# ACCESS DBA Handbok ND-30.022.4 SW

# ACCESS DBA Handbok ND-30.022.4 SW

January 1982 Version 1 (English) Preliminary Issue January 1983 Version 2 (English) August 1984 Version 3 (English) January 1987 Version 4

The information in this manual is subject to change without notice. Norsk Data A.S assumes no responsibility for any errors that may appear in this manual. Norsk Data A.S assumes no responsibility for the use or reliability of its software on equipment that is not furnished or supported by Norsk Data A.S. Copyright © 1985 by Norsk Data A.S.

Förord:

#### PRODUKTEN

Denna manual beskriver Data Dictionary-DBA modulen i fråge- och transaktionssystemet ACCESS, version E.

Produkten har registrerats med ND-numren:

ND210185E för ND-100 ND210529E för ND-500

### 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

- ACCESS kan nu använda användarnamn som tas från User Environment eller SINTRAN för att kontrollera om Du har rätta att köra ACCESS (se sidan 17). Du får inte längre frågan om användarnamn och lösenord när Du startar ACCESS.
- Fjärrfiler kan nu användas.
- Felmeddelanden från SINTRAN, SIBAS och BCD (Binary Coded Decimal) skrivs ut som text (tagna från UE-fel biblioteket) och är inte bara ett nummer (se sidan 62 och 65).

- Namnen på kommandofilerna har ändrats från ACCESS-HELP-XX:TEXT till ACCESS-XX-E:HELP och från DBA-HELP-XX:TEXT till DBA-XX-E:HELP (där xx är språksuffixet). Namnet på ACCESS INIT filen har ändrats till ACCESS-xx-E:INIT.
- ACCESS hämtar användarens språkkod från UE-användarprofil och läser korrekt språkversion av filen ACCESS-xx-E:HELP eller ACCESS-XX-E:INIT (se sidan 61).
- Om Du är inloggad via User Environment, kontrollerar ACCESS om användaren har rätt att göra SINTRAN-kommandon.
- En ny rad har lagts till i ACCESS-XX-E:HELP filen som innehåller språkkoden (se sidan 61 och 62).
- ACCESS använder nu SIBAS HEADINGS för ACCESS-tabellens namn och ACCESS itemnamnen vid kopiering från SIBAS databeskrivning. Alla datatyper kopieras också (se sidan 37).

#### Följande kommandon har ändrats:

Gammalt kommandonamn	Ändrat
*SKAPA-TABELLER (från ACCESS)	Ändrat till REDIGERA-TABELL
*ACCESS-ANVÄNDARE	Funktionen är ändrad (se sidan 17)
**KOPIERA-REALM och **KOPIERA-DATABAS	Kopierar nu SIBAS HEADINGS och alla datatyper (se sidan 37)

Alla större ändringar har markerats med ett vertikalt streck i marginalen. Handboken innehåller en bilaga som förklarar speciella felsituationer. Se också förordet för ACCESS användarhandbok.

ANDRA AKTUELLA HANDBÖCKER

ACCESS Användarhandbok	ND-60,153
SINTRAN III Reference Manual	ND-60.128
SINTRAN III Timesharing/Batch Guide	ND-60.132
ISAM User Guide	ND-60.108
SIBAS II User Manual	ND-60.127
SIBAS-II Operators's Manual	ND-30.009

# INNEHÅLLSFÖRTECKNING

0 1		
SOL	t 1 On	
UCA	01011	

1 A	CCESS OCH DBA-PROGRAMMET
1.1 1.2	Introduktion
2 D	BA I SINTRAN
2.1 2.2 2.2.1 2.2.2 2.2.3 2.2.4 2.3 2.4 2.5 2.6 2.7	DDI filer77DDI processen77DDI och SINTRAN78Att använda flera än en DDI process78Starta DDI processen79Stoppa DDI processen70Starta DBA-programmet100Start av ACCESS11Användarkonceptet i systemet11Att ge flera användare tillgång till samma databas12Kommandon13
3 S	TARTNIVÅN
3.1 3.1.1 3.2 3.3 3.3.1 3.3.2 3.3.3 3.4 3.5 3.6 3.6.1 3.6.2 3.7 3.8 3.9 3.10	Hur Du definierar ACCESS-användare17Från DBA17Från ACCESS18Hur Du definierar DBA-användare19Hur Du skapar dataelementen19Hur Du specificerar lagringsformatet20Motsvarighet med COBOL data-beskrivning23Reserverade datatypsnamn23Hur Du tar reda på var en datatyp används24Odefinierade-datatyper24Hur Du definierar en ny databas25Från DBA25Från ACCESS26Hur Du kan få en lista på tillgängliga databaser26Hur Du stryker en databas-beskrivning27Avslutning av DBA startnivå27
4 тл	ABELLDEFINITIONSNIVÅN
4.1 4.1.1 4.2	Hur Du specificerar användares tillgång till databaser 31 Hur Du specificerar behörighet från ACCESS 32 Hur Du definierar dataelement i tabeller 32

<ul> <li>4.3 Hur Du definierar och redigerar tabeller .</li> <li>4.3.1 Söknycklar i ISAM-tabeller</li> <li>4.3.2 Kolumnbredd och resultatlängd</li> <li>4.3.3 Användning av flat filer</li> <li>4.4 Tabeller och filer i ACCESS</li> <li>4.5 Hur Du definierar SIBAS-tabeller</li></ul>			      	aba		· · · · · · · · · · · · ·	• • • • • • • • • • •	32 334 35 36 30 40 44 44 47 49
6 ALLMÄNT OM KÖRNING AV ACCESS-SYSTEMET		•	•	•••	•	•	٠	57
<ul> <li>6.1 Kommandofiler</li></ul>	      		• • • • • • • • • • • • • • • • • • •	     		· · · · · · · · · · · · · · · · · · ·	•••••••••	59 60 61 62 63 63 63 64 64 65
<ul> <li>7.1 Exempel på hur Du definierar användare .</li> <li>7.2 Exempel på Hur Du definierar en databas</li> <li><u>BILAGA</u></li> </ul>	•••	•	•	· ·	•	•		69 70
A DDI-PROCESSEN SOM EN BATCH-PROCESS .		•	•	•••	•	•		73
B INFORMATION FÖR SYSTEMANSVARIG		•	•		•	•	•	77
C SPECIELLA FELSITUATIONER		•	•	• •	•	•	•	83
D KONTROLLKOMMANDON I DBA MODULEN	• •	•	•		•	•	•	89
Nyckelord								93

ACCESS DBA HANDBOK

# KAPITEL 1

\$ 1

# ACCESS OCH DBA-PROGRAMMET

Norsk Data ND-30.022.4 SW

XAPITEL .

ACCESS OCH DIA-PROGRAMMET

.

Norsk Data ND-30.022.4 SW

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 7. Dessa filer finns normalt under användare DATA-DICTIONARY, men kan också finnas under en annan användare, se sida 8. 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. 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. Processen måste vara aktiv när ACCESS körs. Den ger ACCESS en fullständig beskrivning av datastrukturen systemet arbetar med.

Vi behöver också ett hjälpmedel för att skapa systemet för användaren, dvs skapa nya beskrivningar och ändra gamla. Detta är <u>DBA, D</u>ata <u>B</u>as <u>A</u>dministratören, 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 ge ACCESS den information som behövs under det dagliga arbetet. DBAprogrammet 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 <u>hur</u> 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 flat 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 eller samma användare som DDI-processen körs under.



Fig. 1. En översikt över ACCESS och DBA

Norsk Data ND-30.022.4 SW

•

# KAPITEL 2

# DBA I SINTRAN

Norsk Data ND-30.022.4 SW

.

2 DBA I SINTRAN

### 2.1 DDI filer

ACCESS behöver bland annat information om datastrukturen, användaren och rättigheter. Detta kallas Data Dictionary Information och har lagrats på 14 SINTRAN-filer. Dessa är följande:

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)

Vanligen finns dessa under SINTRAN-användaren DATA-DICTIONARY.

