Project 1 - Utility Billing System

Databases - Fall 2024

1 Introduction and Overview

The project is about writing SQL queries using the given Utility Billing System Database. The project submission deadline is **Tuesday October 8th, 11:55 PM**.

There are a total of 16 queries, and each query carries 10 points. You will be graded out of a total of 160, based on the correctness of your queries. So, make sure that your queries are correct, and return only the required data (i.e. no extra tuples, and attributes are returned except the ones that were required.) Please find the schema diagram in the appendix, Figure 1. The SQL files for creating and populating the relevant tables in the database are provided as a zip archive, which can be downloaded from LMS.

2 Queries

1. Retrieve the `FirstName`, `LastName`, `Address`, `DivisionName`, `SubDivName`, `Connection Type Description`, `MeterType`, `Import_PeakUnits`, `Import_OffPeakUnits` and `Total Amount Before DueDate` for each customer connection for the month of January 2024.

2. Retrieve the `FirstName`, `LastName`, `TariffSlab`, `ConnectionID`, and `ConnectionType (Code and Description)` of all those Customers whose monthly electricity Bill for the month of March 2024 is PKR 100,000 or more.

3. Retrieve the `FirstName`, `LastName`, `TariffSlab`, `ConnectionID`, and `ConnectionType (code and description)` for all customers who have received a nonzero subsidy on their bills from the Government of Punjab for at least two months between January 2024 and June 2024.

4. Retrieve the `FirstName`, `LastName`, `ConnectionID`, `BillID`, `Billing month`, and `Billing year` for all customers who have unpaid bills.

5. Retrieve the `name` of subdivision(s) that has all the types of Industry connections.

6. Find the *total subsidy paid* by the Government of Punjab in the month of June 2024 to Residential Customers (`ConnectionTypeCode` = 1, 2, or 3).

7. Retrieve the `FirstName`, `LastName`, `ConnectionID`, and `ConnectionType (code and description)` of the Residential Customer whose Total Bill Amount is the highest among all residential customers for the month of May 2024.

8. Find the *total number of Residential Customers* (`ConnectionTypeCode` = 1, 2, or 3) who have received a nonzero subsidy on their bills from the Government of Punjab for the month of June 2024.

9. For each combination of Division and Subdivision, find the *total number of Commercial Customers* (`ConnectionTypeCode` = 11, 12, 13, 14, 15, 16) whose monthly electricity Bill for the month of March 2024 is PKR 500,000 or more.

10. For each combination of Division and Subdivision, calculate the *total TaxAmount billed to all customers* for the month of August 2024.

11. For each combination of Division and Subdivision, calculate the *total TaxAmount received* from all the paid bills for the month of August 2024.

12. For each `ConnectionType`, find the *total number of Customers* who have received a non-zero subsidy on their bills from the Government of Punjab for the month of June 2024.

13. For each combination of Division and Subdivision, retrieve the `FirstName`, `LastName`, `ConnectionID`, and `ConnectionType (code and description)` for the Residential Customer who has the highest monthly bill (`TotalAmount_BeforeDueDate`) compared to all other customers in the same (division, subdivision).

14. Identify the *subdivision in Lahore* division that has *billed the highest total amount in taxes*, and *retrieve this maximum tax amount.*

15. Retrieve the `FirstName`, `LastName`, `ConnectionID`, and `ConnectionType (code and description)` for the Residential Customer who consumed the highest total peak units during the months of June, July, and August 2024.

16. Retrieve the `FirstName`, `LastName`, `ConnectionID`, and `ConnectionType (code and description)` for all customers whose current bill for August 2024 is less than half of their average bill calculated from January to July 2024.

3 Getting Started

The following files are provided via LMS:

```
Project1_files.zip
    |-- schema
    |   |-- UtilitySchema.sql
    |-- populate
    |-- |-- popBill.sql
    |-- |-- popConType.sql
    |-- |-- popDivInfo.sql
    |-- |-- popPaymentMethods.sql
    |-- |-- popSubsidy.sql
    |-- |-- popTaxRates.sql
    |-- |-- popConnection.sql
    |-- |-- popCustomer.sql
    |-- |-- pop_PaymentDetails.sql
    |-- |-- popReadings.sql
    |-- |-- popTariff.sql
    |-- submission
    |   |-- s_rollnumber.sql
    |-- myexec.sql
    |-- dropall.sql
```

The `schema` directory contains the schema definition file. The `populate` directory contains the scripts to populate the database with data. The `submission` directory contains a sample submission file. The `myexec.sql` file contains the script to create the tables and populate them with data. The `dropall.sql` file contains the script to drop all the tables.

