training people

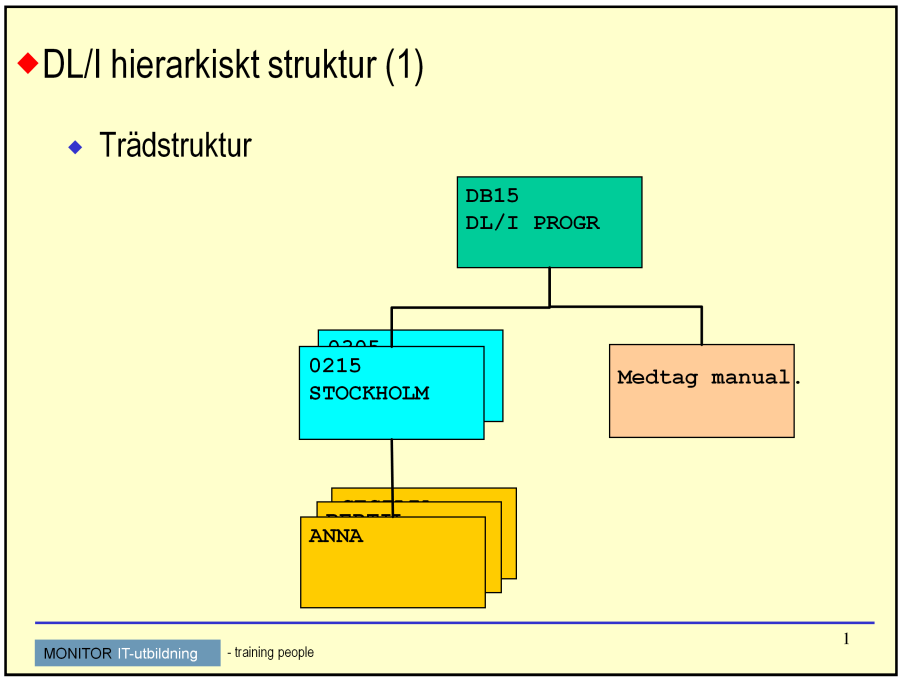
monitors

DL/I programmering från grunden

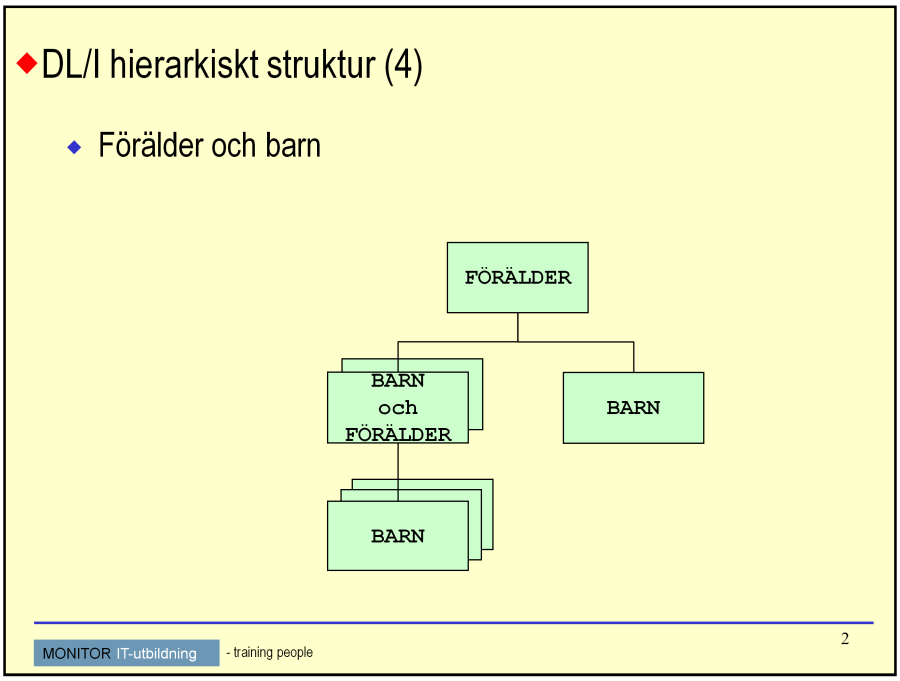
Exempelsidor

Peter Sterwe

Lär dig grunderna i DL/I-programmering på ett översiktligt och pedagogiskt sätt från företaget som har mer än trettio års erfarenhet av utbildning inom IBM z/OS Mainframe.