#### 2.2 DDI processen

DDI-processen använder DDI-filerna för att förse ACCESS med beskrivningar på datastrukturen, Data Dictionary. När ACCESS ska användas måste denna process vara aktiv. Du kan inte starta ACCESS om DDI processen är passiv.

På samma sätt så kan inte DBA modulen som används för att ändra Data Dictionary öppna DDI-filerna om DDI processen är aktiv. Innan du startar DBA så måste Du stoppa DDI processen.

Kommunikationen mellan DDI processen och ACCESS går via två internal device i SINTRAN. Enheterna är reserverade från en batch process som startas genom att använda två filer:

ACCESS-DDI:BATC som batch indatafil ACCESS-DDI:LOG som batch utdatafil

För mer information om batch-processen se bilaga A.

### 2.2.1 DDI och SINTRAN

Innehållet i batch indatafilen (ACCESS-DDI:BATC) är viktigt av två orsaker:

- Första raden bestämmer från vilken SINTRAN-användare DDI processen skall köras. DDI filerna <u>måste</u> finnas under samma användare.
- De tredje och fjärde raderna bestämmer vilka internal device som DDI processen skall använda.

Var klar över att om innehållet i filen ACCESS-DDI:BATC är fel så kan Du starta fel DDI process från fel SINTRAN-användare som arbetar mot fel DDI filer och kommunicerar med fel enhet.

# 2.2.2 Att använda flera än en DDI process

Om Du vill använda en oberoende Data Dictionary så kan du använda en annan uppsättning av DDI filer under en annan SINTRAN-användare.

En oberoende databasbeskrivning tillåter dig att utveckla databasen utan att det dagliga arbetet berörs. Det gör det också möjligt för nybörjare att testa och experimentera utan att röra den "riktiga" körningen av programmet under DATA-DICTIONARY. Den kan också användas för att fysiskt separera två olika utseenden av samma data.

För att få en oberoende databasbeskrivning så måste Du göra följande innan Du startar DDI processen. Kopiera filerna:

(DATA-DICTIONARY)ACCESS-DDI:BATC (DATA-DICTIONARY)ACC-DDI-STOP:MODE

från användare DATA-DICTIONARY till den användare där Du vill att DDI filerna skall finnas. Ersätt namnet DATA-DICTIONARY på <u>första</u> raden på ACCESS-DDI:BATC med namnet på den SINTRAN-användare Du är inloggad på när Du skall köra DBA-programmet. Ändra också numren på de internal device som skall användas av processen i filerna ACCESS-DDI:BATC och ACC-DDI-STOP:MODE.

Du skall nu starta två DDI processer på två olika batch processorer. De två processerna måste använda olika internal device, och varje måste ha en egen kopia av indata och utdatafilerna i APPEND-BATCHkommandot som beskrivs i sektion 2.2.3. (Jämför denna med beskrivningen på hur ACCESS laddas i SINTRAN från en diskett). Varje kopia på indatafilen har en egen SINTRAN-användare på första raden.

Om Du vill använda två olika DDI processer så kan situationen illustreras enligt nedan.



För information om filerna ACCESS-SW-E:HELP och ACCESS-SW-E:INIT se sidan 59.

### 2.2.3 Starta DDI processen

För att se om DDI processen är aktiv eller passiv ge SINTRANkommandot:

#### @LIST-BATCH-PROCESS

Om användare DATA-DICTIONARY är inloggad så är DDI processen aktiv. Om inte så startar Du DDI processen genom att ge kommandot:

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

De två sista parametrarna är indata och utdatafiler 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-SW-E:HELP med kommandot beskrivet ovan, men <u>utan</u> "@". (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.

ACCESS programmet är beroende av svar från DDI processen för att fungera så DDI <u>måste</u> vara aktiv när ACCESS används.

Ytterligare information om batch-systemet finns i SINTRAN III Timesharing/Batch Guide, ND-60.132. Se också bilaga A.

# 2.2.4 Stoppa DDI processen

Systemet innehåller en mode-fil som stoppar DDI-processen. Filen heter ACCESS-DDI-STOP:MODE. Den innehåller numren för de internal device som används av processen. Dessa nummer <u>måste</u> motsvara numren som givits i batch indatafilen ACCESS-DDI:BATC.

Ge kommandot:

#### @MODE ACCESS-DDI-STOP:MODE

från SINTRAN-användaren som har den aktuella DDI filen för processen (och från den användare där DDI processen körs).

OBS!

Innan Du stoppar DDI processen så måste Du vara säker på att ingen använder ACCESS. Om Du stoppar processen när någon kör ACCESS så kommer ACCESS att stoppas när den försöker kommunicera med DDI.

För att se om någon använder ACCESS ge SINTRAN-kommandot:

#### @TERMINAL-STATISTICS

Se om Du kan se en förkortning "ACCESS".

### 2.3 Starta DBA-programmet

DBA-programmet används för att redigera informationen i Data Dictionary-filerna. Programmet använder alltid DDI filerna på aktuell SINTRAN-användare.

För att starta DBA, logga in som användare DATA-DICTIONARY där beskrivningen på databasen finns. Om Du vill redigera DDI filerna någon annanstans logga då in som den användare där de aktuella DDI filerna finns. DBA programmet påverkar bara DDI-filern som finns under den SINTRAN-användaren där Du startar DBA. Om DBA inte hittar DDI filerna så skapas de automatiskt. Eftersom DDI processen reserverar DDI filerna så måste den stoppas för att frigöra filerna när DBA-programmet skall använda dem.

Du är nu klar att gå in i DBA-programmet. Ge kommandot:

@DBA⊷

DBA kommer fråga efter användaridentifikation för att försäkra sig om att Du har rättighet att köra DBA-programmet och ändra Data Dictionary informationen. Bara DBA-användare har rätt att starta DBA. Se sektion 2.5.

Om Du startar med en tom DDI fil (t ex om Du vill skapa en oberoende databas som beskrivits på sidan 8) kommer DBA inte att fråga om din identifikation den första gången. Du kommer direkt in i användarbeskrivningen där än så länge den enda användaren definierats är den Du loggade in som. Ändra denna användare till önskat användarnamn och ge arbetsområde och lösenord (se sidan 17). Det nya användarnamnet kommer automatiskt att registreras som DBA-användare.

När Du lämnat DBA-programmet så kan ACCESS köras igen. För uppstart av en ny ACCESS-session så måste DDI procesesn vara aktiv (se section 2.2.3).

### 2.4 Start av ACCESS

ACCESS använder filerna ACCESS-SW-E:HELP (eller ACCESS-SW-E:INIT) för att veta var den skall hitta Data Dictionary informationen. ACCESS läser raderna 3 och 4 i denna fil för att hitta internal device som skall användas vid kommunicering med DDI processen.

Det kan finnas olika språkversioner på filerna och de kan finnas på olika SINTRAN-användare. Se sidan 59 för detaljerad information om hur ACCESS skapar dessa filer.

### 2.5 Användarkonceptet i systemet

ACCESS transaktionssystemet har två kategorier av användare:

- 1. Användare som har rätt att köra ACCESS
- 2. Användare som har rätt att köra ACCESS och DBA

När Du försöker att gå in i ACCESS så kontrollerar systemet att Du definierats som ACCESS användare. Detta görs genom att jämföra ditt UE användarnamn (eller om Du inte använder UE (User Environment) ditt SINTRAN-användarnamn) med definitionerna på ACCESS-användare. När Du startar DBA-programmet så får Du frågan om användarnamn. Detta görs för att skydda DBA-modulen. Du får köra DBA bara om Du har definierats som DBA-användare. Standard DBA-användarnamn är namnet Du använder i UE (eller SINTRAN-användarnamnet). Om Du har ett lösenord så frågar systemet efter det också.

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

Endast <u>en</u> ACCESS användare kan använda samma arbetsområde (filer) samtidigt, men <u>flera</u> personer (terminaler) kan vara inloggade i SINTRAN under samma SINTRAN-användarnamn. Förhållandet mellan SINTRANanvändare och arbetsområden förklaras på sidan 17.

