**Nanyang Technological University  
Nanyang Business School**

**BC2402 – Designing and Developing Databases**Semester 1

**Individual Assignment**

**Too Fast Too Furious**

**1. INTRODUCTION**

*A. Driving Safety and Sustainable Transport*

Driving is a common and hazardous activity that is a prominent cause of death worldwide. Driver behavior represents a predominant cause, contributing to over 90% of crashes. Drivers adapt and drive safely in a broad range of situations but fail when expectations are violated or when feedback is inadequate.

Relatedly, eco-driving is a style of vehicle driving that reduces energy consumption, maximizing mileage per unit of energy consumed. Eco-driving requires adherence to speed limits, accelerating and braking smoothly, avoiding acceleration, use of engine braking, and maintaining a constant speed. Eco-driving may also include vehicle maintenance, trip planning, switching to other transport where appropriate and vehicle choice. Eco-driving may save up to 25% of fuel. There is an overlap between eco-driving and safe driving since safe driving entails observing speed limits and avoiding harsh acceleration and braking.

*B. Journey towards a “Car-Light Singapore”*

The recently released Urban Redevelopment Authority (URA) Draft Master Plan 2013 may be a developmental blueprint, but it is also very much a land transport road map - one which supplements the Land Transport Authority's own master plan.

It contains a strong two-pronged theme: reclaiming the city from the car and reducing the need to commute.

- The Straits Times, Nov 28, 2013 https://www.straitstimes.com/singapore/journey-towards-a-car-light-singapore

**2. DATA DESCRIPTIONS**

You are provided with a dataset extracted from a local car rental company. The dataset consists of multiple tables. Specifically,

1. CustomerTBL

*This table contains data about a customer.*

[customerID] uniquely identifies each record

[name] presents the first 4 characters of the customer’s full name. The remaining characters are removed to anonymize the records

[gender] 0 indicates female, 1 indicates male

[member\_since] contains the “month day, year” value

[driving\_age\_yr] contains the customer’s years of driving experience

[postal] contains the first 2 numbers of the customer’s home postal code

1. VehicleTBL

*This table contains data specific to each rental vehicle.*

[vehicleID] uniquely identifies each record

[type] contains the value representing a carType

[brand] contains the brand name of the vehicle

[model] contains the model value of the vehicle

[age] is the age (in years) of the vehicle

[total\_distance\_km] is the vehicle’s total traveled mileage in km.

[full\_tank\_gal] is the full-tank volume in gallons

[km\_per\_gal\_highway] is the estimated fuel efficiency on the highway

[km\_per\_gal\_city] is the estimated fuel efficiency in the city

[CO2avg\_WLTP] is the estimated carbon footprint based on the Worldwide harmonized Light vehicles Test Procedures, i.e., grams of carbon emissions per km

[CO2avg\_NEDC] is the estimated carbon footprint based on the New European Driving Cycle, i.e., grams of carbon emissions per km

[fee\_per\_hour] is the rental fee for an hour of usage

1. orderTBL

*This table contains data specific to an order (i.e., a rental).*

[orderID] uniquely identifies each record

[custeromID] represents the customer for the order

[vehicleID] represents the vehicle used in the order

[startDate] represents the starting date of the order

[endDate] represents the ending date of the order

[startTime\_hr] represents the time (the specific hour) at which the order commences

[startTime\_min] represents the time (the specific minute) at which the order commences

[startTime\_ss] represents the time (the specific second) at which the order commences

[endTime\_hr] represents the time (the specific hour) at which the order completes

[endTime\_min] represents the time (the specific minute) at which the order completes

[endTime\_ss] represents the time (the specific second) at which the order completes

[startLoc] contains the first 2 digits of the starting location postal code

[endLoc] contains the first 2 digits of the ending location postal code

[start\_fuel\_gal] indicates the amount of fuel (in gallons) at the beginning of the order

[end\_fuel\_gal] indicates the amount of fuel (in gallons) at the end of the order

1. complaintTBL  
   *This table contains data specific to a complaint.*

[complaintID] uniquely identifies each record

[customerID] represents the customer for the complaint

[date] represents the date at which the complaint is filed

[issue] contains the issue category

[severity] contains a numeric severity, i.e., 1 indicates least severe, 5 indicates most severe

[resolved] contains 0 or 1, i.e., 0 indicates unresolved, 1 indicates resolved

1. incidentTBL

*This table contains data about an incident (i.e., violations of traffic regulations)*

