BAX-421 - Data Management
Homework 2

Mehul Rangwala

Fall 2022

Total Points: 54 - Due: Monday, November 7, 2022 11:59 PM

Instructions

1. Please follow the instructions available on the recording on the homework submission guidelines.
2. Write the queries for two databases in MySQL:
   - ENTERTAINMENT AGENCY database
   - ACCOUNTS PAYABLE database
3. You should have already installed these databases from prior weeks. Continue to work with it.
4. Please reuse the ER diagrams for the Entertainment Agency and Accounts Payable databases from prior week assignments.
5. Write the queries for the **Solmaris Condo Group** Database in SQL Server. This exercise is optional but highly recommended. The script to generate the Solmaris Condo Group database and the ER diagram are available under Files.
6. Please do not use Window functions or Common Table Expressions (CTEs) for any queries on this assignment. This includes the bonus SQL Server queries at the end. Window functions and CTEs (though perfectly applicable in some cases) will earn no credit **for now**. You can start writing queries using Window functions and CTEs from Homework 3. For this assignment, please stick to Data Manipulation, Subqueries, and Joins.

ENTERTAINMENT AGENCY DATABASE *(8 points)*

Question 1

Display Customers and their preferred styles, but change 50's, 60's, 70's, and 80's music to 'Oldies'. This query should return 36 rows.

Question 2

Display all the engagements in October 2017 that start between noon and 5 p.m. Note: This database already has fields using the correct datatypes (date and time). Assume the dates and times were stored as strings. Write this query under such an assumption. This query should return 17 rows.

Question 3

List entertainers and display whether the entertainer was booked (on the job) on Christmas 2017 (December 25th). For this, you have to display three columns – EntertainerID, Entertainer Stage Name, and a new column indicating if the engagement was booked on Christmas or not. The query should return 13 rows.

Question 4

Find customers who like Jazz but not Standards. The query should return 2 rows.

ACCOUNTS PAYABLE DATABASE *(16 points)*

Question 1

Display the invoice totals from the invoices column and display all the invoice totals with a $ sign.

Question 2

Write a query to convert invoice date to a date in a character format and invoice total in integer format. Both conversions should be performed in the same query. Please note, then integers have no decimals.

Question 3

In the Invoices table, pad the single-digit and double-digit invoice numbers with one or two zeros before the invoice numbers. For example, the invoice number 1 should be displayed as 001, invoice number 20 should be displayed as 020, etc.

Question 4

Write a query to return the invoice_total column with one decimal digit and the invoice_total column with no decimal digits.

Question 5

Create a new table named Date_Sample using the script given below. Download this script from Canvas and run it in your MySQL Workbench on the ap database. Running this will create the Date_Sample table in your ap database.

```
USE ap;
CREATE TABLE date_sample
(
  date_id       INT       NOT NULL,
  start_date    DATETIME
);

INSERT INTO date_sample VALUES
(1, '1986-03-01 00:00:00'),
(2, '2006-02-28 00:00:00'),
(3, '2010-10-31 00:00:00'),
(4, '2018-02-28 10:00:00'),
(5, '2019-02-28 13:58:32'),
(6, '2019-03-01 09:02:25');
```

Display the start_date column, a new date column - call it Format_1 which displays date in this format: Mar/01/86, a new date column - call it Format_2 which displays 3/1/86 where the month and days are returned as integers with no leading zeros, and a third date column - call it Format_3 which displays only hours and minutes on a 12-hour clock with an am/pm indicator, for example, 12:00 PM.

Question 6

Write a query that returns the following columns from the Vendors table:

- The vendor_name column
- The vendor_name column in all capital letters
- The vendor_phone column
- A column that displays the last four digits of each phone number

When you get that working right, add the columns that follow to the result set. This can be more difficult for some students as these columns require use of nested functions.

- The vendor_phone column with the parts of the number separated by dots as in 111.111.1111

- A column that displays the second word in each vendor name if there is one and blanks if there isn't

Question 7

Write a query that returns these columns from the Invoices table:

- The invoice_number column
- The invoice_date column
- The invoice_date column plus 30 days
- The payment_date column
- A column named days_to_pay that shows the number of days between the invoice date and the payment date
- The number of the invoice date's month
- The four-digit year of the invoice date

When you have this working, add a WHERE clause that retrieves just the invoices for the month of May based on the invoice date, and not the number of the invoice month.

Question 8

Create a new table named string_sample using the script given below. Download this script from Canvas and run it in your MySQL Workbench on the ap database. Running this will create the string_sample table in your ap database.

```
USE ap;
CREATE TABLE string_sample
(
  emp_id        VARCHAR(3),
  emp_name      VARCHAR(25)
);

INSERT INTO string_sample VALUES
('1', 'Lizbeth Darien'),
('2', 'Darnell O''Sullivan'),
('17', 'Lance Pinos-Potter'),
('20', 'Jean Paul Renard'),
('3', 'Alisha von Strump');
```

Write a query that returns these columns from the string_sample table you created with the above script:

- The emp_name column
- A column that displays each employee's first name
- A column that displays each employee's last name

Use regular expression functions to get the first and last name. If a name contains three parts, everything after the first part should be considered part of the last name. Be sure to provide for last names with hyphens and apostrophes. You can refer to references online to learn about the regular expressions in MySQL. It is required that you use regular expressions and no other way. Any other way will not earn you credit (partial or full) even if your results are right.