### 2.6 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. SINTRAN-användare har av säkerhetsskäl inte rättighet att skriva på andra användares filer. Av samma skäl har en användare av ACCESS normalt inte rättighet 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:



Skriva

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.

Detta problem uppstår inte om Du använder SIBAS-databas.

2.7 Kommandon

DBA programmet har två nivåer. Nivå 1 kallas för Startnivå och nivå två för Tabelldefinitionsnivå. Kommandon ges i kommandopositionen som är utmärkt med \* eller \*\* nederst på skärmen.

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:



# KAPITEL 3

# STARTNIVÅN

Norsk Data ND-30.022.4 SW

Norsk Data ND-30.022.4 SW

### 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. Eftersom ACCESS inte frågar efter användarnamnet vid uppstart så måste namnet Du ger här för vanlig ACCESS-användare motsvara UE-användarnamnet (eller SINTRANanvändarnamnet) i ditt system. För DBA användare så kan Du ge andra namn men det rekommenderar vi inte eftersom denna användare då inte tillåts att använda ACCESS även om han/hon definierats som ACCESSanvändare. Användare kan strykas med tangenterna SKIFT + STRYK på en NOTIS-terminal (eller CTRL+D CTRL+L).

Om Du startar med en tom DDI-fil så kommer listan att innehålla ditt eget UE-användarnamn eller SINTRAN-användarnamn.

#### Lösenord

För att få ett säkert DBA-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. Vanliga ACCESS-användare har inget lösenord. En DBA-användare är inte tillåten att se eller ändra en annan DBA-användares lösenord. Bara ditt eget lösenord visas i denna kolumn.

#### 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 en användare specificerats som "WORKAREA" kommer ACCESS att skapa följande fem filer:

WORKAREA:DATA WORKAREA:SORT WORKAREA:SYS1 WORKAREA:SYS2 WORKAREA:SYS3

Du kan också specificera under vilken SINTRAN-användare Du vill att arbetsområdet skall finnas t ex (ANVÄNDARE-1)WORKAREA i denna kolumn Obs, endast en terminal kan använda arbetsområdet samtidigt. Om Du vill ha flera användare som kör ACCESS samtidigt så måste Du ge individuella arbetsområden till alla användare.

### 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 tilllåts 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

I kolumnen skrivs de ACCESS-användares namn in som skall bli DBAanvändare. Tryck på HJÄLP-tangenten för att få en lista på alla definierade användare.

Observera att Du inte kan redigera/stryka information om andra DBAanvändare, 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 DBAprogrammet.

<u>Alla</u> dataelement har en typ. Denna beskriver hur dataelementen skall lagras, men den begränsar också användandet av elementen. Dataelement av samma typ kan representera varandra, "byta plats".

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". Utskriftformat, 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.

Γ

#### ACCESS DBA HANDBOK STARTNIVÅN

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: -32768 till 32767)	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	n + m, utom SIGNED			
SIGNED † UNSIGNED	siffror	ger n + m + 1			
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-kommandona (sidan 40) för mer detaljerad beskrivning av dessa typer (i ACCESS kallade CHARACTER och NUMERIC).

Norsk Data ND-30.022.4 SW

Om datatypen skall används i SIBAS item-definition <u>måste</u> 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 <u>ord</u>. 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 32 bit CPU	COMP-2 PIC S9(n)V9(m) där n+m <= 6		
REAL6	COMP-2 på 48 bit CPU	COMP-2 PIC S9(n)V9(m)		
REAL8		där n + m >= 7		
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 ACCESSs 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:

### \*<u>VAR-ANVÄNDS-DATATYP</u> <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 ("MIN-TABBELL"), kommer detta att visas. Du får inte använda en tabell som innehåller odefinierade datatyper, eftersom resultatet kommer att bli felaktigt.

24

# 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 flat 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. Databas-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 som 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 tabelldefinitionsnivån.

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.

<u>SIBAS namn</u> 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.

<u>SIBAS nr</u> är numret på den SIBAS-process som kontrollerar SIBASdatabasen.

SIBAS lösenord är lösenordet på databasnivå i SIBAS-databasen.

 $\underline{\rm Skapad}$  av är den DBA-användare som skapat ACCESS-databasen. Med andra ord, skriv ditt DBA-användarnamn.

### 3.6.2 Från ACCESS

Kommandot SKAPA-DATABASER (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:

#### \*<u>VISA-DATABASER</u>⊷J

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

# SLUT

eller ge kommandot:

#### \*SLUT

ACCESS DBA HANDBOK

ACCESS DBA HANDBOK

# KAPITEL 4

### TABELLDEFINITIONSNIVÅN
### LAPITEL A

#### TABELLDRFIELTIONSNIVAH

Norsk Data ND-30.022.4 SW

### 4 TABELLDEFINITIONSNIVÅN

Tabelldefinitionsnivån (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 tilllåter dig inte att se de tidigare definierade rättigheterna.

Kommandot ges enligt:

#### \*REDIGERA-ANVÄNDARE

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

Du ger behörighet genom att svara JA eller NEJ, på följande frågor:

- Utskrift tillåtet.
- Insättning av nya poster är tillåtet.
- Uppdatering av poster är tillåtet.
- 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>			
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 p<="" th=""><th>oå tabellen≻</th><th></th><th></th><th></th></namn>	oå tabellen≻			
Νι	ummer	Söknyckel	Kc	lumnbredd	Radlängd

# 4.3.1 Söknycklar i ISAM-tabeller

Söknycklar i ISAM-tabeller kan vara endera PRIMÄR eller SEKUNDÄRA. För flat 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.

Å andra sidan tar nycklar plats. När en kolumn definierats som en nyckel kommer hela kolumnen att lagras två gånger: en gång i :DATAfilen 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 bara kolumner som nycklar när Du tror att Du kommer att söka i dem ofta. 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 är det inte att rekommendera att definiera det som en primärnyckel, eftersom det kan finns 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 initiala 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 längden 80 tecken och resultatlängden 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 flest rader.

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

# 4.3.3 Användning av flat 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 omdefineras 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-J

Följande bild visas på din skärm (som exempel, har vi förutsatt att Du skapat en tabell kallad "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 kolumen "SINTRAN filnamn" med filtyperna :ISAM och :DATA. I ovanstående exempel, skulle Du behöva filerna PERSONAL:DATA och PERSONAL:ISAM under SINTRANanvändare DATABAS.

För flat 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 SINTRANanvändare 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.

À andra sidan, om Du bara vill ha 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.

SIBAS-databasen (och systemnummer) som skall användas specificeras när ACCESS databasen skapas (se sida 26).

SIBAS-tabeller kan definieras <u>manuellt</u>, på samma sätt som ISAM- och flat tabeller; eller också kan de definieras <u>automatiskt</u>, 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

#### ACCESS DBA HANDBOK TABELLDEFINITIONSNIVÅN

- \*\*KOPIERA-REALM Kopierar en SIBAS realmbeskrivningar till en ACCESS <realm namn> tabell-beskrivning
- \*\*KOPIERA-DATABAS Kopierar alla SIBAS realm beskrivningar till ACCESS tabell-beskrivning

Kommandona KOPIERA-REALM och KOPIERA-DATABAS kopierar alla items, groups-items och sets från SIBAS beskrivningen till ACCESS-beskrivningen.