Med DL/I skapar vi en struktur som ger all den information som krävs, utan att ta upp extra plats i förväg.

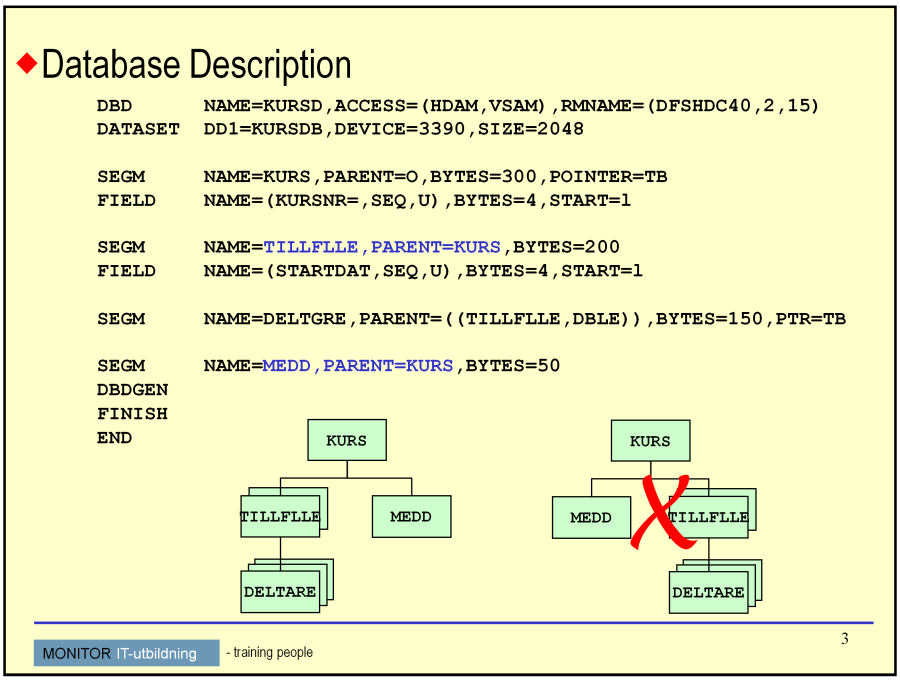


FÖRÄLDER OCH BARN.

I en struktur bestående av rotsegment och ett eller flera beroende segment finns också begreppen förälder och barn.

Rotsegmentet är i de flesta databaser förälder till ett eller flera barn.

Dessa barn kan i sin tur vara föräldrar till ytterliggare barn-segment.



DBD - DataBase Description.

