# SQL 2

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 (det vill säga samma som inlämningsuppgift 6) för att svara på uppgifterna nedan:

1. Lista alla studenter vars efternamn innehåller bokstaven v.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| portalid | födelsedatum | förnamn | efternamn | adress | postnr |
| sasv1010 | 1985-06-06 | Sara | Svensson | Blomstigen 2 | 83432 |
| vivi1501 | 1975-01-13 | Viktor | Viktorsson | Adelsgatan 3 | 62192 |

Svar:

SELECT portalid, födelsedatum, förnamn, efternamn, adress, postnr

FROM student

WHERE portalid LIKE '%v%'

ORDER BY portalid;

1. Lista antalet kurser studenten med portalid bean1100 läst.

|  |
| --- |
| antal |
| 3 |

Svar:

SELECT portalid, COUNT(portalid)

AS antal

FROM betyg

WHERE portalid = 'bean1100'

GROUP BY portalid

ORDER BY portalid;

1. Lista genomsnittsbetyget för alla studenter tillsammans.

|  |
| --- |
| genomsnitt |
| 2.66667 |

Svar:

SELECT ROUND(AVG(betyg),5)

AS genomsnitt

FROM betyg;

(rundade svaret till 5 decimaler).

1. Beräkna genomsnittsbetyget för varje student. Portalid och genomsnitts betyg ska listas.

|  |  |
| --- | --- |
| portalid | genomsnitt |
| kama1203 |  |
| sasv1010 | 3 |
| vivi1501 | 3 |
| bean1100 | 2.3333 |

Svar:

SELECT portalid, ROUND(avg(betyg),4)

AS genomsnitt

FROM betyg

GROUP BY portalid

ORDER BY portalid;

1. Beräkna antalet kurser varje student läst. Studentens portalid, förnamn, efternamn och antalet kurser ska listas. Resultatet ska sorteras så att studenter som har läst flest antal kurser listas först.

|  |  |  |  |
| --- | --- | --- | --- |
| portalid | förnamn | efternamn | antal |
| bean1100 | Bertil | Andersson | 3 |
| sasv1010 | Sara | Svensson | 2 |
| kama1203 | Karl | Martinsson | 1 |
| vivi1501 | Viktor | Viktorsson | 1 |
| anan0912 | Anna | Andersson | 0 |

Svar:

SELECT portalid, COUNT(portalid)

AS antal

FROM betyg

GROUP BY portalid

ORDER BY antal DESC;

1. Lista portalid, kurskod och betyg för den eller de studenter som har fått högst betyg oavsett kurs.

|  |  |  |
| --- | --- | --- |
| portalid | kurskod | betyg |
| bean1100 | DT076G | 4 |
| sasv1010 | DT022G | 4 |

Svar: SELECT portalid, kurskod, betyg

FROM betyg

WHERE betyg >=4

ORDER BY portalid;

1. Lista portalid för de studenter som har sämre genomsnittsbetyg än genomsnittet för alla studenter.

|  |
| --- |
| portalid |
| bean1100 |

Svar:

SELECT portalid

FROM betyg

GROUP BY portalid

HAVING AVG(betyg) < ( SELECT AVG(betyg) FROM betyg);

1. Populära kurser – lista kurskod för den eller de kurser som har fler antal studenter än det genomsnittliga studentantalet på alla kurser. Det vill säga alla kurser som har:  
   (antal\_deltagare\_på\_en\_kurs) > (totala\_antalet\_studenter / antal\_unika\_kurser\_som\_någon\_läst)

|  |  |
| --- | --- |
| kurskod | antal |
| DT011G | 2 |
| DT022G | 3 |

Svar:  
SELECT kurskod, COUNT(kurskod)

FROM betyg

GROUP BY kurskod

HAVING COUNT(kurskod) > COUNT(portalid)/ (COUNT(kurskod))

1. Lista all information (portalid, förnamn, postnr m.m.) om de studenter som inte har läst kursen DT022G.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| portalid | födelsedatum | förnamn | efternamn | adress | postnr |
| anan0912 | 1995-05-05 | Anna | Andersson | Splintvägen 5 | 83172 |
| vivi1501 | 1975-01-13 | Viktor | Viktorsson | Adelsgatan 3 | 62192 |

Svar:

SELECT s.\*

FROM betyg b

RIGHT JOIN student s

ON b.kurskod = 'DT022G' AND s.portalid = b.portalid

WHERE kurskod is null;

1. Lista all information om de studenter som har läst någon kurs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| portalid | födelsedatum | förnamn | efternamn | adress | postnr |
| bean1100 | 1970-10-10 | Bertil | Andersson | Stubbvägen 1 | 83172 |
| sasv1010 | 1985-06-06 | Sara | Svensson | Blomstigen 2 | 83432 |
| kama1203 | 1980-12-12 | Karl | Martinsson | Faktorsvägen 13 | 89597 |
| vivi1501 | 1975-01-13 | Viktor | Viktorsson | Adelsgatan 3 | 62192 |

Svar:

SELECT s.\*

FROM betyg k

LEFT JOIN student s

ON k.portalid = s.portalid

GROUP BY s.portalid

ORDER BY portalid;