När dessa kommandon används observera följande:

- SIBAS group items definieras inte som ACCESS data element. Vill Du det måste alla items i gruppen vara av type CHARACTER och Du måste göra det manuellt. ACCESS använder de kopierade grupp itemen vid kontakt med databasen. Grupperna som kopierats kan inspekteras och/eller ändra med kommandot \*\*REDIGERA-GRUPPER.
- SIBAS grupperna som innehåller mer än 6 dataelement kan inte användas av ACCESS som nyckel och de kopieras inte.
- DBA kopierar SIBAS HEADINGS för databas itemen och använder dem som ACCESS data elementnamn. Om inte HEADING har definierats kommer DBA att använda SIBAS itemnamn för ACCESS data elementnamnen. För att ändra detta ge kommandot \*\*SIBAS-ITEMS.
- DBA kopierar SIBAS HEADINGS för REALMS och använder dem som ACCESS tabellnamn. Om ingen HEADING definierats kommer DBA att använda SIBAS REALMnamn som ACCESS tabellnamn. För att ändra detta ge kommandot \*\*SIBAS-REALMS.
- ACCESS data elementnamnen och tabellnamnen kan ha en maximal längd på 32 tecken. Om motsvarande SIBAS HEADING innehåller mer än 32 tecken så används bara de 32 första.
- I SIBAS är det möjligt att ge samma HEADING till fler REALMS och flera ITEMS i en REALM. Detta är inte tillåtet i ACCESS. Om otydlighet uppstår så kommer ITEM namnet eller REALM namnet att användas istället för motsvarande HEADING. Du får alltid ett meddelanden när det händer. Använd kommmandot \*\*SIBAS-ITEMS för att inspektera eller ändra ACCESS data elementnamn. Använd kommandot \*\*SIBAS-REALMS för att inspektera eller ändra ACCESS tabell namn.
- SIBAS multimember sets kan inte användas av ACCESS och kopieras inte. Ett meddelande visas om ACCESS hittar ett sådant set. Använd kommandot \*\*REDIGERA-SETS för att inspektera eller ändra sets.
- ACCESS genererar standard datatypsnamn för motsvarande med SIBAS lagringskod. Använd kommandot \*REDIGERA-DATATYPER för att inspektera eller ändra datatyperna.

Kommandot SIBAS-ITEMS kommer att ge följande bild på skärmen:

PERSONAL		
Dataelementnamn		Item-namn
NAMN CHEF AVD		NAMN CHEF AVD

Vänstra kolumen innehåller HEADINGS som kopierats från SIBAS. Detta blir de dataelementnamn som ACCESS-användaren ser på skärmen. Dessa kan redigeras; (Redigeringen påverkar inte SIBAS databasen) t ex, "AVD" kan ändras till "AVDELNING".

Om Du vill ändra ordningen på dataelementen i ACCESS tabellkolumnen ge kommmandot \*\*REDIGERA-TABELL. Se sidan 33.

För att ändra tabelnamnen, ge kommandot SIBAS-REALMS:

SIBAS realms	
Tabellnamn	SIBAS realm
Personal Avdelning	EMPLOY DEPT

Den vänstra kolumnen innehåller realm HEADINGS kopierade från SIBAS-DRL-beskrivningen. De blir tabellnamn använda av ACCESS men kan redigeras. Det Du redigerar påverkar inte SIBAS databasen.

#### Manuell tabelldefinition

Kommandona SKAPA-TABELL, REDIGERA-TABELL och VISA-TABELL används på samma sätt som när Du definierar ISAM och flat-tabeller (se sidan 33).

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

#### ACCESS DBA HANDBOK TABELLDEFINITIONSNIVÅN

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		
Dataelementna	amn	Item-namn
Namn Chef Lön Avdelning		Namn Chef Lön Avdelning

Här har båda kolumnerna dataelementnamnen definierade med kommandot REDIGERA-TABELL, och därför måste vi ändra den <u>högra</u> 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.)

För information om SIBAS se handboken: SIBAS II User Manual ND-60.127.

4.6 Hur Du stryker en tabell-beskrivning

Ge kommandot:

#### \*\*STRYK-TABELL

med tabellnamnet som parameter, så stryks tabellbeskrivningen. Filen som denna ingår i 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 - tabelldefinitionsnivån - och gå tillbaka till startnivån, tryck på

# SLUT

eller ge kommmandot:

\*\*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 existerande tabell <u>eller</u> 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 flat 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.
- Datatyp Det är möjligt att använda de datatyper som definierats med kommandot REDIGERA-TYPER i DBAprogrammet. ACCESS söker först bland dessa datatyper när Du skriver i en kolumn.

Norsk Data ND-30.022.4 SW

ACCESS accepterar också några datatyper som automatiskt definieras i Data Dictionary när Du anger dem här. Dessa beskrivs nedan. Om Du redan definierat något av de här datatyperna i DBAprogrammet kommer ACCESS att använda dina datadefinitioner - inte de som beskrivs nedan.

Om Du använder en databas som redan finns (t ex en SIBAS databas) så behöver Du de datatyper som används i databasen. Men om du skapar en ny databas för ACCESS så är det vanligtvis tillräckligt med följande datatyper CHARACTER, NUMERIC och FILLER.

Nyckeltyp Möjliga nyckeltyper är PRIMÄR, SEKUNDÄR eller ingen. Se sidan 33 för mer information.

Nycklar används inte tillsammans med flat filer.

#### 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 jämställas 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:

NAMN	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 fördefinerade 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 21.

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 + STRYK

Stryker aktuellt element.

#### ACCESS DBA HANDBOK TABELLDEFINITIONSNIVÅN

CTRL+	
$\square$	. [7]

Lägger till ett element före aktuellt element.



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 användaren som äger databasen för att göra detta):

### \*REDIGERA-TABELL DATABAS PERSONAL

ACCESS svarar med:

Skriv det nya tabellnamnet:	Det gamla namnet visas.
	Om Du ändrar 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. Om filen inte finns så skapas den (använd inte citationstecken).

#### ACCESS DBA HANDBOK TABELLDEFINITIONSNIVÅN

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

TABELL	DEFINITION	
Dataelement Kundnamn Adress	Datatyp character-30 character-40	Nyckeltyp primär
Postadress Telefonnummer	Postadress telefon	sekundär sekundär
	•••••••••••••••••••••••••••••••••••••••	•••••
	· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • •
••••••	•••••••••••••••••••••••••••••••••••••••	•••••
·····	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • •
	••••••••••••	•••••
	· · · · · · · · · · · · · · · · · · ·	•••••
1		

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. Låt oss t ex säga att två användare ANVÄNDARE-A och ANVÄNDARE-B skall arbeta mot olika delar på samma SIBAS databas:

ANVÄNDARE-A

ANVÄNDARE-B

Avdelning Register Konton Personal - bara avdelning och namn

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 <u>Personal</u> och <u>Avdelningar</u> måste definieras i både DATABAS-A och DATABAS-B.

Tabellen <u>Personal</u> 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 42).

#### ACCESS DBA HANDBOK TABELLDEFINITIONSNIVÅN

# 4.11 Storlek och begränsningar för ACCESS-tabeller

ACCESS har följande maximi begränsning för definitionen på tabellerna och dataelementen.

MAX. RECORD LÄNGD	:	I SIBAS : 1000 BYTES annars : 2 K BYTES
MAX. ANTAL DATAELEMENT I ETT RECORD	:	128
MAX. DATAELEMENT LÄNGD	:	2 K BYTES (= RECORD LÄNGD)
MAX. ANTAL PÅ INDEXERADE NYCKLAR I ETT RECORD	:	ANTAL DATAELEMENT I ETT RECORD
MAX. NYCKELLÄNGD	:	MAX. DATAELEMENT LÄNGD
MAX. ANTAL DATAELEMENT I EN GROUP	:	6

#### OBS!

Du får inte alltid ett felmeddelande om Du överskrider gränserna när du definierar tabellen i DBA-programmet – det kan visa sig först när Du använder tabellen i ACCESS.

ACCESS DBA HANDBOK

# KAPITEL 5

# KOMMANDOT LADDA-DATABAS I ACCESS

Norsk Data ND-30.022.4 SW

### KAPITEL 5

#### KOMMARDOT 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 binärt format dvs man kan inte läsa det med hjälp av NOTIS-WP eller PED. 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.

Dessa kommandon kan också användas för att omdefiniera databastabellen. OBS!