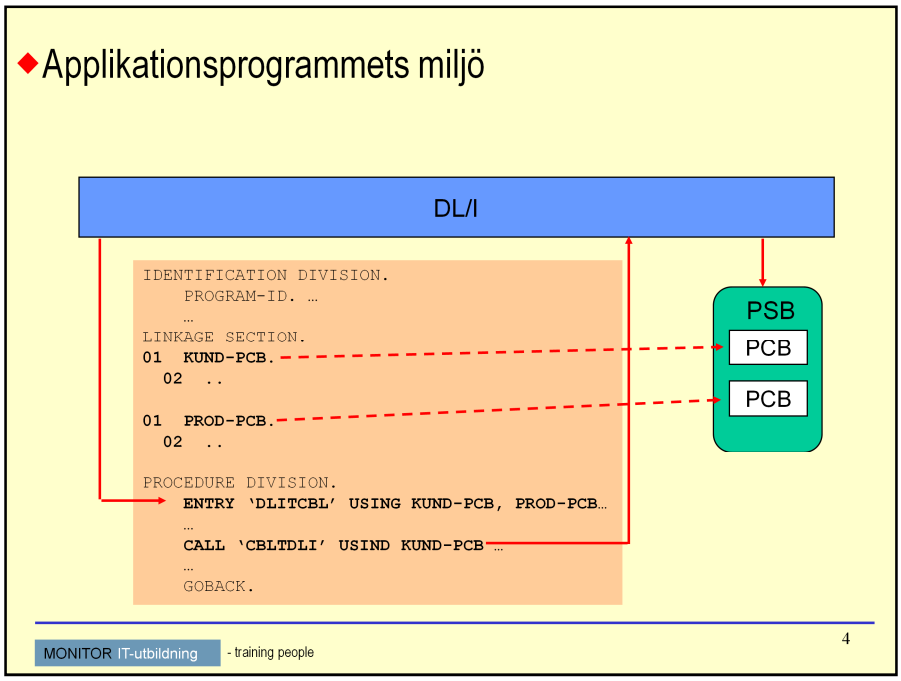
Ett DBD beskriver en databas uppbyggnad. Man skapar ett DBD per databas. Detta DBD kan sedan användas för alla applikationsprogram som ska arbeta med just den databasen.

Här finns information om fysiskt databasnamn, så kallade dd-namn på de dataset som ingår i basen, accessmetod och lagringsmedium. Här beskrivs också alla segment som ingår i databasen, med namn och segmentlängder, eventuella sorteringsfält (nycklar) och sökfält.

Strukturen beskrivs genom att man för varje segment anger vilket segment som är förälder, samt genom den inbördes ordningen mellan segmenten.

Den inbördes ordningen styr i vilken ordning segmenten ska läsas vid sekvensläsning av databasen, den så kallade hierarkiska sekvensen.

Konventionen bjuder att man ritar segmenten i hierarkisk ordning enligt DBD ”uppifrån och ned - från vänster till höger".



Eftersom DL/I betraktar vårt applikationsprogram som en subrutin förbereder DL/I exekveringen genom inladdning av aktuella kontrollblock och laddmoduler.

Applikationsprogrammet får kontrollen vid den entrypunkt som i ett cobolprogram ska heta 'DLITCBL' (*DL/I to Cobol*).

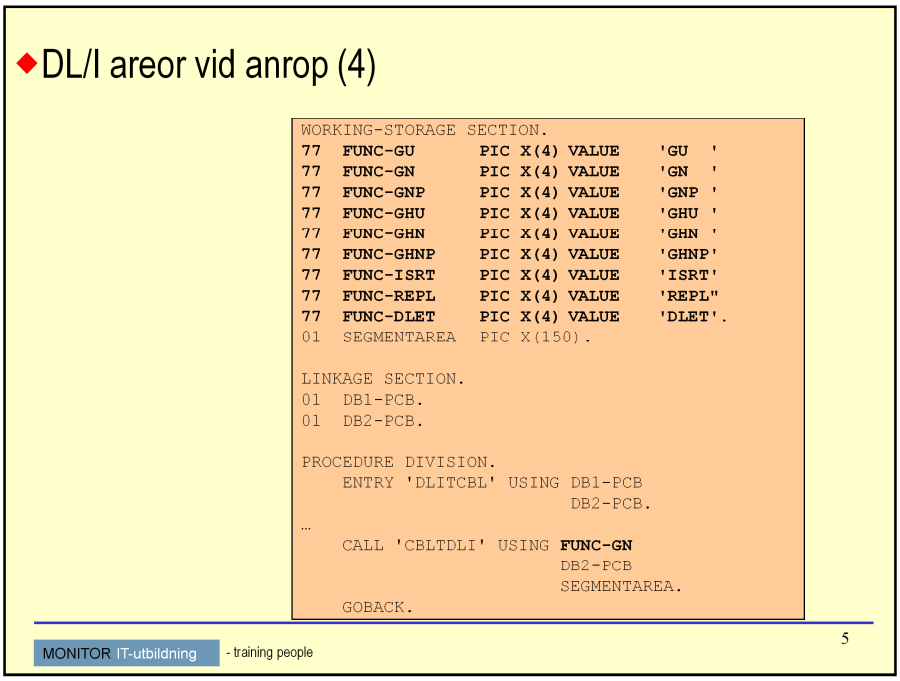
Via ENTRY-satsen får vårt program adressen till aktuella PCB:n.

