## Laboration 1

Under den här labben kommer Du att få arbeta med att skapa olika databasobjekt. Du bör ha tagit del av föreläsning 1 - 5 för att klara labben utan problem.

**OBS!** Tänk på att alltid ta hand om dina transaktioner med commit!

Redovisa labben i en välstrukturerad, tydlig och lättläst labbrapport. Använd detta dokument som labbrapport. **Du fyller i dina kommentarer/frågor och lösningar i befintligt dokument.**

**Uppgift 1**

Skapa en sekvens med namnet **myseq** den skall starta med 1 och öka med 1. Sekvensens metod **nextval** returnerar numerisk datatyp.

**Uppgift 2**

Skapa en tabellstruktur enligt följande ritning

KUND

# persnr

\* username

\* passwd

\* fnamn

\* enamn

\* kredittyp

o telnr

KUNDORDER

# ordnr

(#)persnr

\* datum

KUNDVAGN

# radnr

(#)ordnr

(#)artnr

\* antal

ARTIKEL

# artnr

(#)vgnr

\* artnamn

\* pris

VARUGRUPP

# vgnr

\* vgnamn

ARTIKELBILD

# bildnr

(#)artnr

\* filtyp

\* width

\* height

\* path

**Förklaring till notation**

# = Primary key

(#) = Foreign key

\* = Mandatory (Obligatorisk => NOT NULL)

o = Optional (Kan vara NULL)

### Följande constraints skall finnas i tabellerna:

**kund.kredittyp** CHECK ('hög','medel','låg')

**artikelbild.filtyp** CHECK ('gif','jpg')

**kundorder.ordnr** (genereras med sekvensen myseq)

Eftersom en sekvens returnerar number så vet du vilken datatyp du ska ha på ordnr och radnr i tabellerna KUNDORDER och KUNDVAGN.

**kundvagn.radnr** (genereas med sekvensen myseq)

**kundorder.datum** (datatyp = DATE, SYSDATE)

**kund.username** (skall vara unikt, constraint UNIQUE)

**Alla FK** skall förses med **NOT NULL**

Namnge alla constraints som du deklarerar enligt principen: **tabellnamn\_kolumnnamn\_constrainttyp**, där du med fördel kan använda standarden för constrainttyper: **CK** = CHECK,

**PK** = PRIMARY KEY, **FK** = FOREIGN KEY och **UQ** = UNIQUE.

**OBS!** Tänk noga igenom datatyper och längder för kolumnerna. Vad är en passande datatyp och längd på just denna kolumn. Om någon skriver bara number medför detta att det blir number(38).

**OBS! Gör så här för att ge namn till constraints när de skapas:**

**EXEMPEL**

-- Lagrar information om studenter, deklarera not null när kolumnen skapas!

create table student(

studnr number(6),

persnr varchar2(11) not null,

fnamn varchar2(40) not null,

enamn varchar2(60) not null);

/\* Lägger till alla constraints utom not null.

Se till att ge dina constraints ett namn när de skapas! \*/

alter table student

add constraint student\_studnr\_pk primary key(studnr)

add constraint student\_persnr\_uq unique(persnr);

**Uppgift 3**

Lägg till tre rader i tabellen kund**.**

**Uppgift 4**

Lägg till två rader i tabellen varugrupp.

**Uppgift 5**

Lägg till tre rader i tabellen artikel.

**Uppgift 6**

Genomför en försäljning genom att skapa **en rad** i tabellen kundorder och **två rader** i tabellen kundvagn. **Kom ihåg** att använda sekvensen för att generera PK i tabellerna.

**OBS!** När du har lagt till en rad i parent-tabellen kundorder, så måste du kontrollera vilket värde som PK = ordnr fick från sekvensen. Gör detta genom att skriva:

select \* from kundorder;

Ta detta ordnr och lägg som FK i tabellen kundvagn.

Det är genom FK i kundvagn som vi vet vilken order som raden i kundvagnstabellen tillhör.

**Uppgift 7**

Höj priset på alla artiklar med 23 %.

**Uppgift 8**

Uppdatera telefonnummer för en valfri kund.

**Uppgift 9**

Ta bort alla rader ur tabellen kundorder! **Vad händer och varför!**

**Uppgift 10**

Ta bort alla databasobjekt som du skapat under labben (tabeller och en sekvens).

**Vilken DDL-kod krävs för detta? I vilken ordning måste tabellerna tas bort?**

**Laborationen lämnas i Learn (Blackboard). Klicka på länken Uppgifter.**

**Pär Douhan** [**pdo@du.se**](mailto:pdo@du.se)