

## Assignment 1: SQL Queries

Due on January 24, 2010, at 9:10 am, in class.

This is an *individual* assignment. You can *discuss* each part of the assignment with other students orally prior to writing it up, subject to the following conditions:

- You should specify on your assignment with whom you discussed it. (E.g., “Discussed with Obi-Wan Kenobi and Darth Vader.”)
- You should not take any written notes while discussing the assignment with other students. In particular, you should not be writing up your assignment while discussing it with others.
- The queries, the results, and the explanations included in your assignment should be *your own*, produced based on your memory, your class notes, the book and the standard reference materials (e.g., the MySQL Manual), and not, for example, on emails exchanged with your fellow students.

Collaboration beyond what is described above will be considered an academic violation.

Each exercise is worth 10 points, for a total of 150 (15% of your course grade). Your answers to each of the exercises 1-14 should include three elements:

### 1. The query

Your answer should include the exact SQL query that you used to answer the question. Each question should be answered using just **one** query. For full credit, your query should be designed so that it would produce a table that contains **only the information needed to answer the question** and nothing else. For example, if you are asked to find out who owns the youngest pet, your query should produce a table with just one cell (not counting the header row) showing the name of the person who owns the youngest pet. It should **not** show the name of the pet or its age. Even more importantly, it should **not** show information about any other pets. On the other hand, if you are asked who are the *names* and *owners* of the *two* youngest pets, then your query should produce a table with *two* rows, one for the youngest pet and one for the second youngest pet, with both rows showing the name of the pet and the name the pet’s owners. *If you cannot produce a table showing just the required information, you should turn in a query that produces the most concise table that you can get, for partial credit.*

### 2. An explanation of the query

Your answer should include a brief (30-100 words) explanation, in your own words, of what your query does. See the example below.

### 3. The result of the query and the answer to the question

Your answer should include the exact table returned by MySQL (copy and paste it). Also, unless the result provides an immediate and direct answer to the question, make sure to say what you think the answer is. *This part is not necessary for problems in part C where you are asked to write queries against an imaginary database.*

For example, suppose you are asked: “Animals of which species have a total weight of less than 200 grams?” Your answer may look like this:

```
select species from pet group by species having sum(weight)<0.2;
```

I am asking MySQL to group the rows by species, giving me a results table with one column — the names of the species, including in this table only those species for which the sum of weights of all pets is less than 0.2.

```
+-----+
| species |
+-----+
| bird    |
| hamster |
+-----+
2 rows in set (0.00 sec)
```

The two species with a total weight of less than 200 grams are birds and hamsters.

## Part A: Menagerie

Use the “menagerie” database on the database server for exercises 1 and 2.

### Exercise 1

Who owns the heaviest dog?

### Exercise 2

What is the total weight of dogs owned by Harold?

## Part B: Child Care

Use the “child\_care” database on the database server for exercises 1-6. This database lists 929 child-care centres in Toronto as of September 2010. (This is real data from Toronto's Open Data project.)

### Exercise 3

List the phone numbers of the centres that provide a fee subsidy and have more than 20 infant spaces, ordered by the number of infant spaces?

### Exercise 4

How many centres have a name that includes the word “Center” (with “er” rather than “re” at the end)?

### Exercise 5

What are the top ten wards by the total number of toddler spaces? (List the ward numbers together with the total number of toddler spaces in each of them.)

### Exercise 6

In which wards do toddler spaces available in centres run by commercial agencies account for more than a quarter of all spaces in such centres? List the ward numbers sorted by the ratio of toddler spaces to all spaces in commercial centers, in descending order.

## Part C: Flying Blind

Suppose we have a database of cities, with a table “cities” which has the following fields:

Field	Comment
<b>name</b>	the name of the city (text)
<b>population</b>	the population of the city
<b>country_code</b>	the two-letter “ISO 3166-1 alpha-2” code for the country in which the city is located
<b>elevation</b>	elevation above sea level, in meters

Since you are not given the actual database, you cannot test your query or get real results. So, answer questions 7 and 8 just by providing a query that would answer the question and explaining how it would do this.

### Exercise 7

Which country has the largest number of cities with a population of more than a million people? (Your query should return just the country code.)

### Exercise 8

What is the average elevation of cities in Canada with names starting with “T”?

## Part D: Monkeys

Your personal database on the database server has a table called “monkeys,” which has the following fields:

Field	Comment
<b>name</b>	monkey’s name (text, up to 50 letters)
<b>size</b>	monkey's size in cm
<b>bananas</b>	number of bananas in monkey's possession (an integer)
<b>mood</b>	“H” if the monkey is happy, “S” if it is sad, NULL if the mood is unknown

Note: If at some point you want to delete all data in your database and start again, you can use “delete from monkeys”. This will delete all the records without deleting the database itself.

### Exercise 9

Add five monkeys to your database. Each monkey needs to have a name and size. Each monkey should have one banana. Some monkeys should be happy and some should be sad.

### Exercise 10

Let's give one additional banana to each **sad** monkey and record the resulting number of bananas. (The new number of bananas for each sad monkey should be set to the number of bananas the monkey currently has plus 1. Assume that someone else might have modified the database, so the monkeys may now have a different number of bananas from what you originally recorded.)

## Part E: Diveshop

For questions 11-15 use the “diveshop” database on the database server. Use “join... on...” syntax.

### Exercise 11

What are the names of the customers who ordered vacations costing over \$20,000. (Join DIVECUST and DIVEORDS tables. Do **not** add up the cost of different vacations by the same customer.)

## Exercise 12

What are the two most visited destinations? (If a customer booked multiple trips to the same destination, count all of them separately.)

## Exercise 13

What is the most popular payment method for customers living **outside** California ("CA")?

## Exercise 14

What is the winter temperature in C° at the place where the customer from Michigan ("MI") booked his vacation? (You will need to join DIVECUST, DIVEORDS and DEST tables.)

## Exercise 15

What is the question that the following query aims to answer and how is it doing it?

*Query 1:*

```
select Destination_Name
from BIOLIFE
join BIOSITE on BIOLIFE.Species_No=BIOSITE.Species_No
join SITES on BIOSITE.Site_No=SITES.Site_No
join DEST on SITES.Destination_No=DEST.Destination_No
where Common_Name="Blue Angelfish"
and Skill_Level="Beginning"
and Travel_Cost < 3000;
```

How is query 1 different from queries 2, 3 and 4 below? Which of those queries is best and why?

*Query 2:*

```
select Destination_Name
from BIOLIFE
join BIOSITE using (Species_No)
join SITES using (Site_No)
join DEST using (Destination_No)
where Common_Name="Blue Angelfish"
and Skill_Level="Beginning"
and Travel_Cost < 3000;
```

*Query 3:*

```
select Destination_Name
from BIOLIFE
natural join BIOSITE
natural join SITES
natural join DEST
where Common_Name="Blue Angelfish"
and Skill_Level="Beginning"
and Travel_Cost < 3000;
```

*Query 4:*

```
select Destination_Name
from BIOLIFE
join BIOSITE
join SITES
join DEST
where Common_Name="Blue Angelfish"
and Skill_Level="Beginning"
and Travel_Cost < 3000;
```