Detta behövs för att vi i vårt applikationsprogram ska kunna ta vara på den information DL/I ger oss efter varje anrop.

Applikationsprogrammet gör sina anrop genom CALL på DL/I (CBLTDLI, *Cobol to DL/I*) när det vill läsa eller uppdatera på databasen.

Vid första CALL på en viss databas öppnar DL/I databasen, vilket betyder att vi inte behöver koda något "open" i vårt cobol-program.

Programmet ska avslutas med satsen GOBACK, (aldrig med STOP RUN, programmet är en subrutin till DL/I) då DL/I återfår kontrollen, avslutar exekveringen och stänger databasen mm.



PARM-COUNT

Ej obligatorisk i COBOL och det finns därför ingen anledning att använda denna parameter. För allmän kännedom gäller ändå följande.

Binärt helord i programmets Working-Storage (PIC S9(9) COMP). Ska innehålla antalet efterföljande parametrar (minst 3, max 18).

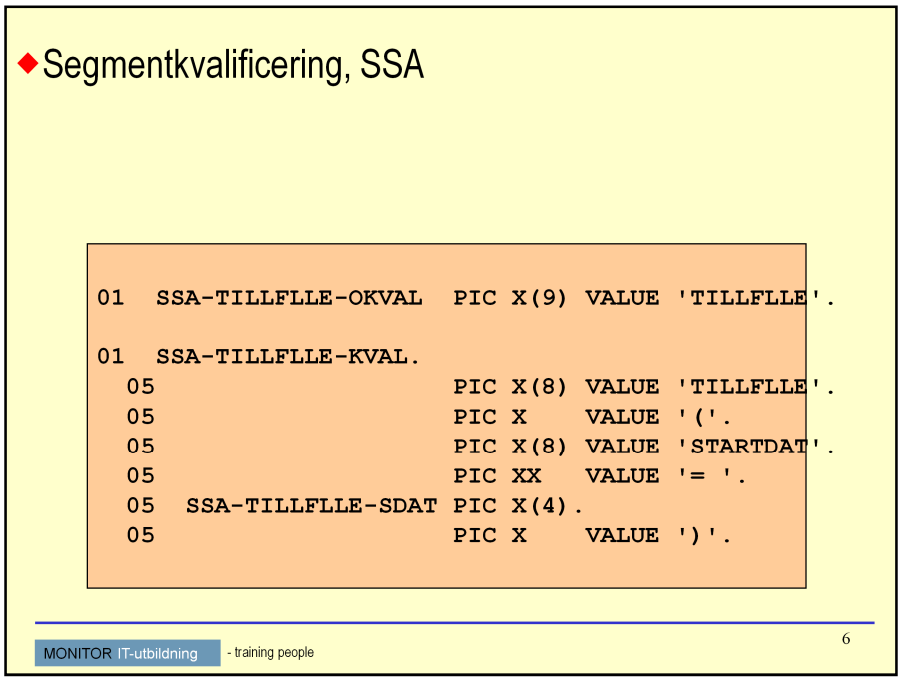
FUNKTIONSKOD

Obligatorisk.

GU	GN	GNP
GHU	GHN	GHNP
ISRT	REPL	DLET

Fyra bytes i programmets Working-Storage, PIC X(4).

Ska innehålla funktionskod enligt ovan.

