addb assignment

ST10053561

Varsity College – Cape Town

ADDB7311

**Table of Contents**

[Question 1 2](#_Toc148446730)

[Question 2 3](#_Toc148446731)

[Create Tables 3](#_Toc148446732)

[Insert Values In a Table 5](#_Toc148446733)

[The output of a Table 15](#_Toc148446734)

[Question 3 18](#_Toc148446735)

[Question 4 19](#_Toc148446736)

[Output of a Question 4.1 20](#_Toc148446737)

[Question 5 22](#_Toc148446738)

[The output of Question 5.1 24](#_Toc148446739)

[View in SQL: 26](#_Toc148446740)

[Security 26](#_Toc148446741)

[Complexity 26](#_Toc148446742)

[Reusability and Consistency 26](#_Toc148446743)

[The output of Question 5.3.2 27](#_Toc148446744)

[Question 6 27](#_Toc148446745)

[The output of Question 6 30](#_Toc148446746)

[Figure 1 : ERD Diagram 2](#_Toc148444684)

[Figure 2 : CustomerDelivery Table Output 14](#_Toc148444685)

[Figure 3 : Staff Delivery Table Output 15](#_Toc148444686)

[Figure 4 : Billing Table Output 15](#_Toc148444687)

[Figure 5 : Delivery Items Table Output 16](#_Toc148444688)

[Figure 6 : Vehicle Table Output 16](#_Toc148444689)

[Figure 7 : Driver Table Output 16](#_Toc148444690)

[Figure 8 : Driver Delivery Table Output 17](#_Toc148444691)

[Figure 9 : Output of Granted Users 17](#_Toc148444692)

[Figure 10 : Output Vehicle less than 80000 19](#_Toc148444693)

[Figure 11 : Most of Deliveries Processed 22](#_Toc148444694)

[Figure 12 : View Table Output 26](#_Toc148444695)

[Figure 13 : Output of Elite Bills Part 1 28](#_Toc148444696)

[Figure 14 : Output of Elite Bills Part 2 29](#_Toc148444697)

# Question 1

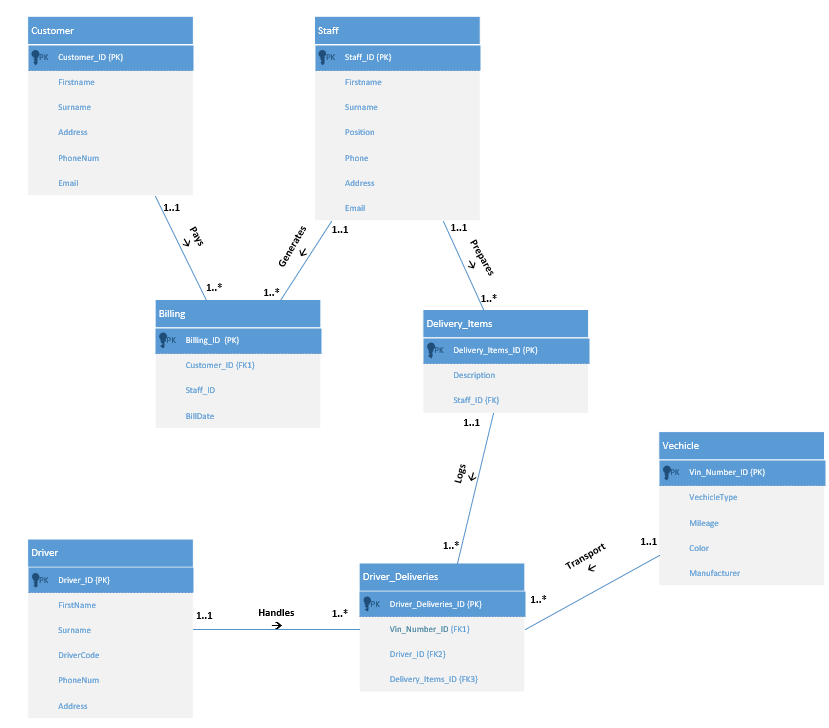


Figure : ERD Diagram

# Question 2

## Create Tables

--QUESTION 2

--CREATE TABLE Customers\_Delivery

CREATE TABLE CustomersDelivery (

CustomerId NUMBER PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

Surname VARCHAR(50) NOT NULL,

Address VARCHAR(100) NOT NULL,

PhoneNumber VARCHAR(12) NOT NULL,

Email VARCHAR(100) NOT NULL

);

