**Marks:** 100

# Introduction

This project tests proficiency in creating database tables and inserting data, imposing various constraints, and running variety of queries.

It is an individual assignment to be completed during scheduled class time. You are not supposed to communicate with people other than your instructor throughout duration of this test.

Points are indicated in the marking rubric at the end of this document.

# Part 1: Create Database Tables and Insert Data

Use Notepad or WordPad to prepare a script named *create\_Project.sql* that creates database Project and tables according to provided design specification and inserts provided data. The script should start with the following statements and contain all CREATE/ALTER TABLE and INSERT commands in the appropriate order (it should run without errors).

USE master

GO

IF DB\_ID('Project') IS NOT NULL

DROP DATABASE Project

CREATE DATABASE Project

GO

USE Project

GO

Run your script, and then execute queries that select all data from each table. **Paste here results of these queries to show that the tables were correctly created and filled with data.** You may just take a screenshot of the results grid from SSMS.

Graphical user interface, application, Word

Description automatically generated

  
ERD of the created Project database ↑

Write a query that selects data from all tables to produce a report identical to the sample data values table that was provided for insert. **Paste both the code of your query AND query results here.**

A screenshot of a computer

Description automatically generated with medium confidence

## Design Specification



Detailed requirements:

* JobID, EmployeeID and ProjectID are auto generated integer primary keys. JobID and ProjectID start with 1, and EmployeeID starts with 100; all are incremented by 1.
* All names (JobName, ProjectName, EmployeeFName, and EmployeeLName have type varchar(50) and are required.
* JobRate is money, required, and must be greater than zero.
* HoursWorked is an integer with default value 0.
* ProjectStartDate is a date (with time) value that allows null.

## Sample Data Values to Insert

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EmpNo** | **EmpFName** | **EmpLName** | **ProjectName** | **ProjStartDate** | **Job** | **JobRate** | **Hours** |
| 100 | Bob | Miller | Database1 | 20-10-2019 | Project Manager | $ 45.00 | 50 |
| 100 | Bob | Miller | Website2 | 10-02-2020 | Project Manager | $ 45.00 | 75 |
| 101 | Tara | Johnson | Website1 | 10-12-2019 | Project Manager | $ 45.00 | 85 |
| 102 | Richard | Sommers | Database1 | 20-10-2019 | Business Analyst | $ 34.50 | 45 |
| 103 | Rebecca | Brown | Database1 | 20-10-2019 | Software Developer | $ 37.50 | 85 |
| 103 | Rebecca | Brown | Website1 | 10-12-2019 | Software Developer | $ 37.50 | 38 |
| 104 | Tom | Walsh | Database1 | 20-10-2019 | Database Administrator | $ 47.50 | 95 |
| 105 | Carl | Logan | Website1 | 10-12-2019 | Software Developer | $ 37.50 | 75 |
| 105 | Carl | Logan | Website2 | 10-02-2020 | Software Developer | $ 37.50 | 58 |
| 106 | Natasha | Wong | Website1 | 10-12-2019 | Web Designer | $ 33.50 | 45 |
| 106 | Natasha | Wong | Website2 | 10-02-2020 | Web Designer | $ 33.50 | 39 |
| 107 | Randy | Clark | Website1 | 10-12-2019 | Software Developer | $ 37.50 | 93 |
| 107 | Randy | Clark | Website2 | 10-02-2020 | Software Developer | $ 37.50 | 64 |
| 108 | Ted | Colton | Database1 | 20-10-2019 | Software Developer | $ 37.50 | 45 |
| 108 | Ted | Colton | Website1 | 10-12-2019 | Software Developer | $ 37.50 | 35 |

Meaning of data:

* EmpNo – employee number; auto-generated starting from 100 and incrementing by 1 (it is OK if you incidentally get different EmpNo values than shown here, due to data insert errors)
* EmpFName – employee’s first name
* EmpLName – employee’s last name
* ProjectName – project’s name
* ProjStartDate – start date of the project
* Job – job title of the employee
* JobRate – hourly rate for the job title (every employee who performs the same job is paid the same)
* Hours – hours worked by a particular employee on a particular project

# Part 2: Data Retrieval

Run attached script CreateMyGuitarShop.sql to create database MyGuitarShop.