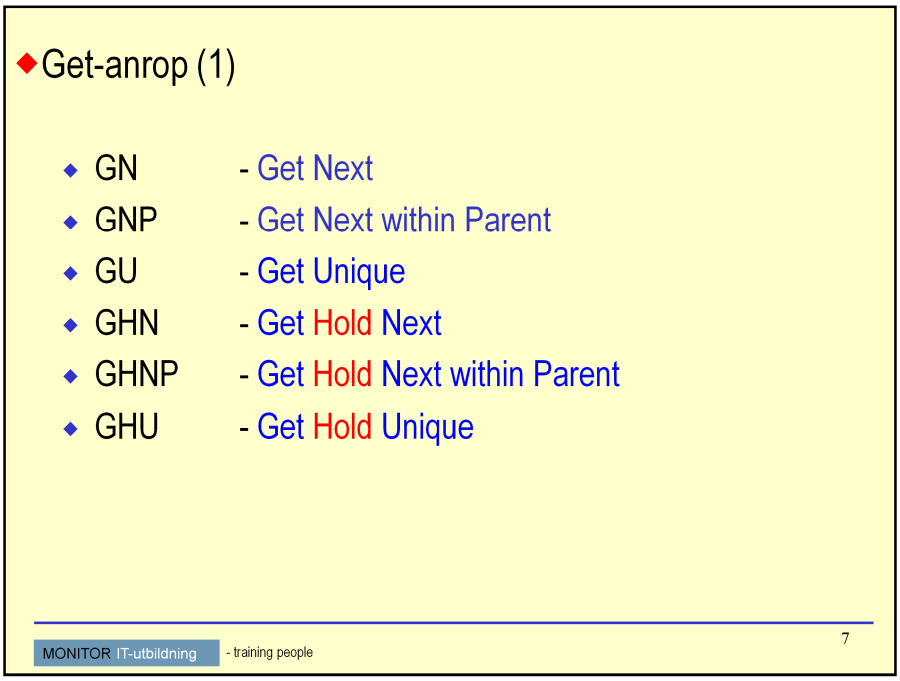


SSA = SEGMENT SEARCH ARGUMENT

För varje typ av läsning på databasen kan vi lämna med sökargument i form av SSA:er.

Ett SSA är en area i programmets Working-Storage. Den arean eller det fältet ska som minimum innehålla segmentnamn (8 bytes) följt av minst ett blanktecken. Ovan visas ett exempel på detta.

Dessutom kan ett SSA innehålla fältnamn, jämförelseoperatorer och jämförelsevärden. Exempel på dettta ser vi också på detta uppslag.



Det finns tre sätt att läsa en DL/I-databas. Följande funktionskoder kan användas:

•GN - GET NEXT

•GU - GET UNIQUE

•GNP - GET NEXT WITHIN PARENT

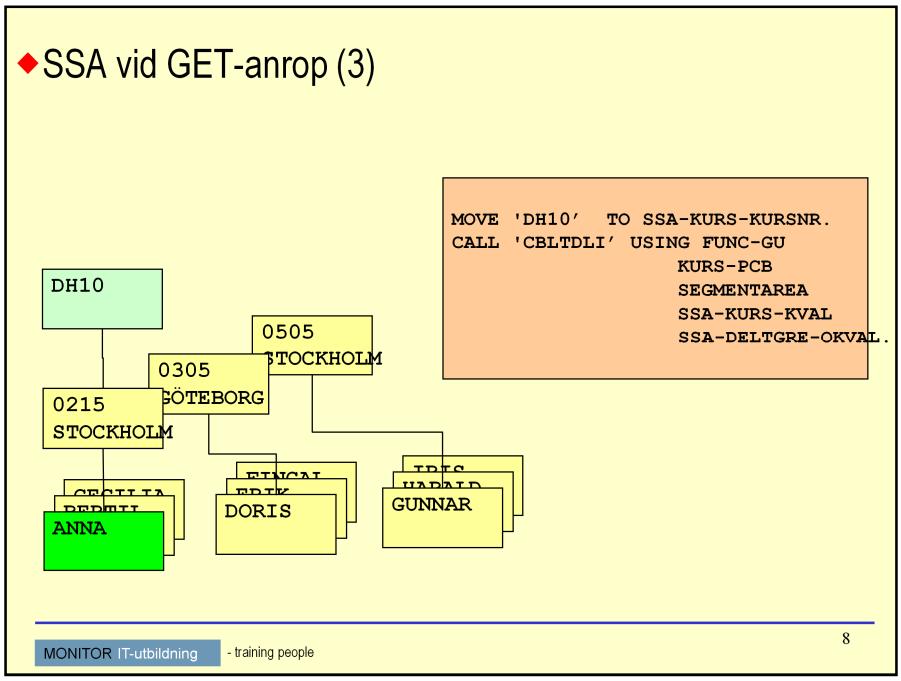
Vid programmering med standard CALL gäller följande funktionskoder om läsning kommer att efterföljas av uppdatering:

•GHN - GET HOLD NEXT

•GHU - GET HOLD UNIQUE

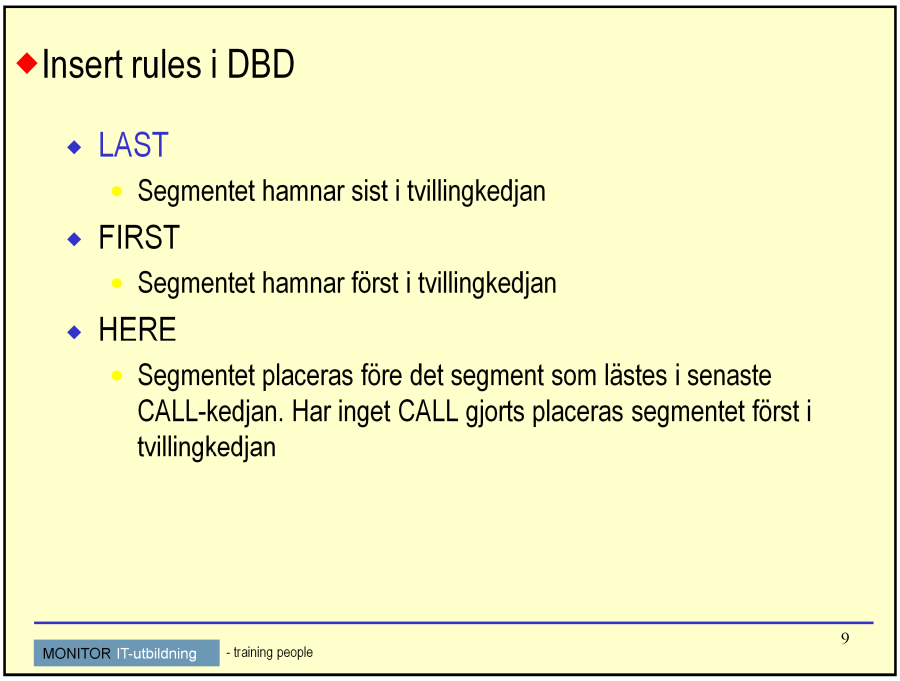
•GHNP - GET HOLD NEXT WTIHIN PARENT

Hold-anropen fungerar exakt likadant som GET - anrop utan HOLD, varför dessa inte vidare kommenteras förrän vi går igenom de uppdaterande anropen.



I det detta exemplet har vi endast kvalificerat SSA på KURS, inget SSA alls på TILLFLLE samt okvalificerat SSA på DELTGRE.

Anropet resulterar i att vi i vår IOAREA får första deltagaren under kursen ’DB15' under första bästa tillfälle, i det här fallet ’0215'.



INSERT - RULES I DBD

När databasadministratören (DBA), skapar ett DBD finns möjligheten att lämna med en sk. option vad gäller regler för hur INSERT ska ske.

De options som finns är LAST, FIRST eller HERE.

Dessa gäller för de segment som saknar unik nyckel (segment med unik nyckel hamnar där de ska, dvs. i nyckelordning).

- LAST Segmentet hamnar sist i tvillingkedjan.
- FIRST Segmentet hamnar först i tvillingkedjan.
- HERE Segmentet placeras före det segment som lästes i sista CALL.
  Har inget CALL gjorts placeras segmentet först i tvillingkedjan.

LAST är det som gäller om inte denna parameter lämnas med, och så är fallet med kursdatabasen.