--CREATE TABLE Staff\_Delivery

CREATE TABLE StaffDelivery (

StaffId NUMBER PRIMARY KEY NOT NULL,

FirstName VARCHAR(50) NOT NULL,

Surname VARCHAR(50) NOT NULL,

Position VARCHAR(50) NOT NULL,

PhoneNumber VARCHAR(20) NOT NULL,

Address VARCHAR(100) NOT NULL,

Email VARCHAR(100) NOT NULL

);

--CREATE TABLE Billing

CREATE TABLE Billing (

BillId NUMBER PRIMARY KEY NOT NULL,

CustomerId NUMBER NOT NULL,

StaffId INT NOT NULL,

BillDate VARCHAR(20) NOT NULL,

FOREIGN KEY(CustomerId) REFERENCES CustomersDelivery(CustomerId),

FOREIGN KEY(StaffId) REFERENCES StaffDelivery(StaffId)

);

--CREATE TABLE DeliveryItems

CREATE TABLE DeliveryItems (

DeliveryItemId NUMBER PRIMARY KEY NOT NULL,

Description VARCHAR(100) NOT NULL,

StaffId NUMBER NOT NULL,

FOREIGN KEY (StaffId) REFERENCES StaffDelivery(StaffId)

);

-- Create the table

CREATE TABLE Drivers (

Driver\_ID NUMBER(5) PRIMARY KEY NOT NULL,

First\_Name VARCHAR(50) NOT NULL,

Surname VARCHAR(50) NOT NULL,

Driver\_Code VARCHAR(5) NOT NULL,

Phone\_Num VARCHAR(15) NOT NULL,

Address VARCHAR(100) NOT NULL

);

--CREATE TABLE Vehicle

CREATE TABLE Vehicle (

VinNumber VARCHAR(20) PRIMARY KEY NOT NULL,

VehicleType VARCHAR(50) NOT NULL,

Mileage NUMBER NOT NULL,

Colour VARCHAR(20) NOT NULL,

Manufacturer VARCHAR(50) NOT NULL

);

--CREATE TABLE DriverDelivery

CREATE TABLE DriverDelivery (

DriverDeliveryId NUMBER PRIMARY KEY NOT NULL,

VinNumber VARCHAR(20) NOT NULL,

Driver\_ID NUMBER NOT NULL,

DeliveryItemId NUMBER NOT NULL,

FOREIGN KEY (VinNumber) REFERENCES Vehicle(VinNumber),

FOREIGN KEY (Driver\_ID) REFERENCES Drivers(Driver\_ID),

FOREIGN KEY (DeliveryItemId) REFERENCES DeliveryItems(DeliveryItemId)

);

## Insert Values In a Table

--INSERT VALUES FOR CustomersDelivery

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11011,'Bob','Smith','18 Water rd','0877277521','bobs@isat.com');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11012,'Sam','Hendricks','22 Water rd','0863257857','shen@mcom.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11013,'Larry','Clark','101 Summer lane','0834567891','larc@mcom.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11014,'Jeff','Jones','55 Mountain way','0612547895','jj@isat.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11015,'Andre','Kerk','5 Main rd','0827238521','akerk@mcal.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11016,'Wayne','Smith','13 Water rd','0877277522','ws@isat.com');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11017,'John','Hendricks','29 Water rd','0863257851','jhen@mcom.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11018,'Sally','Clark','111 Summer lane','0834567892','sallyc@mcom.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11019,'Bridget', 'Bitterhour', '125 Mountain way', '0612547896', 'bb@isat.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES (11111,'Nicole', 'Kerk', '175 Main rd', '0827238529', 'nk@mcal.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES(11112,'Catherine', 'Smith', '19 Water rd', '0877277523', 'cath@isat.com');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES(11113,'Mel', 'Hendricks', '5 Water rd', '0863257852', 'melh@mcom.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES(11114,'Lucy', 'Du Plessis', '221 Summer lane', '0834567892', 'ldup@mcom.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES(11116,'Josh', 'Maverick', '155 Mountain way', '0612547897', 'joshm@isat.co.za');

INSERT INTO CustomersDelivery (CustomerId, FirstName, Surname, Address, PhoneNumber, Email)

VALUES(11117,'Stuart', 'Jones', '35 Main rd', '0827238521', 'sjones@mcal.co.za');