[incidentID] uniquely identifies each record

[vehicleID] represents the vehicle involved in the incident

[date] indicates the incident date

[issue] contains an incidentType value

1. incidentTypeTBL

*This table contains data about the 13 types of incidents.*

[incidentTypeID] uniquely identifies each record

[detail] provides details on the violation

1. vehicleTypeTBL

*This table contains data about the types of rental vehicles.*

[vehicleTypeID] uniquely identifies each record

[type] provides details on the specific type

1. postalCodeTBL

*This table contains the general locations based on the first 2 digits of postal codes.*

[postalCode] uniquely identifies each record

[generalLoc] provides the details of the location

To manage your workload, you are encouraged to use the SQL database implementations provided together with this document.

Do note that you must submit your database implementation, if you choose to use the datasets to implement your own databases. Otherwise (if you shall be using the provided databases), you are not required to submit the database implementations.

**3. PROJECT DELIVERABLES**

***The due date for the assignment is 6 October 2023 (23:59 hrs NTULearn server time)***

You are not expected to modify the provided mySQL database implementation. You are only expected to submit one file, as follows:

1. 1 x SQL script file

**A. SQL script file**

You are tasked to develop some SQL scripts to query the data, as follows:

You are to create SQL scripts (e.g., SELECT statements) that generate answers for the following queries.

1. Table considered: <postalCodeTBL*>*

What are the unique [generalLoc]?

SELECT DISTINCT generalLOC

FROM postalcodetbl;

1. Table considered: <incidentTypeTBL>

How many incident types are related to speed limit issues?

Ans: 4

SELECT COUNT(\*) AS "Number Of Incident Types"

FROM incidentTypeTBL

WHERE IncidentTypeTBL.detail LIKE '%Exceeding the road speed limit%';

1. Table considered: <incidentTBL>

For each [year], on each [incidentType], how many incidents are recorded?

SELECT YEAR(STR\_TO\_DATE(date, '%M %d, %Y')) AS Year, IncidentType, COUNT(\*) AS IncidentCount

FROM incidentTBL

GROUP BY Year, IncidentType

ORDER BY Year ASC, IncidentType ASC;

A screenshot of a black and white list

Description automatically generated

Check the asc order

1. Tables considered: <complaintTBL> + <customerTBL>

For each year (2010 to 2018), on each issue category, display the total number of complaints, and the respective breakdowns between females and males

SELECT

YEAR(STR\_TO\_DATE(c.date, '%M %d, %Y')) AS Year,

c.Issue AS IssueCategory,

SUM(IF(cu.Gender = 0, 1, 0)) AS FemaleComplaints,

SUM(IF(cu.Gender = 1, 1, 0)) AS MaleComplaints,

COUNT(\*) AS TotalComplaints

FROM complaintTBL c

JOIN customerTBL cu ON c.CustomerID = cu.CustomerID

WHERE YEAR(STR\_TO\_DATE(c.date, '%M %d, %Y')) BETWEEN 2010 AND 2018

GROUP BY Year, IssueCategory

ORDER BY Year ASC, IssueCategory ASC;

1. Tables considered: <vehicleTBL> + <orderTBL> + <customerTBL> + <postalcodeTBL>

For each year, on each generalLoc, display the total customer value (i.e., total rental fees recorded for the year).

*Note: You need to sum up the values for the same generalLoc (under different postalCode).*

SELECT

YEAR(STR\_TO\_DATE(o.endDate, '%d-%b-%Y')) AS RentalYear,

po.generalLoc,

SUM(

(

(o.endTime\_hr - o.startTime\_hr) +

(o.endTime\_min - o.startTime\_min) / 60 +

(o.endTime\_ss - o.startTime\_ss) / 3600

) \* v.fee\_per\_hour

) AS TotalCustomerValueForEachGeneralLoc

FROM orderTBL o

JOIN vehicletbl v ON o.vehicleID = v.vehicleID

JOIN customertbl cu ON o.customerID = cu.customerID

JOIN postalcodetbl po ON cu.postal = po.postalcode

GROUP BY RentalYear, po.generalLoc

ORDER BY RentalYear, po.generalLoc;

1. Tables considered: <complaintTBL> + < orderTBL > + <vehicleTBL>

*Complaint analytics. The purpose of this query is to prepare the data for analyses on complaint frequency and patterns.*

For each year, on each vehicle, display the average traveled distance for a complaint.

