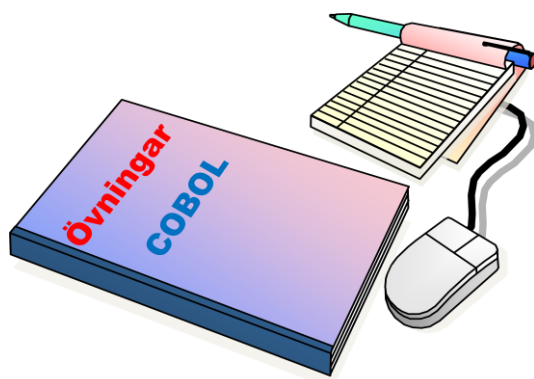


# Övningar II



---

# Innehållsförteckning

Allmänt.....	3
Uppgift 1 – Frequent Flyer File.....	4
Uppgift 2 – Frequent Flyer Program .....	5
Uppgift 3 – Bonus Report .....	6
Uppgift 4 – Frequent Flyer Mileage.....	7
Uppgift 5 – Mileage File .....	8
Uppgift 6 – Mileage Table .....	8
Uppgift 7 – Date-and-Day Program .....	9
Uppgift 8 – Airline ID.....	10
Uppgift 9 – Date Summary Report.....	11
Uppgift 10 – Destinations Report.....	11
Uppgift 11 – Day-of-Week-Name.....	11
Uppgift 12 – File Update Program.....	12
Uppgift 13 – Mileage Table, sort, pekare.....	13

## Allmänt

---

*I den här övningen kommer du att programmera ett antal COBOL-program för att skapa olika rapporter kring flera filer som innehåller information om "Frequent Flyer". Du kommer även att skapa några filer med JCL samt fylla dessa filer med olika data.*

---

- Du kommer att få information om var dina grunddata finns, samt successivt i olika övningar beskrivet vilka program som du skall programmera. Du kommer även att skapa några olika dataset samt skriva den JCL som behövs för att förverkliga detta i en IBM COBOL Batch-miljö.

## Uppgift 1 – Frequent Flyer File

Denna fil är grunden för flertalet uppgifter i övningen. Filen innehåller information om olika flygresor som en person gjort. Uppgifter som datum (*Flight-Date*), avrese- och destinationsort (*City-Pair*), flygbolag (*Airline-Id*), etc. finns lagrade här.

- Filen har ett utseende som du ser i nedanstående bild.

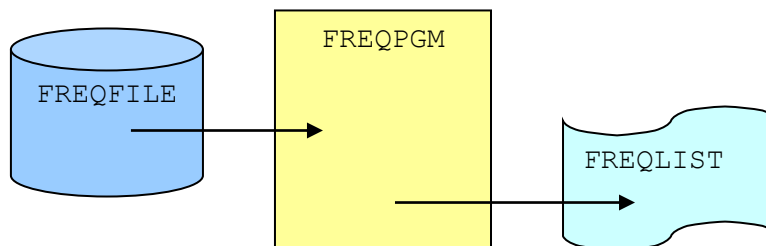
Kol	Fält	Format
01	Reserved	
02-06	Flight-Date	mm/dd
07-13	City-Pair	xxx-xxx
14-15	Airline-ID	xx
16-19	Flight-Number	nnnn
20	Class-of-Travel	x
22-25	Mileage	nnnnn
26-80	Filler	

- Skapa en indexerad fil (VSAM KSDS), där nyckeln skall bestå av *Flight-Date*. Använd programmet IDCAMS med funktionen REPRO för att fylla filen, som du kan skapa under ISPF (3.2) om du vill.
- Filens namn skall vara *userid.KURS.FREQFILE*
- Data hittar du i filen *instruktör.KURS.MILEDAT1*. Försäkra dig om att data är sorterat i stigande datumordning genom att exekvera sorteringsprogrammet (Batch) innan du fyller den indexerade filen.
  - Gör ett jobb som i Steg1 sorterar filen till ett temporärt dataset.
  - I steg 2 exekverar du IDCAMS för att använda den temporära, nu sorterade, filen som input till REPRO.

