sätt prioritet till 42
```

Prioriteten kan kontrolleras med kommandot:

```
@LIST-RT-DESCRIPTION BCH03↵
```

För denna batch-process, skall prioriteten vara 42. För en vanlig batch-process, skall den vanligtvis vara mindre än 20.







ATT ÅTERSTÄLLA PRIORITETEN

Om en batch-process inte längre skall användas av ACCESS, skall Du sätta tillbaka prioriteten till det "normala" med följande procedur:

```
@SINTRAN-SERVICE↵  
*INSERT-IN-TIME-SLICE 674D↵ % Logical unit number  
MEMORY? Y↵  
IMAGE? Y↵  
SAVE-AREA? Y↵  
*EXIT↵
```


B I L A G A C

KONTROLLKOMMANDON I DBA MODULEN

-  eller CTRL+D CTRL+D stryker fältet där markören är positionerad.
- SHIFT +  eller CTRL+D CTRL+L stryker hela raden där markören är positionerad
-  eller CTRL+G visar tillgängliga kommandona i aktuell situation
-  eller CTRL+O skriver bilden på skärmen till en fil. (Du får frågan om filnamn.)
-  eller CTRL+A stryker ett tecken - där markören är positionerad.
-  eller CTRL+E gör det möjligt att lägga till tecken före markörens position

Nyckelord

| | |
|-------------------------------------|------------|
| ACCESS-ANVÄNDARE | 17. |
| Användardefinition | 17, 61. |
| Användare | 10. |
| Användarnamn | 17. |
| Användarområde | 11. |
| Arbetsområde | 10, 17. |
| Batch-prioritet | 74. |
| Batch-process | 3, 69. |
| Data dictionary | 3. |
| Databas | 3. |
| beskrivning | 9. |
| strykning | 25. |
| tillgänglig | 25. |
| Databas-definition | 23, 62. |
| Dataelement definition | 19. |
| Dataelement definition | 30. |
| DBA | 3, 9, 12. |
| DBA-ANVÄNDARE | 18. |
| DBA användar-definition | 18. |
| DDI | |
| filer | 3, 13. |
| process | 3. |
| start | 9. |
| stopp | 9. |
| DDI-process | 69. |
| Definiering DBA-användare | 18. |
| Device internal | 11. |
| DUMP-HELP-FIL | 54. |
| Felmeddelanden | 57. |
| Fil | 33. |
| Fil, | |
| FLAT | 30, 32. |
| ISAM | 3. |
| Filer kommando | 53. |
| Filer, | |
| FLAT | 3. |
| ISAM | 30, 33. |
| FLAT-fil | 3, 30, 32. |
| Fleranvändarbibliotek | 55. |
| Förkorta | 12. |
| Hjälpinformation | 53. |
| Internal device | 11. |
| ISAM | |
| fil | 3. |
| filer | 30, 33. |
| ISAM-FILER | 33. |
| Kolumnbredd | 32. |
| Kommando | 12. |
| Kommandofiler | 53. |
| Kommunikationsproblemm | 55. |
| Kontrollkommandon | 79. |
| KOPIERA-DATABASE | 34. |

| | |
|---|---------|
| KOPIERA-REALM | 34. |
| LADDA-DATA-FIL | 45. |
| LADDA-DATABAS | 45. |
| LADDA-TEXT-FIL | 45. |
| Lagringsformat | 19. |
| Lösenord | 17. |
| Nivå | |
| start | 12, 17. |
| tabell definition | 12. |
| Nivå, tabell definition | 29. |
| Oberoende databas beskrivning | 11. |
| Odefinierade-datatyper | 23. |
| OPTIMERA-TABELL | 45. |
| Prioritet | 74. |
| Process, | |
| batch | 3, 69. |
| DDI | 3, 69. |
| Realms | 3. |
| REDIGERA-ANVÄNDARE | 29. |
| REDIGERA-DATABAS | 26. |
| REDIGERA-DATATYPER | 19, 30. |
| Resultatlängd | 32. |
| SIBAS realms | 3. |
| SIBAS-ITEMS | 35. |
| SIBAS-REALMS | 35. |
| SIBAS-tabeller | 34. |
| SIBAS realm | 30. |
| SINTRAN | 9. |
| Skapa dataelement | 19. |
| SKAPA-ANVÄNDARE | 18. |
| SKAPA-DATABAS | 24. |
| SKAPA-DATABASER | 25. |
| SKAPA-TABELL | 37. |
| SLUT | 26, 37. |
| Start level | 12. |
| Startnivå | 17. |
| STRYK-DATABAS | 25. |
| STRYK-TABELL | 36. |
| Systemansvarig | 73. |
| Söknyckel | 31. |
| Söktid | 56. |
| Tabell | 33. |
| Tabelldefinitionsnivå | 12, 29. |
| Tid sök | 56. |
| Tillgänglig databas | 25. |
| Transaktionsbibliotek | 55. |
| VAR-ANVÄNDS-DATATYP | 23. |
| VISA-DATABASER | 25. |
| VISA-REALMS | 34. |
| VISA-TABELLER | 36. |
| XON/XOFF | 73. |
| Ägare | 24. |

***** **SEND US YOUR COMMENTS!!!** *****



Are you frustrated because of unclear information in this manual? Do you have trouble finding things? Why don't you join the Reader's Club and send us a note? You will receive a membership card — and an answer to your comments.

Please let us know if you

- * find errors
- * cannot understand information
- * cannot find information
- * find needless information

Do you think we could improve the manual by rearranging the contents? You could also tell us if you like the manual!



***** **HELP YOURSELF BY HELPING US!!** *****

Manual name: ACCESS DBA Användarhandbok

Manual number: ND-30.020.3 SW

What problems do you have? (use extra pages if needed) _____

Do you have suggestions for improving this manual ? _____

Your name: _____ Date: _____

Company: _____ Position: _____

Address: _____

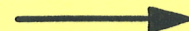
What are you using this manual for ? _____

NOTE!

This form is primarily for documentation errors. Software and system errors should be reported on Customer System Reports.

Send to:

Norsk Data A.S
Documentation Department
P.O. Box 25, Bogerud
0621 Oslo 6, Norway

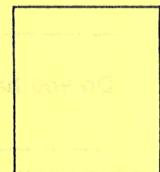


Norsk Data's answer will be found on reverse side

Answer from Norsk Data: _____

Answered by: _____

Date: _____



Norsk Data A.S
Documentation Department
P.O. Box 25 BOGERUD
N - 0621 OSLO 6 - Norway

Systems that put people first

NORSK DATA A.S OLAF HELSETS VEI 5 P.O. BOX 25 BOGERUD 0621 OSLO 6 NORWAY
TEL.: 02 - 29 54 00 - TELEX: 18284 NDN