

# PRÀCTICA 3

**Autors:** Sergi Carol Bosch i Enric Lenard Uró

**Data:** 26/03/2015

## 2.1 ELS SENSORS

1. Write a query (using the SELECT statement) that will compute times and ids when any sensor's light reading was above 550. Show both the query and the first few lines of the result.

```
SELECT result_time, nodeid
FROM sensors
WHERE light>550;
```

```
2015-03-05 06:10:18|1
2015-03-05 08:20:10|1
2015-03-05 09:10:18|1
2015-03-05 10:20:10|1
2015-03-05 10:30:11|1
2015-03-05 11:33:15|1
2015-03-05 11:40:23|1
2015-03-05 12:42:28|1
2015-03-26 19:17:56|1
2015-03-26 19:17:56|1
2015-03-26 19:17:56|1
2015-03-05 06:00:00|2
2015-03-05 07:01:00|2
2015-03-05 10:30:11|2
2015-03-05 11:33:15|2
2015-03-05 11:40:23|2
2015-03-05 12:42:28|2
2015-03-26 19:17:57|2
2015-03-26 19:17:58|2
2015-03-26 19:17:58|2
2015-03-05 09:10:18|3
2015-03-05 10:20:10|3
2015-03-05 10:30:11|3
2015-03-05 11:33:15|3
2015-03-05 11:40:23|3
2015-03-05 12:42:28|3
2015-03-26 19:17:59|3
2015-03-26 19:17:59|3
2015-03-26 19:17:59|3
2015-03-05 09:10:18|1
2015-03-05 10:20:10|1
```

2. Write a query that will compute the average light reading at sensor 1 between 6 PM and 9 PM (inclusive of 6:00:00 PM and 9:00:00 PM). Show the query and the result.

```
SELECT avg(light)
FROM sensors
WHERE nodeid = 1 AND time(result_time) BETWEEN '06:00:00' AND '09:00:00';
```

555.5

3. Write a single query that computes the average temperature and light reading at every sensor between 6 PM and 9 PM, but exclude any sensors whose maximum voltage was greater than 418 during that time period. Show both the query and the result.

LIGHT

```
SELECT avg(light)
FROM sensors
WHERE (time(result_time) BETWEEN '06:00:01' AND '09:59:59')
EXCEPT SELECT avg(light)
FROM sensors
WHERE voltage > 418;
```

555.3333333333333

TEMPERATURE

```
SELECT avg(temp)
FROM sensors
WHERE (time(result_time) BETWEEN '18:00:01' AND '21:59:59')
EXCEPT SELECT avg(temp)
FROM sensors
WHERE voltage > 418;
```

16.6923076923077

4. Write a query that computes the average calibrated temperature readings from sensor 2 during each hour, inclusive, between 6 PM and 9 PM (i.e., your answer should consist of 4 rows of calibrated temperatures).

```
SELECT avg(temp)
FROM sensors
WHERE nodeid = 2 AND time(result_time) BETWEEN '18:00:00' AND '19:00:00' UNION
SELECT avg(temp)
FROM sensors
WHERE nodeid = 2 AND time(result_time) BETWEEN '19:00:00' AND '20:00:00' UNION
SELECT avg(temp)
FROM sensors
WHERE nodeid = 2 AND time(result_time) BETWEEN '20:00:00' AND '21:00:00';
```

10.0

15.0

30.0

5. Write a query that computes all the epochs during which the results from sensors 1 and 2 arrived more than 1 second apart. Show the query and the result. Note that you can use the difference (minus) operator on timestamps in Postgres, and that the string '1 second' refers to a period of 1 second.

```

SELECT sensor1.result_time, sensor2.result_time, sensor1.epoch, sensor2.epoch
FROM sensors as sensor1, sensors as sensor2
WHERE (sensor1.nodeid=1 AND sensor2.nodeid=2) AND
((strftime('%Y',sensor1.result_time)-strftime('%Y',sensor2.result_time)==0 AND
strftime('%m',sensor1.result_time)-strftime('%m',sensor2.result_time)==0 AND
strftime('%d',sensor1.result_time)-strftime('%d',sensor2.result_time)==0 AND
strftime('%H',sensor1.result_time)-strftime('%H',sensor2.result_time)==0 AND
strftime('%M',sensor1.result_time)-strftime('%M',sensor2.result_time)==0 AND
strftime('%S',sensor1.result_time)-strftime('%S',sensor2.result_time)>=1) OR
(strftime('%Y',sensor1.result_time)-strftime('%Y',sensor2.result_time)==0 AND
strftime('%m',sensor1.result_time)-strftime('%m',sensor2.result_time)==0 AND
strftime('%d',sensor1.result_time)-strftime('%d',sensor2.result_time)==0 AND
strftime('%H',sensor1.result_time)-strftime('%H',sensor2.result_time)==0 AND
strftime('%M',sensor1.result_time)-strftime('%M',sensor2.result_time)==0 AND
strftime('%S',sensor1.result_time)-strftime('%S',sensor2.result_time)<=-1)
)
AND sensor1.epoch=sensor2.epoch;

2015-03-26 19:17:56|2015-03-26 19:17:57|727|727
2015-03-26 19:17:56|2015-03-26 19:17:58|729|729

