

Olusola Ogun

Assignment #4

## Section 1

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

- Add yourself as a new record in the “person” table. Use your first name and last name for one of the new records that you are inserting.
  - **INSERT INTO Person (person\_id, first\_name, last\_name) VALUES (7, 'Olusola', 'Ogun');**
- Alter the Person table by adding a new property of your choice with the data type of your choice.
  - **ALTER TABLE Person ADD city VARCHAR(50)**
- Update your record in the Person table by setting some value to your new property.
  - **UPDATE person SET city = 'Lisle' WHERE person\_id = 7**
- Delete the Diana Taurasi record
  - **DELETE FROM person WHERE person\_id = 6**
- Alter the Contact List table and add column for Favorite
  - **ALTER TABLE contact\_list ADD favourite VARCHAR(1) DEFAULT NULL**
- Update the existing records in the “contact\_list” table to use the new column that you created. Update the records in the table by setting Michael Phelps as everyone's favorite contact (contact\_id = 1). The value for the “favorite” column should be set to “y” for these records.
  - **UPDATE contact\_list SET favourite = 'y' WHERE contact\_id = 1**
- Update the existing records in the “contact\_list” table to use the new column that you created. Update the remaining record(s) in the table by setting every contact who is NOT Michael Phelps (contact\_id <> 1) to not be a favorite. The value for the “favorite” column should be set to “n” for these records
  - **UPDATE contact\_list SET favourite = 'n' WHERE contact\_id <> 1**
- Insert at least 3 new records in the “contact\_list” table. Note: You are required to add at least 3 new records with yourself as a new contact in the “contact\_list” table. Make sure that you provide a value (y or n) for the new “favorite” column.
  - **INSERT INTO contact\_list (connection\_id, person\_id, contact\_id) VALUES (15, 7, 1, 'y')**
  - **INSERT INTO contact\_list (connection\_id, person\_id, contact\_id) VALUES (16, 7, 2, 'y')**
  - **INSERT INTO contact\_list (connection\_id, person\_id, contact\_id) VALUES (17, 7, 3, 'n')**
- Create a new table image with columns (image\_id, image\_name, image\_location). Image\_id is the primary key
  - **CREATE TABLE IMAGE (image\_id NUMBER NOT NULL, image\_name VARCHAR(25) NOT NULL, image\_location VARCHAR(100) NOT NULL, PRIMARY KEY (image\_id))**
- Create a message-image table with columns (image\_id,message\_id). Image\_id and message\_id are the primary keys
  - **CREATE TABLE message - IMAGE (image\_id NUMBER NOT NULL, message\_id NUMBER NOT NULL, PRIMARY KEY (image\_id, message\_id), FOREIGN KEY (image\_id) REFERENCES IMAGE (image\_id))**
- Insert 5 new records in the “image” table
  - **INSERT INTO IMAGE (image\_id, image\_name, image\_location) VALUES (1, 'img1', 'location1')**

- **INSERT INTO IMAGE (image\_id, image\_name, image\_location) VALUES (2, 'img2', 'location2')**
- **INSERT INTO IMAGE (image\_id, image\_name, image\_location) VALUES (3, 'img3', 'location3')**
- **INSERT INTO IMAGE (image\_id, image\_name, image\_location) VALUES (4, 'img4', 'location4')**
- **INSERT INTO IMAGE (image\_id, image\_name, image\_location) VALUES (5, 'img5', 'location5')**
- Insert 5 new records in the “message-image” table
  - **INSERT INTO message - IMAGE (image\_id, message\_id) VALUES (1, 1)**
  - **INSERT INTO message - IMAGE (image\_id, message\_id)VALUES (2, 2)**
  - **INSERT INTO message - IMAGE (image\_id, message\_id)VALUES (3, 3)**
  - **INSERT INTO message - IMAGE (image\_id, message\_id)VALUES (4, 4)**
  - **INSERT INTO message - IMAGE (image\_id, message\_id)VALUES (5, 5)**
- Find All of the Messages that Michael Phelps Sent. Display the following columns
  - **select message\_id, sender\_id, receiver\_id , message, send\_datetime from message where sender\_id in (select person\_id from person where first\_name like 'Michael' and last\_name like 'Phelps')**
- Find the Number of Messages Sent for Every Person. Display the following columns
  - **SELECT count(message.message\_id), person.person\_id, person.first\_name, person.last\_name FROM message INNER JOIN person ON message.reciever\_id = person.person\_id GROUP BY message.reciever\_id**
- Find All of the Messages that Have At Least One Image Attached Using INNER JOINS. Display the following columns
  - **select message\_id,image\_name,image\_location,message,send\_datetime from image inner join message\_image on image.image\_id= message\_image.image\_id inner join**

## Section 2

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

- Find the names of all employees who are directly supervised by 'Franklin T Wong'
  - **SELECT fname, lname FROM employee WHERE super\_ssn = 333445555**
- For each project, list the project name, project number, and the total hours per week (by all employees) spent on that project
  - **SELECT Pname, SUM(Hours) FROM PROJECT, WORKS\_ON WHERE PROJECT.Pnumber=WORKS\_ON.Pno GROUP BY PROJECT.Pnumber**
- 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 EMPLOYEE,DEPARTMENT WHERE Dno=Dnumber GROUP BY Dname ORDER BY DEPARTMENT**
- Retrieve the average salary of all female employees
  - **SELECT AVG (Salary) FROM EMPLOYEE WHERE Sex='F'**
- For each department whose average salary is greater than 44,000, retrieve the department name and the number of employees in that department
  - **SELECT DNAME, EMPLOYEECOUNT FROM DEPARTMENT D, (SELECT DNO, COUNT(SSN) AS EMPLOYEECOUNT FROM EMPLOYEE GROUPBY DNO HAVING AVG(SALARY) > 44000) E WHERE D.DNUMBER = E.DNO**
- Retrieve the names of employees whose salary is within 23,000 of the salary of the employee who is paid the most in the company (e.g., if the highest salary in the company is \$83,000, retrieve the names of all employees that make at least 60,000.)
  - **SELECT FNAME, MINIT, LNAME, FROM EMPLOYEE, WHERE SALARY > (SELECT MAX(SALARY) - 23000 FROM EMPLOYEE) AND SSN != (SELECT SSN FROM EMPLOYEE WHERE SALARY = (SELECT MAX(SALARY) FROM EMPLOYEE))**

In []: ,