- När data laddas från en textfil skapad med NOTIS-WP kom ihåg att sätta "Lagringsformat" på dokumentet till 7 (i meny 1) och sätt Ta bort ej blanka i radslut till N (meny 3) innan lagring.
- När laddning av data från en existerande ISAM-fil (:DATA delen) får det inte finnas strykna poster i filen. För att ta bort strukna poster använd kommandot \*\*OPTIMERA-TABELL.

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:

<u>Datatyp</u> 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 datatypsfältet blankt, kommer ACCESS att ge denna kolumn värdet blank eller noll i alla poster.

 $\underline{Start}$  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 resten 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 <u>start</u> i fältet efter det fält Du inte vill ha med. Systemet kontrollerar att fälten inte överlappar varandra, till exempel om du ger fel <u>start</u> och <u>längd</u>-värden. Om start- och längdvärden inte ges, kommer det värde som är definierat i DBA att tas som standardvärde.

När Du fyllt i fälten starta laddningen genom att trycka på UTFÖRtangenten.



Du kan avbryta kommandot genom att rycka på HEMMA- eller SLUT-tangenten.

För datafilerna så får Du frågan:

Skriv record längd:

Det totala standard postlängden är summan av postlängderna Du gav i föregående tabell.

Skriv filnamn:

Här måste Du ge namnet på indatafilen du laddar från. Standard filtyper är :TEXT och :DATA. Filen kan vara på en fjärrdator.

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.

- 1) Om Ja, blir databasen laddad med de registrerade data efter frågan om den gamla informationen skall strykas eller inte.
- 2) Om Nej, får Du frågan: Vill du redigera tabell-beskrivningen? J/N
  - a) Om Ja, kan Du ändra post-beskrivningen dvs datatyp, start, längd.
  - b) Om Nej, kommandot avslutas.

Om fel uppstår under laddning (t ex om Du försöker ladda poster med lika värden i söknyckeln när detta inte är tillåtet) kommer de posters nummer som orsakade felet att visas och laddas inte i tabellen.

Om ett fatalt fel uppstår så slutar laddningspoerationen och ett felmeddelande ges. Uppstår fatalt fel vid laddning till SIBAS-databas så lämnas databasen öppen!

#### 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 till 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

#### Exempel 2: Optimera-tabell

När Du använder en tabell för att stryka, uppdatera och lägga till data, blir innehållet i tabellen dåligt organiserat 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: <u>Detta kommando skall aldrig ges utan att man</u> 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:

Söknyckel	. Duplikat	
NAMN	NEJ	
AVDELNING	JA JA	

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 SIBASrealmer och flat 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.

- Använd ACCESS för att skriva ut data i orginal tabellen 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
-	<b>→</b>	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:

PE	RSONAL	NAMN	CHEF	LÖN.GRUPP	AVDELNING	ANST.DAT
$\rightarrow$ D $\rightarrow$ S $\rightarrow$ L	atatyp: Start: Jängd:	CHAR 1 32	CHARACTER 33 32		CHARACTER 65 20	

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

.

# KAPITEL 6

# ALLMÄNT OM KÖRNING AV ACCESS-SYSTEMET

Norsk Data ND-30.022.4 SW

### A FILL &

#### ALLMART ON KORNING AV ACCESS-SYSTEMET

# 6 ALLMÄNT OM KÖRNING AV ACCESS-SYSTEMET

## 6.1 Kommandofiler

Systemet använder kommandofilerna ACCESS-XX-E:HELP och DBA-XX-E:HELP, där xx är språkversion av filen. 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.

Filerna innehåller följande information:

- ACCESS/DBA version identifikation
- Språkkod
- Numren på de två internal device (endast ACCESS-xx-E:HELP)
- SINTRAN-kommando som utförs vid avslutning av ACCESS/DBA
- Namn på standard biblioteket (endast ACCESS-xx-E:HELP)
- Namn på standard trans.(endast ACCESS-xx-E:HELP)
- All text som visas på skärmen:
  - Uppstarts bilden
  - Alla andra bilder
  - HJÄLP-bilderna
  - Felmeddelanden och andra meddelanden
- Alla tillgängliga kommandon
- Alla operatorer

6.1.1 Redigera i kommandofilerna

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 HELPfilen. 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 ordet VOID.

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

STRYK % 2 Stryk en post

Användaren kan då inte använda operatorn "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-xx-E:INIT, kommer ACCESS att läsa från denna fil istället för från ACCESS-xx-E:HELP. Detta betyder att för att få den ovannämnda processen att fungera, måste Du stryka filen ACCESS-xx-E:INIT eller "dumpa" kommandofilen (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.

6.1.2 Använda fler än en kommandofil

Olika SINTRAN-användare kan ha personliga versioner av ACCESS-XX-E:HELP under sina egna arbetsområden så att de kan använda de språkversioner som de föredrar. Det är möjligt att ha alla versioner under användaren DATA-DICTIONARY. Men det är viktigt att veta hur ACCESS och DBA hittar de rätta filerna vid start. ACCESS försöker alltid att läsa HELP-filen på följande sätt:

Om Du är inloggad via User Environment:

ACCESS hämtar användar-profilens språkkod från UE-profilen ( 2bokstavskod, XX) och försöker läsa HELP-filen på följande sätt:

- 1) ACCESS-xx-E:INIT
- 2) ACCESS-xx-E:HELP
- 3) ACCESS--E:INIT
- 4) ACCESS--E:HELP
- 5) (DATA-DICTIONARY)ACCESS-xx-E:INIT
- 6) (DATA-DICTIONARY) ACCESS-xx-E:HELP
- 7) (DATA-DICTIONARY)ACCESS--E:INIT
- 8) (DATA-DICTIONARY)ACCESS--E:HELP

XX är språkkoden för HELP-filen.

Om inte inloggad via User Environment:

ACCESS försöker läsa HELP-filen på följande sätt:

- 1) ACCESS--E:INIT
- 2) ACCESS--E:HELP
- 3) (DATA-DICTIONARY)ACCESS--E:INIT
- 4) (DATA-DICTIONARY)ACCESS--E:HELP

Om Du lagrar kommandofilerna under användare SYSTEM störs denna sekvens och ACCESS hittar <u>inte</u> kommandofilerna under DATA-DICTIONARY. Därför så rekommenderar vi att Du inte har kommandofilerna på användare SYSTEM.

Det samma gäller för DBA-programmet förutom att DBA-programmet inte använder :INIT-filen.

# 6.1.3 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-xx-E:INIT. Den använder språkkoden på rad 2 på kommandofilen för att veta var den skall dumpa filen. Observera att språkkoden hämtad från UE-profilen inte används här.

:INIT-filen måste skapas först, helst som en kontinuerlig fil på 60 sidor.

ACCESS försöker dumpa till en :INIT fil på följande sätt:

1) ACCESS-xx-E:INIT (där xx är språkkoden från rad 2 i aktuell HELP-fil).

2) ACCESS--E:INIT

ACCESS dumpar på aktuellt SINTRAN-användarområde. Om Du vill dumpa versionen på en kommandofil som skall användas av flera användare så måste Du logga in som användare DATA-DICTIONARY och dumpa filen här.

ACCESS svarar med meddelande:

Kommandfilen dumpas till: ACCESS-xx-E:INIT

Observera att Du måste stryka din gamla :INIT fil för att ACCESS skall kunna läsa ACCESS-xx-E:HELP-filen vid start. Detta är viktigt för t ex Du kan ha ändrat innehållet på kommandofilen och vill dumpa en ny version.

6.1.4 Språkkoden i ACCESS-xx-E:HELP

Rad 2 i kommandofilen för ACCESS innehåller en språkkod som tjänar två syften:

- Då vet ACCESS vilket UE-Felbibliotek som skall användas. Biblioteket innehåller alla felmeddelanden som man får från olika subsystem. Biblioteket måste finnas i olika språkversioner. Om ACCESS inte hittar UE-biblioteket: UE-ERMSG-xx:ERR, så visas alla meddelanden från SINTRAN, SIBAS eller BCD med ett nummer.
- 2) Då vet ACCESS var den skall dumpa ACCESS-xx-E:HELP filen när kommandot DUMP-HELP-FILE ges. Se föregående sektion.