--insert statments for staffDelivery

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES (51011,'Sally','Du Toit','Logistics','0825698547','18 Main rd','sdut@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES (51012,'Mark','Wright','CRM','0836984178','12 Cape Way','mwright@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES (51013,'Harry','Sheen','Logistics','0725648965','15 Water Street','hsheen@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES (51014,'Jabu','Xolani','Logistics','0823116598','18 White Lane','jxo@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES (51015,'Roberto','Henry','Packaging','0783521451','55 Cape Street','rhenry@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES(51016,'Pat', 'Durant', 'Logistics', '0825698542', '1 Main rd', 'pd@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES(51017,'Steve', 'Maritz', 'CRM', '0836984173', '2 Cape Way', 'sm@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES(51018,'Maxwell', 'Dube', 'Logistics', '0725648964', '5 Water Street', 'max@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES(51019,'Shane', 'Mane', 'Logistics', '0823116595', '8 White Lane', 'smane@isat.com');

INSERT INTO StaffDelivery (StaffId, FirstName, Surname, Position, PhoneNumber, Address, Email)

VALUES(51111,'Bob', 'Truth', 'Packaging', '0783521456', '35 Cape Street', 'btruth@isat.com');

--INSERT VALUES FOR Billing table

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (800,11011,51011,'06-Sep-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (801,11012,51013,'07-Sep-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (800,11011,51011,'06-Sep-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (801,11012,51013,'07-Sep-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (802,11014,51015,'10-Nov-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (803,11015,51012,'09-Dec-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (804,11013,51014,'09-Dec-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (805,11111,51011,'06-Sep-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (806,11012,51013,'07-Sep-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (807,11014,51015,'10-Nov-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (808,11015,51012,'09-Dec-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (809,11113,51018,'09-Dec-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (810,11011,51011,'06-Sep-22');

INSERT INTO Billing (BillId, CustomerId, StaffId, BillDate)

VALUES (811,11012,51013,'07-Sep-22');

INSERT INTO Billing (BillId,CustomerId ,StaffID ,BillDate )

VALUES(812,11014,51016,'10-Nov-22');

INSERT INTO Billing (BillId,CustomerID ,StaffID ,BillDate )

VALUES(813,11117,51012,'09-Dec-22');

INSERT INTO Billing (BillId,CustomerID ,StaffID ,BillDate )

VALUES(814, 11013,51014,'09-Dec-22');

INSERT INTO Billing (BillId,CustomerID ,StaffID ,BillDate )

VALUES(815,11012,51111,'06-Sep-22');

INSERT INTO Billing (BillId,CustomerID ,StaffID ,BillDate )

VALUES(816,11012,51019,'07-Sep-22');

INSERT INTO Billing (BillId,CustomerID ,StaffID ,BillDate )

VALUES(817,'11014','51015','10-Nov-22');

INSERT INTO Billing (BillId,CustomerID ,StaffID ,BillDate )

VALUES(818,11112,51012,'09-Dec-22');

INSERT INTO Billing (BillId,CustomerID ,StaffID ,BillDate )

VALUES(819,11013,51014,'09-Dec-22');

INSERT INTO Billing (BillId,CustomerID ,StaffID ,BillDate )

VALUES(820,11116,51019,'09-Dec-22');

--insert staements for deliveryitems

INSERT INTO DeliveryItems (DeliveryItemId, Description, StaffId)

VALUES(71011,'House relocation',51011);

INSERT INTO DeliveryItems (DeliveryItemId, Description, StaffId)

VALUES (71012,'Delivery of specialized consignments',51017);

INSERT INTO DeliveryItems (DeliveryItemId, Description, StaffId)

VALUES (71013,'Delivery of specialized consignments',51015);

INSERT INTO DeliveryItems (DeliveryItemId, Description, StaffId)

VALUES (71014,'Office relocation',51012);

INSERT INTO DeliveryItems (DeliveryItemId, Description, StaffId)

VALUES (71015,'Delivery of specialized consignments',51014);

-- INSERT VALUES FOR Drivers table

INSERT INTO Drivers (Driver\_ID, First\_Name, Surname, Driver\_Code, Phone\_Num, Address)

VALUES (81011, 'Buthelezi', 'Marshall', 'C1', '0725698547', '18 Leopard creek');

INSERT INTO Drivers (Driver\_ID, First\_Name, Surname, Driver\_Code, Phone\_Num, Address)

VALUES (81012, 'Tina', 'Mtati', 'C', '0636984178', '12 Cape rd');