You will be using the Oracle Cloud ATP database, created in Project 0. You can either use the `sqlplus` tool, or the online SQL worksheet provided by Oracle Cloud, in order to write and test your queries. To get started, make sure that your cloud database is up and running. Visit your Oracle Cloud Dashboard, and click on the database you created in Project 0. If the database is **paused, or stopped**, click on "More Actions" and select "Start" from the dropdown. Figure 2 shows how to start a paused, or stopped database instance.

1. Go to the directory where you have the `.zip` file that you downloaded from LMS.
2. Extract the contents of the archive.
3. Open the directory, which contains the extracted contents of the archive, in the terminal of your choice (`Powershell/Bash/Zsh`). For example:

```
$ cd path/to/project1
$ ls
dropall.sql  myexec.sql  populate  schema  submission
```

4. Once the database is up and running, connect to your database instance, using the `sqlplus` client, by running the following command (as done in Project 0).

```
$ sqlplus <user_name>/<your_password>@<connection_alias>
```

5. Once you are connected to the database, you can run the `myexec.sql` script, by running the following command (inside `sqlplus`):

```
> @myexec;
```

This will create the tables, and populate them with data. The process should take around 20-30 minutes. Wait for the database to get populated correctly with all the values. Here's a sample output of the script execution:

```
$ sqlplus 'rafay/<my_password>@dbfall24_high'

SQL*Plus: Release 23.0.0.0.0 - Production on Mon Sep 23 12:55:12 2024
Version 23.4.0.24.05

Copyright (c) 1982, 2024, Oracle.  All rights reserved.

Last Successful login time: Mon Sep 23 2024 12:52:42 +05:00

Connected to:
Oracle Database 23ai Enterprise Edition Release 23.0.0.0.0 - Production
Version 23.6.0.24.07

SQL> @myexec;

Table created.


Table created.

...

1 row created.


1 row created.


1 row created.

...
```

```
Commit complete.

SQL>
```

6. Once the script has executed successfully, you can start writing your queries. Try to run a simple query, in the `sqlplus` tool e.g. a `describe` R query returns the schema of the relation R. For instance:

```
SQL> DESCRIBE Tariff;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
TARIFFCODE                                NOT NULL VARCHAR2(50)
TARIFFTYPE                                NOT NULL NUMBER(38)
CONNECTIONTYPECODE                        NOT NULL NUMBER(38)
SLAB                                      NOT NULL NUMBER(38)
STARTDATE                                 NOT NULL DATE
ENDDATE                                   NOT NULL DATE
THRESHOLDLOW_PERHOUR                      NOT NULL NUMBER(10,4)
THRESHOLDHIGH_PERHOUR                     NOT NULL NUMBER(10,4)
TARRIFDESCRIPTION                         NOT NULL VARCHAR2(50)
RATEPERUNIT                               NOT NULL NUMBER(10,2)
MINAMOUNT                                          NUMBER(10,2)
MINUNIT                                            NUMBER(38)
```

4 Submission Instructions

You only need to submit a single `.sql` file, containing all of your queries. Make sure your submitted file follows the following conventions:

1. At the top of the file, add two comments containing your name, and your roll number, as follows:

```
-- Name: Abdul Rafay
-- Roll Number: 24100173
```

2. Before each query, add a comment containing the question number, as follows:

```
-- Q1
SELECT ...
```

3. Make sure that your queries are correct, and return the expected results.

4. Submit your query file on LMS. Make sure that the file is named as `s_<roll_number>.sql`, e.g. `s_24100173.sql`.

A sample submission file, `submission/s_rollnumber.sql` is provided. You can choose to use that file to write your queries Make sure to update the file with your name, roll number and your queries. Rename the file according to your roll number as well.