*Note: If you encounter problems with the data, provide a script to illustrate the issue. You can also propose possible workarounds.*

*SELECT*

*YEAR(STR\_TO\_DATE(c.date, '%M %D %Y')) AS ComplaintYear,*

*o.vehicleID,*

*AVG(o.distance\_m)/1000 AS AverageDistanceForComplaint*

*FROM complaintTBL c*

*JOIN orderTBL o ON c.customerID = o.customerID*

*JOIN vehicletbl v ON o.vehicleID = v.vehicleID*

*GROUP BY ComplaintYear, o.vehicleID*

*ORDER BY ComplaintYear, o.vehicleID;*

1. Tables considered: <customerTBL> + < orderTBL > + <vehicleTBL>

*Who makes the green choices? The purpose of this query is to prepare the data for analyses on green consumption.*

For each gender and age group\*, display the percentage of rentals for each vehicle brand and model.

\*consider 3 age groups: between 21 and 30 (both inclusive), between 31 and 50 (both inclusive), and >=51.

SELECT

IF(cu.gender = "0", "F", "M") AS gender,

IF(cu.age BETWEEN 21 AND 30, '21-30',

IF(cu.age BETWEEN 31 AND 50, '31-50', '>=51')) AS age\_group,

v.brand, v.model, COUNT(o.orderID) AS NumberOfRentals,

SUM(COUNT(o.orderID)) OVER (PARTITION BY cu.gender,

IF(cu.age BETWEEN 21 AND 30, '21-30',

IF(cu.age BETWEEN 31 AND 50, '31-50', '>=51'))) AS TotalRentalsOfGenderAge,

100.0 \* COUNT(o.orderID) / SUM(COUNT(o.orderID))

OVER (PARTITION BY cu.gender,

IF(cu.age BETWEEN 21 AND 30, '21-30',

IF(cu.age BETWEEN 31 AND 50, '31-50', '>=51'))) AS Percentage

FROM customertbl cu

JOIN ordertbl o ON cu.customerID = o.customerID

JOIN vehicletbl v ON o.vehicleID = v.vehicleID

GROUP BY cu.gender, age\_group, v.brand, v.model

ORDER BY cu.gender, age\_group, v.brand, v.model;

1. Tables considered: <customerTBL> + <orderTBL>

Display a list of customers (and the corresponding vehicle) who used the same vehicle more than once.

SELECT cu.name, cu.customerID,o.vehicleID, COUNT(o.VehicleID) AS TimesUsed

FROM customertbl cu,ordertbl o

WHERE cu.customerID = o.customerID

GROUP BY cu.name, cu.customerID, o.vehicleID

HAVING COUNT(o.vehicleID) > 1;

1. Tables considered: <customerTBL> + <orderTBL>

For each customer, on each starting and ending location pair, display the number of rentals and the average distance.

SELECT cu.customerID, o.startLoc, o.endLoc, COUNT(\*) AS NumberOfRentals,

AVG(o.distance\_m)/1000 AS AverageDistanceTravelled

FROM customerTBL cu

JOIN orderTBL o ON cu.customerID = o.customerID

JOIN vehicletbl v ON o.vehicleID = v.vehicleID

GROUP BY cu.customerID, o.startLoc, o.endLoc

ORDER BY cu.customerID, o.startLoc, o.endLoc;

1. Tables considered: <customerTBL> + <orderTBL> + <vehicleTBL>

*The purpose of this query is to prepare a dataset for carbon footprint analytics.*

For each order year and customer, display the total carbon footprint in both standards.

SELECT cu.customerID,

cu.name,

YEAR(STR\_TO\_DATE(o.startDate, '%d-%b-%Y')) AS OrderYear,

SUM(v.CO2avg\_WLTP) AS TotalCO2avg\_WLTP,

SUM(v.CO2avg\_NEDC) AS TotalCO2avg\_NEDC

FROM orderTBL o

JOIN customerTBL cu ON o.customerID = cu.customerID

JOIN vehicletbl v ON o.vehicleID = v.vehicleID

GROUP BY cu.name, cu.customerID, OrderYear

ORDER BY cu.name, cu.customerID, OrderYear;

**4. SUBMISSION**

A submission folder will be made available on NTULearn. You can make as many submissions as you deem necessary, but only the latest submission will be evaluated.

The submission must be made by **6 October 2023, 23:59**.