

Assignment 1: Query Implementation using Relational Database

(Issue: 11 May, 2018 , Due: 18 May, 2018 11:59 PM)

- **TA:** Trishla Shah trishla@dal.ca
- **Tutorial:** Monday (10:35-11:25 am) and Thursday (2:05-2:55 pm)
- **Room:** CS 143

1. Objectives:

- 1) To review conventional Relational Database Systems
- 2) To learn difference between local vs cloud-based systems
- 3) To learn how to write RDBMS queries

2. Tasks:

- 1) Setup any local system (for example, install MySQL in your local system) and cloud-based system (for example, IBM DB2).
- 2) Create 7 tables as shown below and load data into the tables from the given data .csv files. (Note: Choose appropriate data types and size for the columns)

Database Description:

Database/Schema Name: *samplecompany*

Table Name	.csv file	Description
customer	customer.csv	It contains company's customer information
employee_addresses	employee_addresses.csv	It contains employee's address
employee_donations	employee_donations.csv	It contains employee's contribution/donations.
employee_payroll	employee_payroll.csv	It contains payroll information for all company employees
order_fact	order_fact.csv	It contains order information
product_dim	product_dim.csv	It contains information related to products provided by suppliers
staff	staff.csv	It contains confidential employee data for both existing and past employees

- 3) Perform following queries on both the systems (local and cloud). Also, measure the execution time:

Question 1:

Goal: Calculating a New Column

Tables: employee_payroll

Query: Write a query that displays the *Employee_ID*, *Employee_Gender*, *Marital_Status*, *Salary* and *Tax* Columns. Calculate the *Tax* column as one-third of the employee's salary.

Question 2:

Goal: Use the CREATE TABLE Statement

Tables: employee_payroll

Query: Write a query to create the bonus table. The new table should contain the following columns: *Employee_ID*, *Salary*, and a calculated column named *Bonus*. The *Bonus* column should represent a 4% bonus.

Question 3:

Goal: Eliminating Duplicates

Tables: employee_addresses

Query: Write a query that displays the cities where employees reside. The result should display unique row per city. Only *city* column should be present in the output.

Question 4:

Goal: Calculating a New Column

Tables: employee_donations

Query: Write a query that display data about employee contribution that exceeded \$90. The result should have the following characteristics:

- Display the columns *Employee_ID*, *Recipients* and the new column *Total* that represents the total charitable contribution for each employee over the four quarters
- Includes only employees whose charitable contribution that for all four quarters exceeds \$90.

Hint: The total amount of the employee's contribution is the sum of Qtr1, Qtr2, Qtr3 and Qtr4.

Question 5:

Goal: Advanced Summarizing Data in Groups

Tables: employee_addresses

Query: Write a query that displays the *countries* and *cities* where employees reside and *the number of employees in each city*. Include only one row per county/city combination. Display the values in country/city order.

Question 6:

Goal: Summarizing Data in Groups

Tables: Customer

Query: Write a query that displays the following statistics for each *country*:

- *Total number of customers*
- *Total number of male customers*
- *Total number of female customers*
- *Percent of all customers that are male (Percent Male).*

Display the result by value of *Percent Male* so that the country with the lowest value is listed first, with the remaining countries following in ascending order.

Question 7:

Goal: Create a result by combining two tables.

Tables: product_dim, order_fact

Query: Create a result by combining two tables.

- Include columns *Product_ID*, *Product_Name* from *product_dim* table.
- Include a column with the label *Total Sold*. Use a summary function to create this column, which displays the quantity sold for each product.
- Specifies the tables *product_dim*, with the alias *p* and *order_fact* with the alias *o*.
- Join the tables by matching the values of the appropriate columns in each table.
- Groups the results by *Product_ID* from *product_dim* table and *Product_Name*.
- Orders the rows so that products with the highest number sold appear at the top of the report and then by *Product_Name*.

Note: Do not use nested queries.

Question 8:

Goal: Create a result by combining three tables

Tables: *product_dim*, *order_fact*, *customer*

Query: Create a result by combining three tables.

- Include a column labeled *Name*, based on the column *Customer_Name*, as the first column.
- Include a column labeled *Purchases* as the second column. Use a summary function to create this column which displays the number of purchases for each customer.
- Specifies the tables *product_dim*, with the alias *p*; *order_fact* with alias *o*; and *customer* with the alias *c*.
- Join the tables by matching the values of the appropriate columns in each table.
- Select only internet orders. Internet orders are those that have the value 99999999 for *Employee_ID*.
- Selects only products in which the supplier's country is different from the customer's country.
- Select only customers who live in either the United States (US) or Australia (AU).
- Orders the row so that customers with the highest number of purchases appear at the top of the report and customers who have the same number of purchases are displayed alphabetical order.

Note: Do not use nested queries.

Question 9:

Goal: Create a result with a self-join.

Tables: employee_addresses, staff

Query: Display result of all trainees and workers at company. For each trainee or temporary worker, the report should include the *employee ID, name* and *job title*, and *manager ID* and *name*. The report should be ordered by *Employee_ID*.

Note (for all queries): In query results, include only necessary columns which are mentioned in the query specification.

- 4) Compare local system and Cloud-based system which you choose (based on functionality, execution time, flexibility, etc.).

3. Submit your Assignment 1 electronically:

- 1) Please use Brightspace to submit your assignment
- 2) In report, give brief explanation of each query along with the execution time (for both – local and cloud-based system).
- 3) Write comparison in the report.
- 4) Include all necessary screenshots in the report.
- 5) Mention structure of each table in the report. (for example, column name, data type, size/length, etc.)
- 6) In addition to the report, please also submit queries. (query1.sql, query2.sql,..., query9.sql). Do not include any comments or other things in .sql files.
- 7) Submit all items within one Zip file (Your_NAME_Ass1.zip).

*** Plagiarism and Intellectual Honesty:** (<http://plagiarism.dal.ca>)

Dalhousie University defines "plagiarism as the presentation of the work of another author in such a way as to give one's reader reason to think it to be one's own." Plagiarism is considered a serious academic offense which may lead to loss of credit, suspension or expulsion from the University, or even the revocation of a degree.