

## CSC 455: Database Processing for Large-Scale Analytics

### Assignment 3

**Due 11:00pm, Sunday, October 18<sup>th</sup>.**

**Supplemental reading:** SQL reference book *Oracle 11g SQL* by Price, ISBN 9780071498500 (available in Books 24x7 DePaul online library as eBook). Sections 2.16-2.21s, 6.1, 6.2 (section names are included in the screenshot)

### Part 1

In this and the next part we will use an extended version of the schema from Assignment 2. You can find it in a file ZooDatabase.sql posted with this assignment on D2L.

Once again, it is up to you to write the SQL queries to answer the following questions:

1. List the animals (animal names) and the ID of the zoo keeper assigned to them.  
**SELECT Animal.Aname, Handles.ZOOKEEPID  
FROM Animal  
INNER JOIN Handles  
ON Animal.AID = Handles.AnimalID;**
2. Now repeat the previous query and make sure that the animals without a handler also appear in the answer.  
**SELECT Animal.ANAME, Handles.ZOOKEEPID  
FROM Animal  
FULL OUTER JOIN Handles  
ON AID = Handles.ANIMALID;**
3. Report, for every zoo keeper name, the total number of hours they spend feeding all animals in their care.  
**SELECT Animal.ANAME, Zookeeper.Zname,  
(Zookeeper.HourlyRate\*Animal.TIMETOFEED)  
FROM Animal  
INNER JOIN Handles  
JOIN Zookeeper ON Handles.ZOOKEEPID = Zookeeper.ZID  
ON Animal.AID = Handles.ANIMALID;**
4. Report every handling assignment (as a list of assignment date, zoo keeper name and animal name). Sort the result of the query by the assignment date in an ascending order.  
**SELECT Handles.Assigned, Zookeeper.ZName, Animal.AName  
FROM Animal  
INNER JOIN Handles ON AID = Handles.ANIMALID  
JOIN Zookeeper ON Handles.ZOOKEEPID = Zookeeper.ZID  
ORDER BY Handles.Assigned ASC;**

5. Find the names of animals that have at least 1 zoo keeper assigned to them.

```
SELECT Animal.AName, Zookeeper.ZNAME  
FROM Animal  
INNER JOIN Handles ON Animal.AID = Handles.ANIMALID  
JOIN Zookeeper ON Handles.ZOOKEEPID = Zookeeper.ZID;
```

6. Find the names of animals that have 0 or 1 (i.e., less than 2) zoo keepers assigned to them.

```
SELECT DISTINCT Handles.ANIMALID, Animal.ANAME  
FROM Animal  
FULL OUTER JOIN Handles ON AID = Handles.ANIMALID;
```

- Optional query:

List all combination of animals where the difference between feeding time requirement is within 0.25 hours (e.g., Grizzly bear, 3, Bengal tiger, 2.75). Hint: this will require a self-join. Avoid listing identical pairs such as (Grizzly bear, 3, Grizzly bear, 3)

## Part 2

- A. Write a python script that is going to read the queries that you have created in Part-1 from a SQL file, execute each SQL query against SQLite3 database and print the output of that query.
- a. You must read your SQL queries from a file, please do not copy SQL directly into python code. The code that would run commands from the ZooDatabase.sql file is provided (runSQL.py), so all you have to do is to change it so that it reads *your* queries from a SQL file and prints the output of your queries.
  - b. You can refer to example code from the previous assignment that prints query results using fetchall(). You do not have to format the output in any particular fashion – however, you must print every row individually using a loop.

CODE Submission Attached runSQL.py

- B. Repeat the work you did in Part-2 of the previous homework using the data file Public\_Chauffeurs\_Short\_hw3.csv attached in D2L in this assignment dropbox. It contains roughly the same data, with two changes: NULL may now be represented by NULL or an empty string (,NULL, or ,,) and some of the names have the following form “Last, First” instead of “First Last”, which is problematic because when you split the string on a comma, you end up with too many values to insert.

CODE Submission sqlite3\_HW3\_Chauffuers.py

### Part 3

Using the company.sql database (posted in with this assignment), write the following SQL queries.

1. Find the names of all employees who are directly supervised by 'Franklin T Wong'.

```
SELECT EMPLOYEE.Fname || ' ' || EMPLOYEE.LNAME AS "Full Name"  
FROM EMPLOYEE  
WHERE EMPLOYEE.Super_ssn = '333445555';
```

2. For each project, list the project name, project number, and the total hours per week (by all employees) spent on that project.

```
SELECT PROJECT.PName , PROJECT.PNUMBER, SUM(WORKS_ON.HOURS)  
FROM PROJECT, WORKS_ON  
WHERE PROJECT.PNUMBER = WORKS_ON.PNO  
GROUP BY PROJECT.PNAME, PROJECT.PNUMBER;
```

3. For each department, retrieve the department name and the average salary of all employees working in that department. Order the output by department number in ascending order.

```
SELECT DNAME, AVG(Salary)  
FROM DEPARTMENT, EMPLOYEE  
WHERE DEPARTMENT.DNUMBER = EMPLOYEE.DNO  
group by DNAME, DNUMBER order by DEPARTMENT.DNUMBER ASC;
```

4. Retrieve the average salary of all female employees.

```
SELECT AVG(SALARY) FROM EMPLOYEE WHERE SEX = 'F';
```

5. For each department whose average salary is greater than \$43,000, retrieve the department name and the number of employees in that department.

```
SELECT DNAME, COUNT(SSN)  
FROM EMPLOYEE, DEPARTMENT  
WHERE DNUMBER = DNO  
AND SALARY > 43000  
group by DNAME;
```

6. Retrieve the names of employees whose salary is within \$22,000 of the salary of the employee who is paid the most in the company (e.g., if the highest salary in the company is \$82,000, retrieve the names of all employees that make at least \$60,000.).

```
SELECT EMPLOYEE.Fname || ' ' || EMPLOYEE.LNAME AS "Full_Name"  
FROM Employee  
WHERE Salary >= (SELECT MAX(Salary) - 22000 from employee);
```

Be sure that your name and "Assignment 3" appear at the top of your submitted file.