1. Lista all information om de studenter som inte har läst någon kurs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| portalid | födelsedatum | förnamn | efternamn | adress | postnr |
| anan0912 | 1995-05-05 | Anna | Andersson | Splintvägen 5 | 83172 |

Svar:  
SELECT s.\*

FROM betyg b

RIGHT JOIN student s

ON s.portalid = b.portalid

WHERE kurskod is null;

1. Lista portalid för de studenter som har läst samma kurs som studenten med portalid kama1203.

|  |
| --- |
| portalid |
| bean1100 |
| sasv1010 |
| kama1203 |

Svar:

SELECT DISTINCT b1.portalid

FROM betyg b1, betyg b2

WHERE b1.kurskod = b2.kurskod AND b1.portalid <> b2.portalid

GROUP BY b1.portalid

ORDER BY b1.portalid;

1. Lista portalid, efternamn och kursnamn för alla studenter som har läst kursen DT011G. Sortera resultatet på efternamnet i bokstavsordning (a-z).

|  |  |  |  |
| --- | --- | --- | --- |
| portalid | förnamn | efternamn | kursnamn |
| bean1100 | Bertil | Andersson | Operativsystem introduktionskurs |
| sasv1010 | Sara | Svensson | Operativsystem introduktionskurs |

Svar:

CREATE VIEW DT011G

AS SELECT kurskod, namn

FROM kurs

WHERE kurskod = 'DT011G';

SELECT s.portalid, s.förnamn, s.efternamn, k.namn

FROM student s, betyg b, DT011G k

WHERE s.portalid = b.portalid

AND b.kurskod = k.kurskod

ORDER BY s.efternamn ASC;

1. Lista portalid, efternamn och kursnamn för alla studenter som har läst en kurs vars kursnamn innehåller ordet data. Sortera resultatet på efternamnet i bokstavsordning.

|  |  |  |  |
| --- | --- | --- | --- |
| portalid | förnamn | efternamn | kursnamn |
| bean1100 | Bertil | Andersson | Databaser, introduktion |
| bean1100 | Bertil | Andersson | Databaser, implementering och modellering |
| kama1203 | Karl | Martinsson | Databaser, introduktion |
| sasv1010 | Sara | Svensson | Databaser, introduktion |

Svar:

CREATE VIEW datakurs

AS SELECT kurskod, namn

FROM kurs

WHERE namn LIKE '%Data%' OR namn LIKE ‘%data%’;

SELECT s.portalid, s.förnamn, s.efternamn, k.namn

FROM student s, betyg b, datakurs k

WHERE s.portalid = b.portalid

AND b.kurskod = k.kurskod

ORDER BY s.efternamn ASC;

1. Lista portalid, förnamn, efternamn och postnr för alla studenter som har samma postnummer som någon annan student. Sortera resultatet efter postnummer i stigande ordning.

|  |  |  |  |
| --- | --- | --- | --- |
| portalid | förnamn | efternamn | postnr |
| bean1100 | Bertil | Andersson | 83172 |
| anan0912 | Anna | Andersson | 83172 |

Svar:

SELECT DISTINCT b1.portalid, b1.förnamn, b1.efternamn, b1.postnr

FROM student b1, student b2

WHERE b1.postnr = b2.postnr AND b1.portalid <> b2.portalid

GROUP BY b1.portalid

ORDER BY b1.postnr ASC, b1.portalid DESC;

(Då både anna och bertil hade samma postnr så valde jag att inkludera en sekundär sorterings metod, då per portalid i sjunkande ordning.

### Uppgift 2: Föreningsdatabas

Använd den föreningsdatabas du skapade i inlämningsuppgift 6, uppgift 2 för att svara på uppgifterna nedan.

1. Lista medlemsnummer, för- och efternamn på alla medlemmar som är medlem i en sektion vars namn innehåller ordet is. Sortera listan efter medlemsnummer i stigande ordning.   
     
   Svar: (Jag la till några fler personer för att se om det här fungerade som det ska, vilket det gjorde)

CREATE VIEW finnsis

AS SELECT namn, beskrivning

FROM sektion

WHERE namn LIKE '%is%' OR namn LIKE '%Is%';

SELECT p.id AS medlemsnummer, p.förnamn, p.efternamn, k.id AS sektionsnummer

FROM person p, medlemssektion m, finnsis k

WHERE p.id = m.medlemsid

AND m.sektionsid = k.id

ORDER BY medlemsnummer ASC;



1. Lista id, för- och efternamn på alla personer som är en administratör men inte en medlem.  
     
   Svar:   
   SELECT A.id, A.förnamn, A.efternamn, B.id, C.id

FROM person A

LEFT OUTER JOIN medlem B ON (A.id = B.id)

LEFT OUTER JOIN administratör C ON (B.id != C.id)

WHERE B.id IS NULL

1. Lista för- och efternamn på alla medlemmar som inte betalt medlemsavgiften för år 2014.

Svar:  
SELECT A.id, A.förnamn, A.efternamn

FROM person A

LEFT OUTER JOIN avgift B ON (B.år = '2004')

LEFT OUTER JOIN medlemsavgift C ON (C.betald = 'false')