INSERT INTO Drivers (Driver\_ID, First\_Name, Surname, Driver\_Code, Phone\_Num, Address)

VALUES (81013, 'Jono', 'Mvuyisi', 'EC1', '0725648965', '15 Circle lane');

INSERT INTO Drivers (Driver\_ID, First\_Name, Surname, Driver\_Code, Phone\_Num, Address)

VALUES (81014, 'Richard', 'Smith', 'C1', '0623116598', '18 Beach rd');

INSERT INTO Drivers (Driver\_ID, First\_Name, Surname, Driver\_Code, Phone\_Num, Address)

VALUES (81015, 'Brett', 'Smith', 'EB', '0883521457', '55 Summer lane');

-- INSERT VALUES FOR Vechile table

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA55858541','Cutaway van chassis',115352,'RED','MAN');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA51858542','Flatbed truck',315856,'BLUE','ISUZU');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA35858543','Medium Standard Truck',789587,'SILVER','MAN');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA15851545','Flatbed truck',555050,'WHITE','TATA');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA35868540','Cutaway van chassis',79058,'WHITE','ISUZU');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA65858541','Cutaway van chassis',215352,'RED','MAN');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA61858542','Flatbed truck',215856,'BLUE','ISUZU');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA65858543','Medium Standard Truck',889587,'SILVER','MERC');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA65851545','Flatbed truck',155050,'WHITE','MAN');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA65868540','Cutaway van chassis',19058,'WHITE','ISUZU');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA75858541','Cutaway van chassis',315352,'RED','MAN');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA71858542','Flatbed truck',115856,'BLUE','ISUZU');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA75858543','Medium Standard Truck',989587,'SILVER','MAN');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA17851545','Flatbed truck',755050,'WHITE','TATA');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA75868540','Cutaway van chassis',29058,'WHITE','ISUZU');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA85858541','Cutaway van chassis',515352,'RED','MERC');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA81858542','Flatbed truck',715856,'BLUE','ISUZU');

INSERT INTO Vehicle (VinNumber, VehicleType, Mileage, Colour, Manufacturer)

VALUES ('1ZA85858543','Medium Standard Truck',789587,'SILVER','MAN');

INSERT INTO Vehicle (VinNumber,VehicleType,Mileage ,Colour ,Manufacturer )

VALUES ('1ZA85851545', 'Flatbed truck', '955050', 'WHITE', 'TATA');

INSERT INTO Vehicle (VinNumber,VehicleType,Mileage ,Colour ,Manufacturer )

VALUES ('1ZA85868540', 'Cutaway van chassis', '39058', 'WHITE', 'MERC');

-- INSERT VALUES FOR DriverDelivery table

INSERT INTO DriverDelivery (DriverDeliveryId, VinNumber, Driver\_ID, DeliveryItemId)

VALUES (91011,'1ZA55858541',81011,71011);

INSERT INTO DriverDelivery (DriverDeliveryId, VinNumber, Driver\_ID, DeliveryItemId)

VALUES (91012,'1ZA35858543',81012,71013);

INSERT INTO DriverDelivery (DriverDeliveryId, VinNumber, Driver\_ID, DeliveryItemId)

VALUES (91013,'1ZA17851545',81011,71015);

INSERT INTO DriverDelivery (DriverDeliveryId, VinNumber, Driver\_ID, DeliveryItemId)

VALUES (91014,'1ZA35868540',81013,71015);

INSERT INTO DriverDelivery (DriverDeliveryId, VinNumber, Driver\_ID, DeliveryItemId)

VALUES (91015,'1ZA15851545',81014,71012);

Select \* from CustomersDelivery;

Select \* from StaffDelivery;

Select \* from Billing;

Select \* from DeliveryItems;

Select \* from Vehicle;

Select \* from Drivers;

Select \* from DriverDelivery;