SOLMARIS CONDO GROUP DATABASE *(0 points) - Optional but Recommended*

Write the following queries in **SQL Server**. The script to generate this database is available under Files and the ERD is also available. **Before running this script, please right-click the Databases folder in the SQL Server Management Studio and create a new database. Then run the script to create the tables.** Even though this part is optional, I highly recommend that you write these 13 queries in SQL Server so that you can get used to the SQL Server environment and work on the bonus queries following this database question.

Question 1

List the last name and first name of every owner who does not live in Bowton.

Question 2

Labor is billed at the rate of $35 per hour. List the condo ID, category number, estimated hours, and estimated labor cost for every service request. To obtain the estimated labor cost, multiply the estimated hours by 35. Use the column name ESTIMATED_COST for the estimated labor cost.

Question 3

List the owner number and last name for all owners who live in Florida (FL), Georgia (GA), or South Carolina (SC).

Question 4

How many one-bedroom condos are located at each location?

Question 5

For every condo, list the location number, unit number, condo fee, owner number, owner's first name, and owner's last name.

Question 6

For every completed or open service request for janitorial work, list the condo ID, description, and status.

Question 7

For every service request for janitorial work, list the condo ID, location number, unit number, estimated hours, spent hours, owner number, and owner's last name.

Question 8

List the first and last names of all owners who have a three-bedroom condo. Write the query in two ways.

Question 9

List the unit numbers of any pair of condos that have the same square footage. For example, one pair would be unit number 201 and unit number 401, because the square footage for both units is 1,030 square feet. The first unit number listed should be the major sort key and the second unit number should be the minor sort key.

Question 10

List the location number, unit number, and condo fee for condos whose owners live in Bowton or own one-bedroom condos.

Question 11

List the location number, unit number, and condo fee for condos whose owners live in Bowton and own a one-bedroom condo.

Question 12

List the location number, unit number, and condo fee for condos whose owners live in Bowton but do not own one-bedroom condos.

Question 13

Find the service ID and condo ID for each service request whose estimated hours is greater than the number of estimated hours of at least one service request on which the category number is 5.

Question 14

Find the service ID and condo ID for each service request whose estimated hours is greater than the number of estimated hours on every service request on which the category number is 5.

Question 15

List the condo ID, square footage, owner number, service ID, number of estimated hours, and number of spent hours for each service request. Make sure each condo is included regardless of whether the condo currently has any service requests for category 4.

SQL SERVER BONUS QUERY 1 *(10 points)*

A two-player computer game has been played by a few pairs of players and their scores are recorded as follows.

First Player	Second Player	Points
Joe	Ryan	120
Sue	Jackie	200
Ryan	Sue	50
Ryan	Joe	100

Write a query in **SQL Server** to generate the following output.

First Player	Second Player	Points
Joe	Ryan	220
Sue	Jackie	200
Ryan	Sue	50

To write this query, you can create a temp table instead of a permanent table in SQL Server. Here is the sccript you can use to generate the temp table. I have provided the script in a *.sql file because copying from this document and pasting did not retain the formatting. You can download the script file available on Canvas which contains this code. Open it in SQL Server Management Studio by clicking on File → Open → File. Run the script first and then begin writing your query underneath it.

```
IF OBJECT_ID('tempdb.dbo.#GameScores','U') IS NOT NULL
DROP TABLE #GameScores;
GO

CREATE TABLE #GameScores
(
FirstPlayer  VARCHAR(10),
SecondPlayer VARCHAR(10),
Score  INTEGER
);
GO

INSERT INTO #GameScores VALUES
```

```
('Joe','Ryan', 120),
('Sue', 'Jackie', 200),
('Ryan', 'Sue', 50),
('Ryan', 'Joe', 100);
GO
```

SQL Server drops a temporary table automatically when you close the connection that created it. So you don't have to worry about deleting it. Every time you run this query, you will have to first create the temporary table.

SQL SERVER BONUS QUERY 2 *(20 points)*

A secretary is trying to manage an executive's schedule which has dates corresponding to various events and meetings that this executive has commitments for. As you see below, the schedule is quite scattered. The secretary wants to create an alternate schedule which shows only non-overlapping dates. Here is the original schedule.

Start Date	End Date
1/11/22	1/13/22
1/11/22	1/15/22
1/11/22	1/12/22
1/13/22	1/19/22
1/20/22	1/22/22
1/24/22	1/26/22
1/25/22	1/29/22

You need to write a **SQL Server** query which displays the following non-overlapping schedule.

Start Date	End Date
1/11/22	1/19/22
1/20/22	1/22/22
1/24/22	1/29/22

To write this query, you can create a temp table instead of a permanent table in SQL Server. Here is the sccript you can use to generate the temp table. I have provided the script in a *.sql file because copying from this document and pasting did not retain the formatting. You can download the script file available on Canvas which contains this code. Open it in SQL Server Management Studio by clicking on File → Open → File. Run the script first and then begin writing your query underneath it.

```
IF OBJECT_ID('tempdb.dbo.##DetailedSchedule','U') IS NOT NULL
  DROP TABLE ##DetailedSchedule;
GO

CREATE TABLE ##DetailedSchedule
(
StartDate DATE,
EndDate DATE
);
GO

INSERT INTO ##DetailedSchedule VALUES
('1/11/2022','1/13/2022'),
('1/11/2022','1/15/2022'),
('1/11/2022','1/12/2022'),
('1/13/2022','1/19/2022'),
('1/20/2022','1/22/2022'),
('1/24/2022','1/26/2022'),
('1/25/2022','1/29/2022');
GO
```

SQL Server drops a temporary table automatically when you close the connection that created it. So you don't have to worry about deleting it. Every time you run this query, you will have to first create the temporary table.