Using this database, answer each of the following requests with a single query. If there are multiple ways to write the query, provide just one. Paste below each question a screenshot from SSMS that shows the query AND results grid (if the are more results than fits the window, it is ok that the grid shows just beginning few and how many rows were returned.

In addition, prepare one Notepad file that contains just the text of all queries with question numbers as comments.

1. For each product, display code, name, category name (not ID), list price, discount percent and current price that is calculated by applying discount percent to the list price. Give the calculated column title CurrentPrice and display it in currency format.

Graphical user interface, application, Word

Description automatically generated

1. Display a list of categories, with IDs and names, and how many products are in each category, and what is the lowest and highest list price in each category. Make sure that all columns have meaningful titles.

Graphical user interface, application

Description automatically generated

1. Display a list of customer IDs and names (concatenate first name with the last name into one column CustomerName) and the phone associated with their billing address. Add two calculated columns that show area code and local phone for each phone number.

Graphical user interface, application, table, Excel

Description automatically generated

1. Display product codes and names with date added and calculated column that shows how many months ago the product was added. Order by the calculated column in increasing order. Display date added without time, and make sure all columns have meaningful titles.

Graphical user interface, text, application

Description automatically generated

1. For each order, display OrderID, ProductTotal, ShipAmount, TaxAmount, and OrderTotal. ProductTotal and OrderTotal are calculated columns. ProductTotal is calculated by summing up (ItemPrice - DiscountAmount) \* Quantity ever all items that belong to the order (OrderItems table). OrderTotal is calculated by adding ProductTotal to ShipAmount and Tax Amount.

Graphical user interface, text, application

Description automatically generated

1. Write a query that displays all data of products that were never ordered. (This query will return empty result set, because all products were ordered.)

Graphical user interface, text, application, email

Description automatically generated

1. Display Customer IDs, first names, and last names, together with how many orders each customer has, but include only customers that have more than one order.

Graphical user interface, application

Description automatically generated

1. Display customers last names and first names together order IDs and dates. Include customers who do not have orders. Display dates without time information. Order the report alphabetically by last names and first names, and for the same customer by the order date displaying most recent order first

Graphical user interface, text, application

Description automatically generated

1. Display order ID and date of the most recent order, together with last name and first name of the customer

Graphical user interface, text, application

Description automatically generated

1. Display category IDs and product code, name and list price for a product that has the highest list price in its category. Order by CategoryID.

I could solve this task only without displaying Product Code and ProductName:

Graphical user interface, text, application

Description automatically generated

1. Display customer IDs, first names and last names, together with order IDs for this customer and products ordered (code, name, and quantity) for each order. (Hint: you will need to join 4 tables.)

Graphical user interface, text, application

Description automatically generated

# Submit

Deliverables:

1. This document with queries and results pasted in places indicated by yellow highlight
2. For Part 1: script named *create\_Project.sql*
3. For Part 2: one Notepad file will all the queries (no results), with question numbers indicated

Zip the folder that contains all deliverables. Submit to the Assignment section of the CPRG 212 course D2L site. Due date is indicated on the course schedule and on the Assignment dropbox on D2L.

# Marking Rubric

|  |  |
| --- | --- |
| **Marking Requirement** | **Marks** |
| **Part 1: Create Database Tables and Insert Data** | **40** |
| Tables creates with appropriate type attributes | 10 |
| Primary key defined for each table | 4 |
| Foreign keys defined as appropriate | 6 |
| Additional constraints imposed as per requirements | 5 |
| Sample data is inserted into the tables | 10 |
| Report queries with results included as required | 5 |
| **Part 2: Data Retrieval** | **60** |
| Q 1 | 5 |
| Q 2 | 5 |
| Q 3 | 5 |
| Q 4 | 5 |
| Q 5 | 5 |
| Q 6 | 5 |
| Q 7 | 5 |
| Q 8 | 5 |
| Q 9 | 5 |
| Q 10 | 5 |
| Q 11 | 5 |
| Screenshots with queries results included | 5 |
| **Total:** | **100** |