## Uppgift 2 – Frequent Flyer Program

I denna uppgift kommer du att skriva ett program som skapar en rapport från information i resefilen.

- Skriv ett program, FREQPGM, som läser samtliga poster i filen `userid.KURS.FREQFILE` och skriver en rapport.



- Rapporten skall ha följande utseende:

```
Frequent Flyer Report

Date   Id Fl#   Cl C-Pair   Mileage

01/07  01 0222   Y  DFW-ATL   731
01/10  02 0322   Y  LAS-DFW   1 056
01/11  07 0351   Y  BNA-DFW   631
. . . .
. . .
. .
.
02/23  04 0633   C  SXM-DFW   2 339
02/24  05 0730   Y  RDU-DCA   227
03/10  07 0049   Y  DFW-LAX   1 235
03/16  12 0495   C  SJU-SXM   192
. . .
. .
.

Total Mileage                nn nnn

Page 1
```

## Uppgift 3 – Bonus Report

- Komplettera ditt program så att rapporten kommer att ha följande utseende:

```
Frequent Flyer Report

Date   Id Fl#   Cl C-Pair   Mileage  Inc Bonus
-----
01/07  01 0222   Y DFW-ATL   731      731
02/11  03 1609   Y ATL-BNA   214      500
02/21  07 0351   Y BNA-DFW   631      631
. . . .
. . .
. .
.
03/03  02 0769   C SXM-DFW   227      500
04/03  01 1263   Y RDU-DFW  1 061    1 061
04/10  07 0049   Y DFW-LAX   1 235    1 235
05/15  11 0428   C LAX-DFW   1 235    1 544
. . .
. .
.

Total Mileage                mm mmm   nn nnn

Page 1

End of Report
```

- Som Du ser så har det tillkommit en kolumn, Inc Bonus. För varje resa så skall en extra bonus tillkomma enligt följande regler:
  - Samtliga klasser skall alltid få minst 500 miles när reslängden är lägre än 500, men ingen ytterligare bonus skall då tillkomma.
  - Turistklass, 'Y', skall inte ha någon extra bonus.
  - Business Class, 'C' skall erhålla 125% på aktuell reslängd.
  - First Class, 'F' skall erhålla 150% på aktuell reslängd.

## Uppgift 4 – Frequent Flyer Mileage

I denna uppgift så kommer du att skapa en ny indexerad fil som skall fyllas med data, så att du från denna kan hämta reseavstånd (Mileage) när du senare skall skapa din rapport.

- Skapa en VSAM KSDS-fil som skall fyllas med data från filen `instruktör.KURS.MILEDAT3`. Primärnyckel skall vara `City-Pair`.
- Hantera denna fil på samma sätt som filen i Uppgift 1, d.v.s den skall vara sorterad på nyckelbegreppet, som är `City-Pair`, innan du fyller den nya filen.
- Namnet på filen skall vara `userid.KURS.FREQMILE`.
- Filen har ett utseende som liknar denna:

Kol	Fält	Format
01-07	City-Pair	xxx-xxx
08	Filler	x
09-13	Mileage	nnnnn
14-80	Filler	x

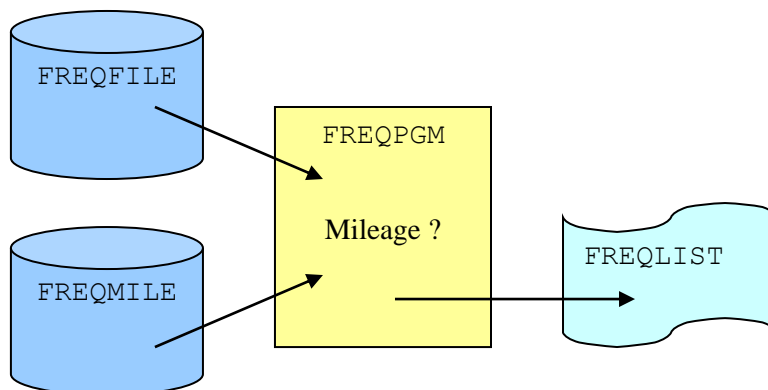
- Innehållet ser du delar av här:

```
AKL-HNL/04406
AKL-MEL/01636
AKL-NAN/01341
AKL-PPT/02544
AKL-SYD/01343
. . .
DFW-DEN/00645
DFW-DSM/00624
DFW-DTW/00987
DFW-ELP/00553
. . .
SYD-NLK/01043
YYZ-DFW/01199
YYZ-LGA/00356
YYZ-ORD/00437
```