```

6. Write a query that determines epochs during which one or two of the sensors did not return results. Show your query and the first few results, sorted by epoch number. You may wish to use a nested query – that is, a SELECT statement within the FROM clause of another SELECT statement.

```

SELECT epoch
FROM sensors
GROUP BY epoch
HAVING COUNT(*) < 3
ORDER BY epoch;

732
734

```

## 2.2 LA MINI-XARXA SOCIAL

1. Obtenir les dades dels usuaris (excepte pwd) que viuen a Manresa.

```

SELECT email, nom, cognom, poblacio, dataNaixement
FROM usuaris
WHERE poblacio == 'Manresa';

enriclenard@gmail.com|Enric|Lenard|Manresa|1994
joanalbets@gmail.com|Joan|Albets|Manresa|2001
peregarcia@gmail.com|Pere|Garcia|Manresa|1990

```

2. Obtenir l'email dels usuaris amb cognom "Albets".

```

SELECT email
FROM usuaris
WHERE cognom = 'Albets';

```

joanalbets@gmail.com

3. Visualitzar els amics (nom i cognom) de l'usuari "Pere", "Garcia" (estat=Acceptada).

```
SELECT nom,cognom
FROM usuaris
WHERE email IN (SELECT email2
                FROM amistats
                WHERE email1='peregarcia@gmail.com'
                AND estat='Acceptada')
OR email IN (SELECT email1
            FROM amistats
            WHERE email2='peregarcia@gmail.com'
            AND estat = 'Acceptada');
```

Berta|Capdevila  
Enric|Lenard  
Anna|Vidella  
Sergi|Carol

4. Obtenir els amics de l'usaris "Pere" "Garcia" que no són amics de l'usuari "Jordi" "Alba".

```
SELECT nom, cognom, email
FROM usuaris
WHERE email IN
    (SELECT email2
     FROM amistats
     WHERE email1='peregarcia@gmail.com' AND estat='Acceptada')
OR email IN
    (SELECT email1
     FROM amistats
     WHERE email2='peregarcia@gmail.com'AND estat = 'Acceptada')
AND email IN
    (SELECT email1
     FROM amistats
     WHERE email2 = 'jordialba@gmail.com' AND estat != 'Acceptada')
OR email IN
    (SELECT email2
     FROM amistats
     WHERE email1 = 'jordialba@gmail.com' AND estat != 'Acceptada')
```

Berta|Capdevila|bertacapdabila@nose.com  
Enric|Lenard|enriclenard@gmail.com  
Anna|Vidella|annavidella@nose.com

5. Obtenir el nombre total de peticions d'amistat rebutjades.

```
SELECT COUNT()
FROM amistats
WHERE estat = 'Rebutjada';
```

6. Obtenir les dades (noms,cognoms) d'amics que viuen a Manresa.

```
SELECT nom,cognom
FROM usuaris
WHERE (poblacio='Manresa')
AND ((email IN (SELECT email1 FROM amistats WHERE ((estat='Acceptada')
AND (email2 IN (SELECT email FROM usuaris WHERE (poblacio='Manresa'))))))
OR (email IN (SELECT email2 FROM amistats WHERE ((estat = 'Acceptada')
AND (email1 IN (SELECT email FROM usuaris WHERE (poblacio = 'Manresa')))))));
```

|       |        |
|-------|--------|
| Pere  | Garcia |
| Enric | Lenard |

7. Obtenir, per cada usuari, el nombre de peticions rebutjades.

```
SELECT nom, count()
FROM amistats, usuaris
WHERE (estat='Rebutjada') AND (usuaris.email==amistats.email1)
GROUP BY nom;
```

|       |   |
|-------|---|
| Anna  | 1 |
| Berta | 1 |
| Jordi | 1 |
| Pere  | 3 |

8. Obtenir els usuaris que no són amics de "Ana", "Vilella".

```
SELECT nom, cognom
FROM usuaris
WHERE (email IN (
SELECT email1
FROM amistats
WHERE email2='annavidella@nose.com' AND estat != 'Acceptada')
OR email IN (
SELECT email2
FROM amistats
WHERE email1='annavidella@nose.com' AND estat !='Acceptada'));
```

|       |       |
|-------|-------|
| Jordi | Alba  |
| Sergi | Carol |