#### 6.1.5 Standard transaktionsbibliotek och standardtransaktion

Standard transaktionsbiblioteket är ett bibliotek som öppnas automatiskt när användaren går in i ACCESS. Namnet på filen har givits på rad 6 på filen ACCESS-xx-E: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, är 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 7 i filen ACCESS-xx-E:HELP.

# 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 skrivrättigheter.
- 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 17 hur Du definierar användare).

6.4 DDI inte aktiv

Detta meddelande visas om någon försöker köra ACCESS-huvudsystem när

- 1) DDI processen inte startats
- 2) ACCESS startats från fel SINTRAN-användare (läser fel ACCESS-xx-E:HELP fil)
- 3) Internal device number i ACCESS-xx-E:HELP inte motsvarar den som är i ACCESS-DDI:BATC
- 4) Det finns en fil ACCESS-xx-E:INIT med andra "interndevice-numret".
- 5) ACCESS läser fel språkkods version på ACCESS-xx-E:HELP med fel "interndevice-numret".

# 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-XX-E:HELP och ACCESS-DDI-E:BATC.

Numren i de två filerna måste överrensstämma!

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-E:BATC och i ACCESS-XX-E: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 O, 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.

Den generella dokumentationen på operativsystemet SINTRAN, är omfattande. Vi föreslår att Du läser SINTRAN III Time-sharing/Batch Guide (ND-60.132) som beskriver det viktigaste om filsystemet.

# 6.7 Andra felmeddelanden

Det finns tre huvudgrupper av felmeddelanden:

1) Användarfel

Användarfel följs alltid av text som beskriver felet. Dessa fel uppstår när en användare försöker göra något otillåtet eller något som inte kan förstås t ex vid ifyllandet av en tabell.

2) Systemfel

Systemfel kan uppstå när ACCESS inte hittar information som behövs för att kunna fungera eller om en vital information har förstörts. De kan också vara interna fel. Sådana felmeddelanden ser alltid ut såhär: "Fel" "System fel - -" eller "ACCESS fel: ". De följs upp av en intern felkod. Om Du inte förstår felmeddelandet så kontakta Norsk Data.

### 3) Fel som ges från andra subsystem

Systemet visar feltexter som hämtades från UE Error-biblioteket som beskriver fel som ges från SINTRAN, SIBAS, DBMS (DataBase Management System) eller BCD. Det kan hända att ACCESS inte kan hitta korrekt UE. Då visas felkoden tillsammans med information om vilket subsystem felet ges från och ger dig en möjlighet att förstå felet.

SINTRAN felkoder beskrivs i SINTRAN Reference Manual, 60.128, SIBAS fel beskrivs i SIBAS II User Manual, 60.127 och ISAM fel i ISAM Reference Manual, 60.108.

SIBAS skriver ut felkoder på en konsol (error device) om något händer. ISAM försöker alltid återvända till ACCESS med en felstatus men situationer kan uppstå när ISAM "kollapsar". Ett meddelande skrivs då ut

#### ISAM FATAL ERROR IN ROUTINE 'ZABUF'/1

och kontrollen ges till SINTRAN. I detta fall skall Du kontrollera om filen har blivit omstrukturerad, om ACCESS körs mot rätt fil och så vidare.
# KAPITEL 7

# FULLSTÄNDIGA EXEMPEL

### KAPITEL 7

### FULLSTERDIGA EXEMPS

•

## 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 2.2.4, 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".

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

#### \*ACC-ANV⊷

som är en förkortning av \*ACCESS-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 MIG ALBERT-NEUMANN	MI	(ACCESS)WORKAREA (ACCESS)WORKAREA-1 (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 definierad 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 ACCESSs databaser som är definierade av andra.

Kommando

#### \*DBA-ANVÄ

som är en förkortning av \*DBA-ANVÄNDARE, ger en tabell på DBAanvändare 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 lösenorder MI. Du kommer in på startnivån (markerad med \* i kommandopositionen). Ge kommadnot \*ACC-ANV för att ge ett lösenord. Observera att lösenordet för de andra inte visas nu och Du kan inte ändra dem nu.

#### \*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

#### ACCESS DBA HANDBOK FULLSTÄNDIGA EXEMPEL

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 kan nu definiera 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 CHEF LÖN AVDELNING	CH-32 CH-32 I-4 CH-20	1 2 3 4	

PERSONA	L		
Nummer	Söknyckel	Kolumnbredd	Radländ
1 2 3 4	P S S S		

För kolumnbredd och radlängd, låter vi systemet välja värde.

Sedan måste ISAM-filerna definieras. Ge kommandot:

#### \*\*ISAM-FILER⊷

och tabellen fylls i enligt nedan:

 ISAM Filer

 Tabellnamn

 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 ALBERT-NEUMANN	JA JA	JA JA	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

Om det har inkluderats som ett termineringskommando i filen DBA-XX-E:HELP, kommer det att utföras automatiskt varje gång Du går ut ur DBA-programmet (se kapitel 2.2.4). Detta gör att databas-definitionen är slutförd, DDI är aktiv och ACCESS kan användas som det beskrivs i ACCESS Användarhandbok. •

### BILAGA A

DDI-PROCESSEN SOM EN BATCH-PROCESS

BILAGA A

DDI-PROCESSEN SON EN EATCH-PROCESS

ACCESS DBA HANDBOK DDI-PROCESSEN SOM EN BATCH-PROCESS

ACCESS kräver information om databaser, rättigheter, etc. Denna <u>databas-beskrivning</u>, som definieras i DBA-programmet och 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 <u>program</u> som inte stannar av sig själv när det en gång har startats. Programmet finns på filen (DATA-DIC)ACCESS-DDI-E: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 DDIprocessen körs.

Informera de andra användarna på systemet att de inte får använda den batch-processor som upptas av DDI-processen. Om de gör detta kommer de att få vänta en lång stund.

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

### (DATA-DIC)ACCESS-DDI-E:BATC

som indatafil och

(DATA-DIC) ACCESS-DDI-E:LOG

som utdatafil. 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 indatafil ACCESS-DDI-D:BATC:

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

Den första raden i filen berättar för batch-processen att DDIprocessen 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>,<maxial tid för processen> Numren för "internal devices" skall skrivas in på rad 3 och 4. Dessa nummer finns också i filen ACCESS-xx-E:HELP, 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

### BILAGA 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 både för terminalen <u>och</u> SINTRAN. 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 tilldess Du har placerat markören på raden "Communication switches". Tryck på ENTER-tangenten. Då får Du följande bild på skärmen.

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

Flytta markören med piltangenterna och välj med ENTER-tangenten.

Logga in som användare SYSTEM och sök terminalnumret med <u>@WHO</u>. 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⊷ Sätt END-OF-LINE-WRAP läge För att köra ACCESS så måste terminalerna som använder systemet måste sättas i END-OF-LINE-WRAP-läge. Om detta inte är fallet så visas bara en rad av resultatet. Gå in i menyn för "Function Switches" på din terminal (förklarats ovan) och set 'End of Line Wrap' till 'Wrap'. Function Switches Time-out 0n Bell 0n Graphic Rendition Mode Attr. Communication Clock ASY Underline Representation Underline Carriage Return CR Beginning of Line Wrap Stop End of Line Wrap Wrap %Måste vara Wrap

### 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).

#### ACCESS DBA HANDBOK INFORMATION FÖR SYSTEMANSVARIG

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 2 3 4	670 672 674 676
•	•
•	•

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

 @SINTRAN-SERVICE

 \*REMOVE-FROM-TIME-SLICE 674D

 MEMORY? Y

 IMAGE? Y

 SAVE-AREA? Y

 \*EXIT

 @PRIOR BCH03 42D

 % Sätt prioriteten till1 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

 MEMORY?