- Notera att filen bara innehåller information om flygväg i ena riktningen, t.ex `DFW-DSM`, men `FREQFILE`-filen kan mycket väl innehålla information om en resa gjord i andra riktningen.

## Uppgift 5 – Mileage File

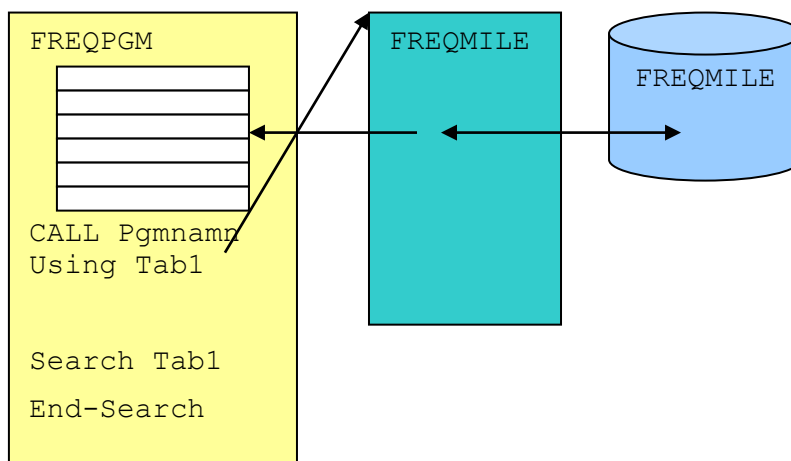
- Komplettera ditt tidigare program så att det hämtar reseavståndet från den nyligen skapade filen, `userid.KURS.FREQMILE`, och inte från posterna i filen `userid.KURS.FREQFILE`.



## Uppgift 6 – Mileage Table

*I denna uppgift så kommer du att skriva ett program som läser in filen med reseavstånden till en intern COBOL-tabell, så att du får söka i denna tabell för att få fram avstånd.*

- Skriv ett nytt program, `FREQMILE`, som läser in filen `userid.KURS.FREQMILE` till en intern COBOL-tabell. Tabellen skall finnas fysiskt i huvudprogrammet, som du tidigare skrivit. Du skall anropa "tabellprogrammet" från huvudprogrammet för att fylla tabellen med data.
- Korrigera sedan huvudprogrammet, så att det söker i tabellen för att finna reseavstånd.





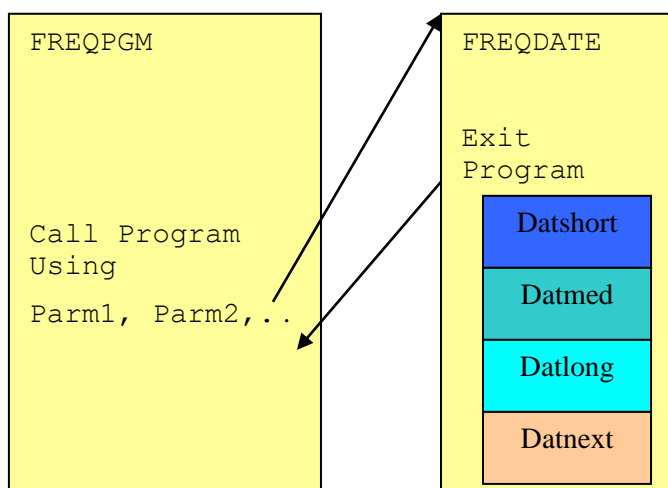
## Uppgift 7 – Date-and-Day Program

I den här uppgiften skall du skriva ett program som skall kunna skapa datum i olika format. Programmet skall kunna anropas med en parameter som beskriver i vilket format datumet skall levereras.

