

SQL 1

När du lämnar in dina svar ska du för varje uppgift kopiera den SQL-fråga du använde i PostgreSQL för att lösa uppgiften.

Uppgift 1: Student

Använd student.sql för att i PostgreSQL skapa de tabeller och lägga till de data som behövs för att svara på uppgifterna nedan:

- 1) Lista (SELECT) alla uppgifter om alla studenter. Sortera resultatet i sjunkande ordning efter studenternas postnummer och därefter deras förnamn (det vill säga studenter med samma postnummer sorteras med förnamn).

Svar:

```
SELECT * FROM student  
ORDER BY postnr DESC, förnamn;
```

- 2) Lista portalid för alla studenter som läst kursen DT022G.

Svar: `SELECT portalid, kurskod FROM betyg WHERE kurskod LIKE 'DT022G' ORDER BY portalid;`

- 3) Lista portalid, kurskod och år för alla studenter som har läst kursen DT011G före (innan) år 2012.

Svar: `SELECT portalid, kurskod, år FROM betyg WHERE kurskod LIKE 'DT011G' AND "år" <= '2012' ORDER BY portalid ;`

- 4) Du ska lista portalid, kurskod och betyg för studenter som har läst kursen DT011G. Låt alla betyg förbättras med 1. Resultatet ska vara sorterat så att studenter med sämst (= lägst) betyg kommer först.

Svar:

```
UPDATE betyg SET betyg = betyg + 1;  
SELECT portalid, kurskod, år, betyg FROM betyg WHERE kurskod LIKE 'DT011G'  
ORDER BY betyg ASC;
```

Uppgift 2: Föreningsdatabas

Du ska i denna uppgift utgå från det lösningsförslag du fått på inlämningsuppgift 4. Du ska med SQL skapa alla tabeller (CREATE TABLE) och lägga in data i dessa (INSERT INTO). I tabellerna nedan ser du bland annat vilken datatyp de olika attributen ska ha, om de är nödvändigt att ange uppgift (NOT NULL) och övrigt som gäller för attributet. Glöm inte bort att lägga till CONSTRAINT för alla främmande nycklar.

När du ska använda de speciella (för PostgreSQL) värdena infinity och -infinity i SQL-kod är det viktigt att du omsluter dessa med enkla citattecken.

Postort

attribut	datatyp	nödvändighet	övrigt
postnr	char 5		Primärnyckel
postort	text	ja	

Person

attribut	datatyp	nödvändighet	övrigt
id	serial		Primärnyckel
förnamn	text	ja	
efternamn	text	ja	
gatuadress	text	ja	
postnr	char 5	ja	
epost	text	ja	Måste vara unikt
lösenord	char 32		

Inloggning

attribut	datatyp	nödvändighet	övrigt
id	serial		Primärnyckel
ip	text	ja	
tidpunkt	timestamp	ja	
lyckad	boolean	ja	

Administratör

attribut	datatyp	nödvändighet	övrigt
id	int		Primärnyckel
arvode	int	ja	Standardvärde 0

Medlem

attribut	datatyp	nödvändighet	övrigt
id	int		Primärnyckel
medlemsnr	serial	ja	
datum_medlem	date	ja	

Ledare

attribut	datatyp	nödvändighet	övrigt
id	int		Primärnyckel
utbildning	text		

Nyhet

attribut	datatyp	nödvändighet	övrigt
id	serial		Primärnyckel
rubrik	text	ja	
innehåll	text	ja	
datum_skapad	timestamp		Standardvärde now() (tid just nu). När nyheten skrevs.
datum_från	timestamp		Standardvärde -infinity (tidigare än alla andra tider). Från vilket datum nyheten ska visas.
datum_till	timestamp		Standardvärde infinity (senare än alla andra tider). Till vilket datum nyheten ska visas.

Avgift

attribut	datatyp	nödvändighet	övrigt
id	serial		Primärnyckel
avgift	int	ja	
år	char 4	ja	Måste vara unikt

Medlemsavgift

attribut	datatyp	nödvändighet	övrigt
medlemsid	int		Primärnyckel
avgiftsid	int		Primärnyckel
betald	boolean		

Sektion

attribut	datatyp	nödvändighet	övrigt
id	serial		Primärnyckel
namn	text	ja	
beskrivning	text	ja	
datum_bildat	date	ja	

Medlemssektion

attribut	datatyp	nödvändighet	övrigt
medlemsid	int		Primärnyckel
sektionsid	int		Primärnyckel

- 1) Skapa ett nytt schema i databasen som heter sportsclub. I detta schema skapar du sen tabellerna (CREATE TABLE) med full entitets-, referens och dataintegritet. Nödvändighet (NULL tillåts eller inte), om attributen ska vara unika och standardvärden för attributen ska även anges korrekt.

Svar: Bifoga en SQL-fil med all kod. Döp filen till create.sql.

- 2) Du ska lägga in påhittade uppgifter (INSERT INTO) i tabellerna du skapade i deluppgift 1. Lägg till minst tre rader i varje tabell.

Svar: Bifoga en SQL-fil med all kod. Döp filen till insert.sql. Alternativt att du skapar en enda SQL-fil som innehåller svaren på både uppgift 1 och 2. Döp i så fall denna fil till create_and_insert.sql.

- 3) Lista namn och beskrivning på alla sektioner. Sortera resultatet i bokstavsordning efter sektionens namn.

Svar: SELECT namn, beskrivning
FROM sektion
ORDER BY namn ASC;

- 4) Lista all information om alla nyheter som är giltiga just nu. Sortera resultaten efter den tidpunkt de var skapade. Med giltiga just nu menas att aktuell tidpunkt ska ligga inom intervallet datum_från till datum_till.

Svar:

```
SELECT rubrik, innehåll, datum_skapad, datum_från, datum_till
FROM nyhet
WHERE (datum_från <= CURRENT_DATE AND datum_från <= datum_till)
ORDER BY datum_skapad ASC;
```