## The output of a Table

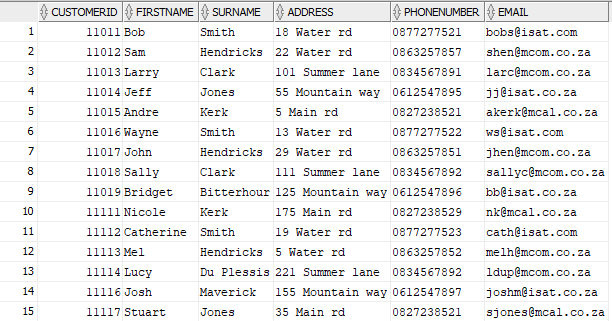


Figure : CustomerDelivery Table Output

A screenshot of a computer

Description automatically generated

Figure : Staff Delivery Table Output

A table of numbers and numbers

Description automatically generated

Figure : Billing Table Output

A screenshot of a computer

Description automatically generated

Figure : Delivery Items Table Output

A screenshot of a computer

Description automatically generated

Figure : Vehicle Table Output

A screenshot of a computer

Description automatically generated

Figure : Driver Table Output

A screenshot of a number

Description automatically generated

Figure : Driver Delivery Table Output

# Question 3

CREATE USER C##Tshepo IDENTIFIED BY tmphoabc2023;

CREATE USER C##Mya IDENTIFIED BY mrobertabc2023;

GRANT SELECT ANY TABLE TO C##Tshepo;

GRANT INSERT ANY TABLE TO C##Mya



Figure : Output of Granted Users

3.2

The separation of duties is crucial for two main reasons based on (Oracle Corporation , 2023):

Security: It limits the power of a single user and reduces the risk of unauthorized data access or manipulation. For example, Tshepo can only read data, not modify it.

Prevention of Fraud: It prevents one person from controlling all aspects of a transaction, reducing the risk of errors and fraud. For instance, Mya can insert data but cannot read all data.

# Question 4

4.1

SET SERVEROUTPUT ON;

DECLARE

--Local variables

Driver\_name Drivers.First\_Name %TYPE;

Driver\_code Drivers.Driver\_Code %TYPE;

Vin\_Number Vehicle.VinNumber %TYPE;

Mileage Vehicle.Mileage %TYPE;

CURSOR D\_miles IS

SELECT

Drivers.First\_Name,

Drivers.Surname,

Drivers.Driver\_Code,

Vehicle.VinNumber ,

Vehicle.Mileage

FROM

Drivers

INNER JOIN

DriverDelivery ON Drivers.Driver\_ID = DriverDelivery.Driver\_ID

INNER JOIN

Vehicle ON Vehicle.VinNumber = DriverDelivery.VinNumber

WHERE

Vehicle.Mileage < 80000

ORDER BY

Vehicle.Mileage;

BEGIN

FOR eachVal in D\_miles

LOOP

Driver\_name := eachVal.First\_Name || ', ' || eachVal.Surname;

Driver\_code := eachVal.Driver\_Code;

Vin\_Number := eachVal.VinNumber;

Mileage := eachVal.Mileage;

dbms\_output.put\_line('---------------------------------------------');

dbms\_output.put\_line('DRIVER : '|| Driver\_name);

dbms\_output.put\_line('CODE : '|| Driver\_code);

dbms\_output.put\_line('VIN NUMBER : ' || Vin\_Number);

dbms\_output.put\_line('MILEAGE : '|| Mileage);

dbms\_output.put\_line('---------------------------------------------');

END LOOP;

END;

## Output of a Question 4.1

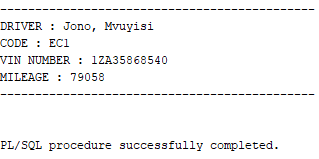


Figure : Output Vehicle less than 80000

4.2

The solution used in Q.4.1 is a PL/SQL anonymous block. An anonymous block is a PL/SQL block that appears within your application, and it is not named or stored in the database. The benefits of using this approach for ABC DELIVERY SPECIALISTS are based on (Oracle Database Group, 2023) And (Anon., 2022).

1. Ad-hoc Execution: Since the PL/SQL block is not stored in the database, it can be run on an ad-hoc basis whenever needed. This is useful for generating reports that do not need to be run on a regular schedule.

2. Performance: PL/SQL allows for SQL commands to be processed in blocks, which can significantly reduce the communication between the application and the database, leading to improved performance.

3. Error Handling: PL/SQL provides robust error handling capabilities, making it easier to debug and manage errors.

4. Cursor Management: The use of cursors in PL/SQL allows for more complex queries and data manipulation, providing greater flexibility in generating reports.

5. Security: Since the code is run directly by the application and not stored in the database, there's less risk of SQL injection attacks.

Overall, using a PL/SQL anonymous block for generating this report provides flexibility, performance benefits, robust error handling, and security advantages.