- Skriv ett program, FREQDATE, som skall kunna anropas som ett subprogram, där du i anropet skall kunna ange att det returnerade datumet skall vara i något av nedanstående format.

```
DATE=S    ( 4/4, 200x )
DATE=M    ( April 4, 200x )
DATE=L    ( Tuesday, April 4, 200x, at hh:mm )
```

- Programmet skall konstrueras så att det består av ett "huvudprogram" som analyserar att anropsparametern är korrekt. Sedan skall de olika datumformaten konstrueras av "inbakade" program i samma laddmodul/källkod.



- Lämpligen skapar du ett huvudprogram som innehåller enskilda program för att skapa de olika datumformaten. Du behöver ju även ett "hjälp-program" som beräknar framtida datum.
- Bestäm själv formatet för interna anropsparametrar etc. Datum som programmet skall returnera är dagens datum samt datum om 30 dagar.
- Komplettera sedan huvudprogrammet så att det kan anropas med en parameter vid exekveringen (DATE=S/DATE=M/DATE=L). Programmet skall även skriva ut ett meddelande om exekveringsparameter utelämnats eller är felaktig och avslutas med lämplig returkod.
- Uppdatera programmet, så att rapporten får ett utseende som liknar denna:

```

Frequent Flyer Report

Printed      : Monday, March 3, 200x, at hh:mm
Valid until  : Friday, April 7, 200x

Date   Id Fl#   Cl C-Pair   Mileage Inc Bonus

. . . . .
. . . . .
. . . . .
. . . .
. .
.
Total Mileage           mm mmm   nn nnn

Page 1

End of Report

```

## Uppgift 8 – Airline ID

- Komplettera rapporten så att den skriver ut namnet på flygbolaget. Detta namn finns i en VSAM KSDS-fil med namnet *instruktör.KURS.FREQLINE*, där Airline-Id är nyckel.
- Filen har följande utseende:

Kol	Fält	Format
01-02	Airline-Id	nn
03	Filler	x
04-29	Airline-Name	xxxx...xxx (25)
30-80	Filler	x

- Det kan vara lämpligt att läsa in denna fil i en intern COBOL-tabell och söka i denna för att hämta Airline-Name.
- Räkna med att maxantalet flygbolag är 25.

## Uppgift 9 – Date Summary Report

- Skriv ett program, `FREQDATR`, som kan skriva ut en rapport över vilka resor som gjorts under en viss tidsperiod, fr.o.m. datum – t.o.m. datum. Denna datumgräns skall du kunna ange som en parameter vid exekveringen.
- Rapporten skall visa vilka flygturer som gjorts under en viss period med tidpunkt enligt parameter eller första resa efter angivet startdatum samt senaste resa innan angivet slutdatum.
- Rapportens utseende skall ha likheter med den tidigare rapporten.