 IMAGE?

 Y

 SAVE-AREA?

 Y

 \*EXIT

## BILAGA C

### SPECIELLA FELSITUATIONER



SPICIAL PRISITUATIONS

Denna bilaga är en referenssektion som förklarar några speciella felsituationer. Troligen kommer Du inte att få vara med om någon sådan så använd därför bilagan som en referens.

## NÅGONTING ÄR FEL PÅ ARBETSOMRÅDET

- Problem: Efter att ha skapat en ny ACCESS-användare och givit honom/henne ett arbetsområde så kan Du få detta meddelande när Du startar ACCESS trots att filerna på arbetsområdet inte finns under användaren. ACCESS skapar inte filerna och ger meddelandet 'Tvetydigt filnamn'.
- Orsak: Det finns filer under användare SYSTEM med namn som motsvarar filnamnen som givits i arbetsområdet. Filsystemet söker automatiskt efter filerna under SYSTEM och återvänder med felmeddelandet.
- Lösning: Ändra namnet på arbetsområdet för denna ACCESSanvändare eller stryk/byt namn på filerna under användare SYSTEM.

ACCESS systemfel – – Felkod är: 160

- Problem: ACCESS startar inte och ger detta felmeddelande.
- Orsak: "Internal device" numret som givits i ACCESS-xx-E:HELP finns inte eller ACCESS kan inte läsa dem.
- Lösning: Kontrollera filen ACCESS-xx-E:HELP. Finns de givna "device" numren? Numren måste skrivas till vänster (i kolumn 1).

SIBAS FEL: Realm not in ready mode / 881

- Problem: ACCESS ger detta meddelanden när Du kör en transaktion.
- Orsak: Ett onormalt stopp av SIBAS kan ha uppstått före detta meddelande. När ACCESS försöker sätta en "realm" till "ready" så får det meddelande 885. En realm har inte avslutats på rätt sätt utan blivit avbrutet. ACCESS försöker att slutföra och meddelandet visas.

Lösning: Återställ "error flag" i SIB-DBM. Detta gör inte databasen felfri men det är möjligt att köra transaktioner igen.

#### INGA POSTER FUNNA

- Problem: ACCESS ger ett meddelande när Du köra en transaktion med element för att slå ihop tabeller trots att Du vet att transaktionen har givit resultat.
- Orsak: Det kan vara inkonsistens i SIBAS databasen.
- Lösning: Verifiera "indexen" och "seten" i databasen med SIB-DBM.

SÖKNINGEN ÄR ALLTFÖR KOMPLEX FÖR DENNA VERSION AV ACCESS

- Problem: ACCESS ger meddelanden när Du försöker köra en transaktion.
- Orsak: Om Du använder en operator IN(....) med många selekteringskriterier och om "item" är definierad som nyckel så kan detta resultera i en "buffer overflow".
- Lösning: Ta bort nyckeln och låt ACCESS söka genom databasen sekventiellt.

#### SIBAS INTERFACE FEL: 0

- Problem: ACCESS ge meddelandet när Du försöker köra en transaktion.
- Orsak: SIBAS databasnamnet motsvarar inte SIBABs process numret.
- Lösning: Ändra SIBAS process numret för databasen eller starta rätt databas för den givna processen.

#### ACCESS DBA HANDBOK SPECIELLA FELSITUATIONER

KONSTIGA FELMMEDELANDEN VID ANVÄNDNING AV ISAM FILER

- Problem: ACCESS svarar med underliga felmeddelanden från ISAM såsom ISAM FEL när Du kör en transaktion.
- Orsak: Detta händer om Du hade ett okontrollerat stopp före detta meddelande. ISAM filerna kan innehålla fel.
- Lösning: Kontrollera att ägaren till ISAM-filerna inte är utan sidor. Kör ISAM-SERVICE programmet för att rätta filerna.

INTERNT ACCESS FEL MED FELMEDDELANDE: 18

- Problem: ACCESS startar inte och ger detta felmeddelande.
- Orsak: ACCESS förstår inte filen ACCESS-xx-E:INIT på grund av fel format eller internt fel i filen.
- Lösning: Rätta filen ACCESS-xx-E:HELP (eller använd en kopia av filen) stryk filen ACCESS-xx-E:INIT skapa en ny och dumpa :INIT filen igen.

### ACCESS VILL INTE STARTA

Problem: ACCESS vill inte starta och "hänger" utan meddelande.

Orsak: Kommunikationsproblem med DDI processen.

Lösning: Stoppa DDI-processen och starta den igen.

.

### BILAGA D

#### KONTROLLKOMMANDON I DBA MODULEN

NO L'ANNOLAT AL MARKAN AND AND

Norsk Data ND-30.022.4 SW

89

.

BILAGA

90

•

RONTROLLEROMMENTON I DOA MODULER

### ACCESS DBA HANDBOK KONTROLLKOMMANDON I DBA MODULEN

STRYK	eller CTRL+D CTRL+D	stryker fältet där markören är positionerad.
SKIFT	+ (STRYK) eller CTRL+D C	TRL+L stryker hela raden där markören är positionerad
(qJALH)	eller CTRL+G	visar tillgängliga kommandon i aktuell situation
SKRIV	eller CTRL+O	skriver bilden på skärmen till en fil. (Du får frågan om filnamn).
<b>a</b>	eller CTRL+A	stryker ett tecken – där markören är positionerad.
	eller CTRL+E	gör det möjligt att lägga till tecken före markörens position.

ACCESS DBA HANDBOK KONTROLLKOMMANDON I DBA MODULEN ·

## Nyckelord

ACCESS start	
ACCESS-ANVÄNDARE	
ACCESS-DDI-STOPMODE	
Användardefinition	
Användarnamn	•
Arbetsområde	
Batch process	•
Batch process $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $3$ .	
$Batch-prioritet \dots \dots$	
Batch-process $\ldots$ $7, 75$ .	
Data dictionary $\ldots$ $3$ .	
Databas	
beskrivning	
strykning	
tillgänglig	
Databas-definition	1
Dataelement definition	•
DBA	
DBA-ANVÄNDARF	
DBA användar-definition	
filer	
och SINTRAN	
process	
start	
stopp	
DDT-filer	
Definition $PPA$ and $r$	
Dimpulsion provides $19$ .	
$DUMP-HELP-FIL  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $	
END-OF-LINE-WRAP läge	
Felmeddelanden	
Felsituationer	
Fel vid laddning	
Fil	
Fil,	
flat	
ISAM	•
Filer kommando	
Filer	
FI ΔT	
TCAM	
Elet fil	•
$\begin{array}{c} \text{Flat fill} \\  $	35.
Fleranvandarbibliotek $\ldots \ldots \ldots$	
Forkorta	
Hjälpinformation	
Internal device	
fil	
filor $3$ .	
$TSAM_{FTLED}$	•
$10nm-rillen \dots 35.$	
$\begin{array}{c} \text{Normal of } \mathbf{a} \\ \text{Korman b of } \mathbf$	
$rommandofiler \dots \dots$	

