Week 1 SQL Tutorial

### Approaches to the Tutorial

There are two approaches to the Weekly SQL Tutorials. The first is to go through them step-by-step on your own. This approach is recommended and will be central to your success in the SQL portion of the course. The second is to use the tutorial as a quick reference, for example, when completing your SQL homework or the term project. Most of the homework questions correspond closely to examples covered here in the SQL tutorial and the SQL tutorials in subsequent weeks. I expect that you will find the tutorial useful for this purpose when completing the SQL HW as well.

#### ***Setting up the Sample Database for the Tutorial***

Launch your MySQL/Turnkey Linux virtual machine per the “Installing MySQL” instructions under course home

### Part 1: Accessing Meta-Data (data about the data that is stored in the DBMS)

To show the databases (which act basically as directories) available in your MySQL installation:

SHOW DATABASES;

To select the ‘employees’ database that this tutorial is based on:

USE employees;

To output the tables that the ‘employees’ database contains:

SHOW TABLES;

To output the attributes the ‘employee’ table in the ‘employees’ database contains:

DESCRIBE employee;

### Part 2: Outputting data from individual tables:

Examples of Simple SELECT Queries

Output the current date and time:

SELECT NOW();

Output all of the attributes for all of the records in the ‘employee’ table:

SELECT \* FROM employee;

Output the gender attribute from the employees table, removing all duplicate results by using the ‘DISTINCT’ keyword:

SELECT DISTINCT gender FROM employee;

Output the ‘first\_name’, ‘last\_name’, ‘gender’ and ‘birth\_date’ attributes for all of the records in the ‘employee’ table:

SELECT first\_name, last\_name, gender, birth\_date FROM employee;

#### ***Filtering Records***

Output the ‘first\_name’ and ‘last\_name’ attributes for only those records in the ‘employee’ table where the ‘birth\_date’ attribute is greater than (i.e. later than) December 31st 1974:

SELECT first\_name, last\_name FROM employee WHERE birth\_date > ‘1974-12-31’;

Output all attributes for those records in the ‘employee’ table where the ‘gender’ attribute is NOT equal to ‘M’:

SELECT \* FROM employees WHERE gender != ‘M’;

Output all attributes from the ‘title’ database where the ‘to\_date’ attribute is null (i.e. missing / unknown):

SELECT \* FROM title WHERE to\_date IS NULL;

Output all attributes from the ‘title’ database where the ‘to\_date’ attribute is present (i.e. not null):

SELECT \* FROM title WHERE to\_date IS NOT NULL;

Output all of the attributes from the ‘title’ table where the ‘title’ attribute is in the following list: 'Assistant Engineer', 'Senior Engineer', or 'Staff Engineer'.

SELECT \* FROM title WHERE title IN ('Assistant Engineer', 'Senior Engineer', 'Staff Engineer');

Output all of the attributes from the records in the salary table where the salary attribute is less than 130,000 AND the salary attribute is greater than 120,000:

Option 1:

SELECT \* FROM salary WHERE (salary < 130000 AND salary > 120000);

Option 2:

SELECT \* FROM salary WHERE salary BETWEEN 120000 AND 130000;

Output all of the attributes from the ‘employee’ table for those employees who were born in 1964:

SELECT \* FROM employee WHERE birth\_date BETWEEN '1964-01-01' AND '1964-12-12';

Output all of the attributes from the records in the salary table where EITHER the salary attribute is greater than 140,000 OR the salary attribute is less than 40,000:

SELECT \* FROM salary WHERE salary > 140000 OR salary < 40000;

#### ***Differences in String Data versus Numeric Data***

Note that when dealing with salary in the WHERE clause above, we do not use quotation marks, while in dealing with dates, like birth\_date, and titles, we surround values with quotation marks. That is because salary is a *numeric* data type, while title is *string* data.

Birth\_date is a bit of a special case, because is a a *date* data type, but dates need to be surrounded by quotes. We’ll discuss data types in further detail later in the course.

Note further that either single quotes – ‘string’ – or double – “string” – will work in MySQL, but for maximum interoperability in DBMSs other than MySQL, it’s a good idea to get in the habit of using single quotes.

#### ***Ordering Output***

Output all employees ordered from the oldest to the youngest:

SELECT \* FROM employee ORDER BY birth\_date;

Output all employees ordered alphabetically, first by last name, then by first:

SELECT \* FROM employee ORDER BY last\_name, first\_name;

List all of the employees’ current salaries from the highest to the lowest:

SELECT \* FROM salary WHERE to\_date IS NULL ORDER BY salary DESC;

Note that the default ordering is ascending and if you want descending, you must specify DESC. Go back and review the ordering oldest to youngest above. Why did that work without specifying DESC?