WEEK 5

SQL – MORE JOIN EXAMPLES

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - Write queries that use aggregate functions such as SUM and MAX
  - Write queries that use EXIST, IN and NOT IN
  - Create queries that count groups of items

MORE ON NESTED SQL QUERIES

- The predicate part of the where clause can contain a nested SQL query, connected to the main query by things like =, <>, >= and then the words SOME, ANY or ALL

Query: Find the employee who makes the greatest salary.

Example:

```
mysql> SELECT name,salary FROM emp WHERE salary = (SELECT
  MAX(salary) FROM emp);
+-------------+--------+
| name        | salary |
+-------------+--------+
| Monty Burns | 100000 |
+-------------+--------+
1 row in set (0.00 sec)
```

DIFFERENT TYPES OF JOINS

- Consider the following tables:

Driver:

FirstName	LastName	DriverLic
Jane	Banks	W2
Mary	Poppins	E3
Hugh	Grant	R4

Car:

CarLic	Make	Model	DriverLic
ABC 123	Honda	Civic	E3
CCD 345	Honda	Civic	R3
EER 232	Toyota	Tercel	E3
DRS 345	Oldsmobile	LaSabre	X3

- Query: Find the last name of the people who drive our cars and the model name of the cars they drive:

SELECT LastName, Model FROM Driver D INNER JOIN Car AS C ON C.DriverLic = D.DriverLic;

QUESTION: What table will result?

- Also have the IN which connects the outer query with the inner one. Note that the thing before IN must have the same "shape" as the tuples that result from the nested query.
- Note that IN, =ANY and =SOME all produce the same answer.

Query: Find the last name of the people who drive one of our cars:

SELECT LastName FROM Driver WHERE DriverLic IN (SELECT DriverLic FROM Car)

QUESTION: What table will result?

- **Query: Find Employees who work in London-located departments:**

*SELECT * FROM Employee WHERE DeptNo IN (SELECT DeptNo FROM Department WHERE deptlocation = 'London')*

- **Query: Find the last name of people who don't drive our cars:**

SELECT LastName FROM Driver WHERE DriverLic NOT IN (SELECT DriverLic FROM Car)

QUESTION: What table will result?

QUESTION: Write the query to find the model names of cars that aren't driven by our drivers:

- **EXISTS:** Exists returns true if the result of the query after it is not empty. Here's an example using EXISTS, a correlated sub query which also uses a tuple variable defined in the outer query.

Query: Find all departments which have projects in a city that is not the same location of the department.

SELECT * FROM Department AS d WHERE EXISTS (SELECT * FROM Project WHERE d.location <> projectlocation AND d.deptno = Project.deptno)

- **Query: Find the last name of people who don't drive our cars:**

SELECT LastName FROM Driver WHERE DriverLic NOT IN (SELECT DriverLic FROM Car)

QUESTION: What table will result?

GROUP BY & HAVING SQL QUERIES

- Group by creates groups (equivalence classes) of tuples with equal values on the grouped by attribute(s).
- Having applies a predicate to the **whole** group.
- Once you have created the groups, you can apply aggregate functions to them. These are things like **Sum, Avg, Min, Max** and **Count.**

Using the following table:

PET:

Name	Type	Weight	Category
Poodle	Dog	35	DG
Calico	Cat	14	CA
Siamese	Cat	13	CA
Lab	Dog	50	DG
Budgie	Bird	1	BI
Canary	Bird	2	BI
German Shepard	Dog	65	DG
Boxer	Dog	34	DG

QUESTION: What would the following query return?

SELECT Type, Weight FROM pet

```
mysql> SELECT type, weight FROM pet;
+------+--------+
| type | weight |
+------+--------+
| Dog  |     35 |
| Dog  |     65 |
| Dog  |     34 |
| Dog  |     50 |
| Cat  |     14 |
| Cat  |     13 |
| Bird |      1 |
| Bird |      2 |
+------+--------+
```

QUESTION: What do you think the following query will return?

SELECT Type FROM pet GROUP BY type

```
mysql> SELECT type FROM pet GROUP BY type
    -> ;
+------+
| type |
+------+
| Bird |
| Cat  |
| Dog  |
+------+
3 rows in set (0.00 sec)
```

QUESTION: What about:

SELECT Type, SUM(Weight) FROM pet GROUP BY type

```
mysql> SELECT type, SUM(weight) FROM pet GROUP BY type
    -> ;
+------+-------------+
| type | sum(weight) |
+------+-------------+
| Bird |           3 |
| Cat  |          27 |
| Dog  |         184 |
+------+-------------+
3 rows in set (0.00 sec)
```

NOTE:

- *SELECT type, weight FROM pet GROUP BY type* → THIS WON'T DO WHAT YOU EXPECT!!

- *SELECT type, weight FROM pet GROUP BY type, weight* → THIS IS OKAY

- *SELECT type, name FROM pet GROUP BY type, name* gives:

Type	Name
Dog	Poodle
Cat	Calico
Cat	Siamese
Dog	Lab
Bird	Budgie
Bird	Canary
Dog	German Shepard
Dog	Boxer

- *SELECT type, COUNT(*) AS 'Num Of Pets' FROM pet GROUP BY type HAVING SUM(weight) < 40*

gives:

Type	Num of Pets
Bird	2
Cat	2