Appendix

Database schema

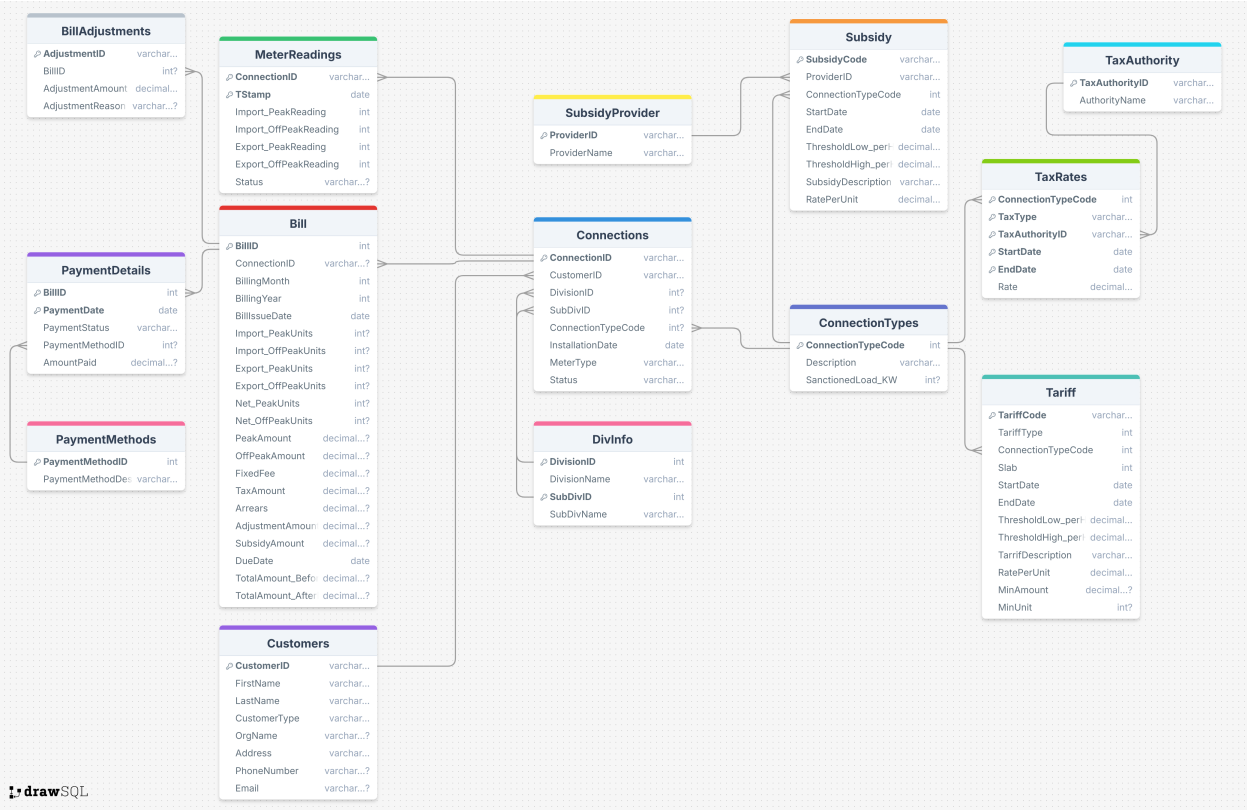

Figure 1: The diagram shows the schema you will be working with. The "key" icon next to attribute names indicates that it is part of the primary key. The corresponding datatype for each attribute is shown here as well. For more details, refer to the file `schema/UtilitySchema.sql`

.

Making sure DB instance is up and running

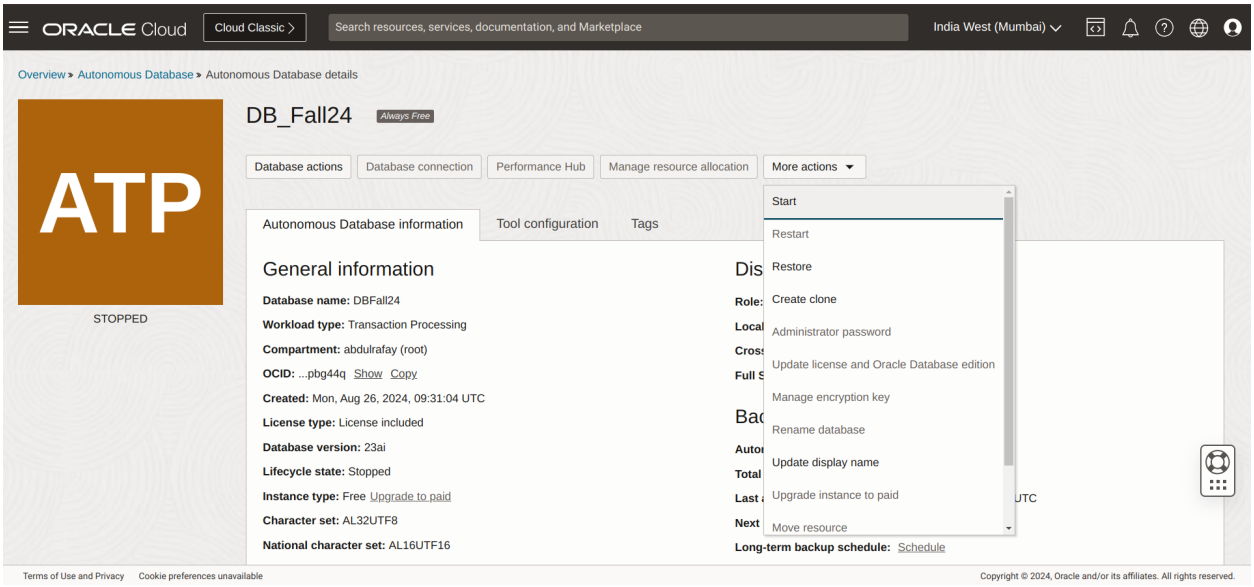

Figure 2: In order to start the stopped database, click on the "Start" button from the "More Actions" dropdown.