Kommunikationspro	ble	m	•			•			•										64.		
Kontrollkommandon	ι.	•																	91.		
KOPIERA-DATABASE													•						36.		
KOPIERA-REALM										•			•						36.		
Kopiera-Realms .		•	•						•										24.		
LADDA-DATA-FIL .		•				•											÷	Ì	51.		
LADDA-DATABAS		•			•					•					·			•	44	51.	
LADDA-TEXT-FIL .						•	•											•	51	54.	
LAGRA-DATA	•															•		•	55.	J.,	
LAGRA-TEXT																	Ì	·	55		
LAGRA-UTAN-RAM .															Ż		·	·	51		
Lagringsformat .																	·	·	20		
Lösenord							·							•	•	•	•	•	18		
Nivå				-		-	·	•	•	•	•	•	•	•	•	•	•	•	<b>TO</b> •		
start		_																	13	17	
tabell defini	tic	n n	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	12,	т <b>!</b> •	
Nivå tabell defi	nit	.i.o	• n	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	1). 21		
Oberoende databas	hoc	: br	-1 x 7	• ni	•	• • •	•	•	•	•	•	•	•	•	•	•	•	•	о Ст.		
Odefinierade-data	ntur	0 0 0 0 1	τv	11.1.	ng	ar		•	•	•	•	•	•	•	٠	•	•	•	0.		
	rcyt	er		•	•	•	•	•	•	•	•	•	٠	•	•	•	•	•	24.	- 1	
Prioritot	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	•	•	٠	٠	51,	54.	
Prococe DDT	•	•	•	•	•	•	•	•	•	•	•	٠	•	•	•	٠	•	•	80.		
Process DDI	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	1.		
hotoh																			0		
	•	•	•	•	•	•	•	•	•	٠	•	•	•	•	٠	•	٠	•	3,	75.	
	•	•	٠	•	•	٠	•	٠	•	•	•	•	•	•	•	٠	٠	٠	3,	75.	
Realms	•	•	•	•	٠	٠	•	•	•	•	٠	•	•	٠	•	٠	٠	٠	3.		
REDIGERA-ANVANDAF	(E	•	•	•	•	•	•	•	•	•	٠	•	•	•	•	•	•	٠	31,	32.	
REDIGERA-DATABAS	•	٠	•	•	•	•	٠	•	•	٠	٠	٠	•	•	٠	•	•	•	27,	31.	
REDIGERA-DATATYPE	ER	•	•	•	•	٠	•	٠	٠	•	•	٠	•	•	•	•	•	•	19,	24,	32.
Redigera-tabell .	•	•	•	•	•	•	•	٠	•	•	٠	•	٠	٠	•	•	•	•	24,	33,	44.
Resultatlängd	•	•	•	•	•	•	•	•	•	•	٠	•	•	•	•	٠	•	•	34.		
SIBAS																					
grupper	•	•	•	•	•	•	•	•	•	•	•	•	•			•			37.		
HEADING	•	•	•	•	•	•	•	•		•									37.		
lösenord	•	•	•	•	•	•	•			•		•						•	26.		
nummer		•	•		•			•		•									26.		
realms		•	•	•															3.		
SIBAS-ITEMS		•																	38.		
SIBAS-REALMS		•				•													38.		
SIBAS-tabeller .							•												36.		
SIBAS realm	•													Ż				•	32		
SINTRAN														•	•	·	•	•	7		
Skapa dataelement									•	•	•	•	•	•	•	•	•	•	10		
SKAPA-ANVÄNDARE .									·	•	•	•	•	•	•	·	•	•	18		
SKAPA-DATABAS	·	•	•	•	•	•	•	•	•	•	•	•	•	·	•	•	•	•	25	21	
SKAPA-DATABASER	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	29, 26	JT.	
SKAPA-TABELL	•	•	•	•	•	•	•	·	•	•	•	•	•	•	•	·	·	٠	20.	10	
SLUT	•	•	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	j4,	40.	
Starta DRA	•	•	•	•	•	•	•	٠	•	•	٠	•	٠	·	•	٠	•	٠	21,	40.	
Stantnivå	•	•	•	•	•	•	•	•	٠	•	٠	٠	•	٠	·	٠	•	•	10.		
Stoplak ACCECC +	• •	•	•	•	•	٠	•	•	٠	•	٠	•	•	٠	•	•	٠	٠	17.		
STOLLEK AUGESS-TS	lear	1	·	•	•	•	•	•	•	•	٠	•	•	٠	•	٠	•	•	47.		
SIKIK-DATABAS	•	•	•	•	•	•	•	•	•	•	•		•	•			•		27.		

STRYK-TABI	- LL	L L	•	•	•	•																	30	
Systemans	var	rig	•															•	•	•	•	•	70	
Söknyckel		0			•	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	٠	•	٠	19.	
Söktid	•	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	•	٠	•	•	٠	٠	33.	
JUNCIU .	•	•	•	٠	٠	٠	٠	•	•	•	•	•	•	•	•	•	•	•		•			64.	
Tabell .	•	•	•	•	٠	٠	•	•	•		•			•									35.	
Tabelldefi	Ĺni	ti	on	sr	iv	7å						_							•	•	·	•	12	21
Tid sök .										•	•	•	•	•	•	•	•	•	•	•	·	•	1), (h	51.
Tillgängli	ior	da	+ 0	hr	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	٠	64.	
Thongoliti	-5	ua L·	נם. ייי	.00	ເ <u>ຣ</u>	;	•	•	•	•	٠	•	٠	٠	٠	•	•	•	•	٠	•	٠	26.	
Transakti(	ons	501	рТ	10	ote	ek	٠	•	•	•	•	•	•	•	•	•	•	•					63.	
VAR-ANVANI	)S-	·DA	ΤA	ΤY	'Ρ	•	•	•						•									24	
VISA-DATAE	3AS	SER																		•	•	•	26	21
VISA-REALM	4S								-	-	•	•	•	•	•	•	•	•	•	•	•	•	20,	21.
VISA-TABEL	ਜ 1	PP -	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	٠	•	٠	•	36.	
VON /VORE		311	•	•	•	•	•	٠	•	•	•	•	•	٠	٠	•	٠	•	٠	•	•	٠	40.	
XUN/AUFF	•	٠	٠	٠	•	•	٠	٠	•	•	•	•	•	•	•		•	•		•			79.	
Agare	•	•	•	•	•	•	•	•			•				•								25	
																							-)•	





### SEND US YOUR COMMENTS!

Are you frustrated because of unclear information in our manuals? 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 our manuals by rearranging the contents? You could also tell us if you like the manual.

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

#### NOTE!

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

Manual Name:ACCESS DBA Handbok	Manual number:ND-30.022.4 SW	
Which version of the product are you using?		
What problems do you have? (use extra pages if needed)		
Do you have suggestions for improving this manual?		
A DAME TO A CONTRACT OF A C		
And the open of the second		
1425 BOLD & BORNEY MILLION		
Your name:	Date:	
Company:	Position:	
Address:		
vnat are you using this manual for?		

Norsk Data's answer will be found on the reverse side.

	and hardward warded		
A Second Hilder Hand Heat and H	nysi ooyofi densal Tohebesteri ooroo		
Archine results an place pro	200855500000500000		
		in the second	
	nemolo bell'regelse i here		
	alaan na danarang mu alaan na Chistaala saad	an series	
	State	Mar	
a A B Barrier Daga Constant	and American Anometer Charles		
	10580-19		
	13700		
	at all and a constitution		
ari Archeve opresationes press	1320043		
4.21 <u>0</u> .98 (0.11) (0.21)	mate he initiality and a second s		
d by:		_ Date:	
d by:		_ Date:	
d by:		Date:	
d by:		_ Date:	
d by:	Norsk Data A.S	_ Date:	
d by:	Norsk Data A.S Documentation Departn P.O. Box 25 BOGEBUD	Date:	
1 by:	Norsk Data A.S Documentation Departn P.O. Box 25 BOGERUD N - 0621 OSLO 6 - Norv	_ Date:	
d by:	Norsk Data A.S Documentation Departn P.O. Box 25 BOGERUD N - 0621 OSLO 6 - Norv	Date:	
d by:	Norsk Data A.S Documentation Departn P.O. Box 25 BOGERUD N - 0621 OSLO 6 - Norv	nent	
d by:	Norsk Data A.S Documentation Departn P.O. Box 25 BOGERUD N - 0621 OSLO 6 - Norv	nent way	
d by:	Norsk Data A.S Documentation Departn P.O. Box 25 BOGERUD N - 0621 OSLO 6 - Norv	nent way	
d by:	Norsk Data A.S Documentation Departn P.O. Box 25 BOGERUD N - 0621 OSLO 6 - Norv	Date:	
d by:	Norsk Data A.S Documentation Departn P.O. Box 25 BOGERUD N - 0621 OSLO 6 - Norv	Date:	

