

Laborationen redovisas i form av ett SQL-script som steg för steg löser uppgifterna i den ordning som de finns beskrivna nedan.

Varje steg ska vara tydligt märkta med en kommentar till exempel

```
-- Uppgift 1
INSERT INTO....
SELECT * FROM.....
-- Uppgift 2
UPDATE.....
```

Det kan vara smart när man är klar med en uppgift att man kommenterar bort sina föregående queries så att de inte körs om och om igen. Tips: Det finns fler metoder i MySQL än vad man tror!

Databasen som vi vill ställa frågorna till finns i filen **Laboration1.sql** som vi kan importera till MySQL. Tänk på att SQL skiljer sig mellan olika databasimplementationer och när ni söker information om hur något kan göras kolla att det gäller för den version av MySQL som vi använder.

## *MoonMissions*

1. Använd "CREATE TABLE..AS SELECT" för att ta ut alla kolumner för alla lyckade uppdrag (Successful outcome) i "moonmission" och sätt in i en ny tabell med namn "successful\_mission".
2. Skriv en query som ändrar tabellen "successful\_mission" så att kolumnen "mission\_id" blir en primärnyckel och auto inkrementerar. Ta inte bort tabellen och skapa den på nytt, enbart egenskaperna för mission\_id ska ändras.
3. I kolumnen 'operator' har det smugit sig in ett eller flera mellanslag i operatörens namn. Skriv queries som uppdaterar och tar bort mellanslagen kring operatören för både "successful\_mission" och "moon\_missions" .
4. Skriv en query som tar bort alla uppdrag utförda 2010 eller senare från "successful\_mission".
5. Gör en SELECT för att ta ut samtliga rader och kolumner från tabellen "account", men slå ihop 'first\_name' och 'last\_name' till en ny kolumn 'name', samt lägg till en extra kolumn 'gender' som du ger värdet 'female' för alla användare vars näst sista siffra i personnumret är jämn, och värdet 'male' för de användare där siffran är udda.
6. Skriv en query som tar bort alla kvinnor födda före 1970.
7. Skriv en query som returnerar två kolumner 'gender' och 'average\_age', och två rader där ena visar medelåldern för män, och den andra medelåldern för kvinnor för alla användare i tabellen 'account'.

## Book Store

I de följande uppgifterna ska vi skriva SQL kod för att skapa en egen databas, egna tabeller och användare som får komma åt databasen. Om databasen redan existerar så ska den tas bort och skapas på nytt.

Skapa en databas som heter bookstore med följande tabeller:

Tabell: "author"

I tabellen författare vill vi ha en "Identitetskolumn" (identity) kallad id som PK. Därutöver vill vi ha kolumnerna: first\_name, last\_name och birth\_date i passande datatyper.

Tabell: "book" I tabellen böcker vill vi ha isbn (ISBN13) som primärnyckel med lämpliga constraints. Utöver det vill vi ha kolumnerna: title, language, price, och publication\_date av passande datatyper.

Sist vill vi ha en FK-kolumn "author\_id" som pekar mot tabellen "author".

*Kolumnen language får gärna brytas ut till en egen tabell för att inte lagra samma språk om och om igen utan istället lagra en fk till language tabellen.*

Tabell: "bookstore" Utöver ett identity-id så behöver tabellen kolumner för att lagra butiksnamn samt stad.

Tabell: "inventory" I denna tabell vill vi ha 3 kolumner: store\_id som kopplas mot Butiker, isbn som kopplas mot böcker, samt amount som säger hur många exemplar det finns av en given bok i en viss butik. Som PK vill vi ha en kompositnyckel på kolumnerna store\_id och ISBN.

Lägg in ett par författare, böcker, butiker och lagersaldon för böcker som finns i de olika butikerna.

### Vy: "total\_author\_book\_value"

Skapa sedan en en vy som sammanställer data från tabellerna.

Vyn ska innehålla följande 4 kolumner (med en rad per författare):

"name" – Hela namnet på författaren.

"age" – Hur gammal författaren är.

"book\_title\_count" – Hur många olika titlar vi har i "Böcker" av den angivna författaren.

"inventory\_value" – Totala värdet (pris) för författarens böcker i samtliga butiker.

Exempel:

"SELECT \* FROM total\_author\_book\_value LIMIT 1"

name	age	book_title_count	inventory_value
Eddie Neumann	23 år	3 st	4182 kr

## Användare och Behörigheter:

För att accessa databasen vill vi ha två konton med rätt behörigheter.

Den ena användaren ska vara ett utvecklarkonto och ha rättigheter att på bokdatabasen lägga till, ta bort och uppdatera tabeller samt deras innehåll.

Det andra kontot är tänkt för en web server att ansluta till databasen via och ska endast ha

rättigheter att utföra CRUD operationer och inte skapa eller ta bort databaser.  
Inget av kontona ska få lägga till nya användare eller skapa nya databaser.