## Uppgift 10 – Destinations Report

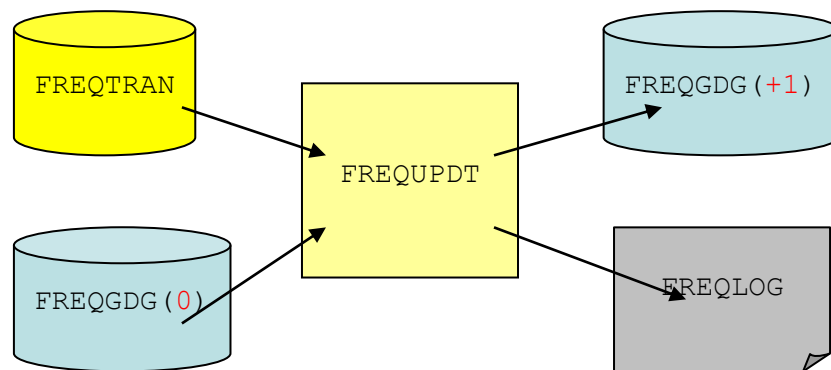
- Skriv ett program, `FREQDSTR` som kan skriva ut en rapport över vilka resor som gjorts med ett visst City-Pair.
- Läs in `City-Pair` via `SYSIN`. Flera `City-Pair` skall kunna anges som olika poster/rader.
- Programmet skall även kunna hantera situationen när endast avrese- eller destinations-ort anges, så att ledningen kan se vilka resor som gjorts från eller till en viss stad. Datum skall givetvis finnas med.

## Uppgift 11 – Day-of-Week-Name

- Komplettera rapporten så att den innehåller dagnamn när resan gjorts.
- Du kommer att behöva komplettera datum-programmet, `FREQDAT`, så att det kan leverera ett dagnamn med ledning av datum. Antag att året är innevarande år.

## Uppgift 12 – File Update Program

I denna uppgift så kommer du att skriva ett program som läser in en transaktionsfil med information om förändringar som skall göras i resefilen. Du kommer att överföra information från resefilen till en ny generation i ett GDG, som du även kommer att skapa.



- Skapa en generationsdatasetgrupp med namnet `userid.KURS.FREQGDG`.
- Gruppen skall innehålla maximalt 10 generationer, och föråldrade generationer skall raderas automatiskt.
- Skapa ett jobb med JCL för att med programmet IDCAMS läsa in filen `instruktör.KURS.FREQFILE` och skapa med REPRO en ny generation i den nyligen skapade generationsdatasetgruppen
- Gör samtidigt en LISTCAT för att kontrollera att gruppen har de egenskaper som du tänkt dig.
- Skriv sedan ett program, FREQUPDT, som läser in filen `instruktör.KURS.FREQTRAN`, som innehåller poster med transaktioner som skall appliceras på den gällande generationen (FREQGDG (0)). Den uppdaterade filen skall vara en ny generation (+1) i filen `userid.KURS.FREQGDG`.
- Transaktionerna har följande utseende:

```
Pos 1 : A (Add)
Pos 2 - 80 : Komplet format, som du ser i Uppgift 1.

Pos 1 : R (Remove)
Pos 2 - 6 : Flight Date

Pos 1 : U (Update)
Pos 2 : Flight Class
```

- Programmet skall även skriva en loggfil, `FREQLOG`, med information om gjorda förändringar, samt även information om felaktiga transaktioner, om några sådana skulle upptäckas. Loggen skall skrivas ut på `Sysout`.
- När programmet är avslutat så skall efterföljande jobbsteg exekvera programmet `IDCAMS` för att återskapa filen `FREQFILE` med den nya generationen som indata.
- Exekvera sedan programmet `FREQPGM` för att skapa en ny rapport.
- Kontrollera resultatet och jämför med en tidigare rapport samt transaktionerna i `FREQTRAN`.

## Uppgift 13 – Mileage Table, sort, pekare

*I denna uppgift så kommer du att använda pekare och LE-call för att förändra programmen i Uppgift 6.*

- Programmet `FREQMIL2` skall använda file `userid.KURS.FREQMILE` för fylla en intern COBOL-tabell med data från filen samt kunna göra sökning i tabellen.
- Filen antas vara osorterad och programmet skall göra en COBOL internsort av filen innan tabellen fylls med data. Nyckeln är som tidigare `City-Pair`.
- Utrymme för tabellen skall allokeras genom att utnyttja *LE Storage Management-call*.
- Programmet `FREQPGM` skall anropa `FREQMILE` med begäran om sökning av visst `City-Pair` samt med en pekare, som detta program skall peka på den sökta posten i tabellen vid lyckad sökning.
- Om sökningen misslyckas skall pekaren ha värdet `Null`.

