

Lab 4 Tasks: Solutions

Solutions Lab 4 Tasks

These are some proposed SQL SELECT statements. Surely, these are not unique; you can provide your own alternatives/versions of the same SQL query ☺.

SQL SELECT Statements 'Dog' Database

SQL 1:

```
SELECT *  
FROM owner  
WHERE phone like '141%' or phone IS NULL
```

SQL 2:

```
SELECT distinct(breedname)  
FROM dog  
ORDER BY breedname DESC
```

SQL 3:

```
SELECT dog.kennelname  
FROM dog  
WHERE dog.name = dog.mothersname  
OR dog.name = dog.fathersname  
OR dog.mothersname IS NULL
```

SQL 4:

```
SELECT Dog.name as Dog_Name, Breed.breedname as Breed_Name  
FROM Dog, Breed  
WHERE Dog.breedname = Breed.breedname  
ORDER BY Dog_Name
```

Or...*without* associating two relations...

```
SELECT Dog.name as Dog_Name, Dog.breedname as Breed_Name  
FROM Dog  
ORDER BY Dog_Name
```

SQL 5:

```
SELECT Attendance.showname AS Show, Owner.name AS Owner,  
Dog.name AS Dog  
FROM Attendance, Dog, Owner  
WHERE Attendance.dogid = Dog.dogid  
AND Owner.ownerid = Dog.ownerid
```

SQL 6:

```
SELECT Owner.phone AS TEL, Kennel.address AS address
FROM Attendance, Dog, Owner, Kennel
WHERE Attendance.dogid = Dog.dogid
AND Owner.ownerid = Dog.ownerid
AND Dog.kennelname = Kennel.kennelname
AND Attendance.place = 1
```

SQL 7:

```
SELECT distinct(O.name) AS Owner
FROM Owner O, Dog D, Kennel K
WHERE O.ownerid= D.ownerid
AND D.kennelname= K.kennelname
AND K.address IS NULL
```

Note: The idea is simply declaring 'IS NULL' to represent the concept of 'unspecified'.

SQL 8:**One approach:**

```
SELECT A.name
FROM dog A, dog B
WHERE B.ownerid= A.ownerid
AND B.name = 'Laser'
AND B.name <> A.name
```

...and another one as proposed by one of your colleagues ☺

```
SELECT D.NAME, O.OWNERID
FROM DOG AS D, OWNER AS O
WHERE D.OWNERID = O.OWNERID
AND D.NAME <> 'Laser'
AND O.OWNERID = (SELECT O.OWNERID
                  FROM DOG AS D, OWNER AS O
                  WHERE D.OWNERID = O.OWNERID
                  AND D.NAME = 'Laser');
```

Note 1: The idea is to get the names of the dogs with the same owner, such that for these dogs, there exists at least one dog with name Laser.

Note 2: Both approaches are correct! As I've mentioned in the talk, the inherent *complexity* of a SQL SELECT statement depends on our *way* of thinking. SQL is giving us the opportunity to explore different versions of SELECT statements resulting into the same outcome. Now, the DB system knows how to *optimally* execute the abovementioned queries,

and this cannot be *enforced* or *dictated* by SQL SELECT statements issued by end-users (only from those being inside the DB system ☺). The first approach is more declarative, that is, let the system to recursively associate the relation Dog over the criteria in the WHERE clause. The second approach is more procedural, that is, we describe some steps (like defining an algorithm therein) and tell the system to follow these steps, like, first execute the inner query and then execute the outer query. All in all, you decide which approach suits you best (and surely, there will be more!).

SQL 9:

```
SELECT A.name, B.name
FROM dog A, dog B
WHERE B.ownerid = A.ownerid
AND B.name < A.name
```

Note: The idea is to get the two names A and B of the pair (A, B) of dogs, which have the same owner. But, we display only once the names of each pair, i.e., either (A,B) or (B,A). We achieve that by simply stating for each pair (A,B) that the dog name A has to be smaller (lexicographically) than the dog name B.