# Question 5

5.1

--QUESTION 5

SET SERVEROUTPUT ON;

DECLARE

driverID StaffDelivery.StaffId %TYPE;

Fname StaffDelivery.FirstName %TYPE;

surname StaffDelivery.Surname %TYPE;

DelProced DeliveryItems.DeliveryItemId %TYPE;

CURSOR DelVal IS

SELECT \* FROM(

SELECT

StaffDelivery.StaffId,

StaffDelivery.FirstName,

StaffDelivery.Surname,

COUNT(StaffDelivery.StaffId ) AS DeliveryCount

FROM

StaffDelivery

INNER JOIN

DeliveryItems ON StaffDelivery.StaffId = DeliveryItems.StaffId

INNER JOIN

DriverDelivery ON DeliveryItems.DeliveryItemId = DriverDelivery.DeliveryItemId

INNER JOIN

Vehicle ON Vehicle.VinNumber = DriverDelivery.VinNumber

GROUP BY

StaffDelivery.StaffId,

StaffDelivery.FirstName,

StaffDelivery.Surname

ORDER BY

DeliveryCount DESC)

WHERE ROWNUM=1 ;

BEGIN

FOR eachDelVal IN DelVal

LOOP

driverID := eachDelVal.StaffId;

Fname := eachDelVal.FirstName;

surname := eachDelVal.Surname;

DelProced := eachDelVal.DeliveryCount;

dbms\_output.put\_line('---------------------------------------------');

dbms\_output.put\_line('DRIVER ID : '|| driverID );

dbms\_output.put\_line('FIRST NAME : '|| Fname);

dbms\_output.put\_line('SURNAME : ' || surname);

dbms\_output.put\_line('DELIVERIES PROCESSED : '|| DelProced);

dbms\_output.put\_line('---------------------------------------------');

END LOOP;

END;

## The output of Question 5.1

A close-up of a document

Description automatically generated

Figure : Most of Deliveries Processed

5.2

A PL/SQL block is divided into three sections: the Declaration section, the Execution section, and the Exception handling section based on (Oracle Tutorial, 2023).

1. Declaration Section: This is where variables, constants, cursors, and other elements are declared for use in the block. In your query, this section includes the declaration of the variables `driverID`, `Fname`, `surname`, and `DelProced`, as well as the cursor `DelVal`.

sql

DECLARE

driverID StaffDelivery.StaffId %TYPE;

Fname StaffDelivery.FirstName %TYPE;

surname StaffDelivery.Surname %TYPE;

DelProced DeliveryItems.DeliveryItemId %TYPE;

CURSOR DelVal IS ...

```

2. Execution Section: This is where the main logic of the program is written. It's where SQL statements are executed and where control structures (like loops or conditional statements) are used. In my query, this section includes the loop that fetches each row from the `DelVal` cursor and outputs the details of each delivery.

```sql

BEGIN

FOR eachDelVal IN DelVal

LOOP

driverID := eachDelVal.StaffId;

Fname := eachDelVal.FirstName;

surname := eachDelVal.Surname;

DelProced := eachDelVal.DeliveryCount;

dbms\_output.put\_line('---------------------------------------------');

dbms\_output.put\_line('DRIVER ID : '|| driverID );

dbms\_output.put\_line('FIRST NAME : '|| Fname);

dbms\_output.put\_line('SURNAME : ' || surname);

dbms\_output.put\_line('DELIVERIES PROCESSED : '|| DelProced);

dbms\_output.put\_line('---------------------------------------------');

END LOOP;

END;

```

3. Exception Handling Section: This is where exceptions (errors) are handled. If an error occurs in the execution section, control is passed to the exception handling section.

In summary, My PL/SQL block includes a declaration section where variables and a cursor are declared, and an execution section where these elements are used to fetch and output data.

5.3.1

## View in SQL:

A View in SQL is a virtual table based on the result-set of an SQL statement. It contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database based on (Navlani, 2019).

Views are used for the following reasons:

Security: Views can display specific columns from the table based on the set conditions. So, a user can be given permission to access the data through the view, without granting the permissions to directly access the base table of the view.

Complexity: A view can simplify the complexity of a query because it can encapsulate complicated SQL queries into a single view, which you can then treat as a regular table.

Reusability and Consistency: Once a view is created, it can be reused many times, providing a consistent result set structure every time it is queried

--Question 5.3.2

CREATE OR REPLACE VIEW mostdeliveries AS

SELECT

-- Selecting the staff ID

StaffDelivery.StaffId,

-- Selecting the first name of the staff

StaffDelivery.FirstName,

-- Selecting the surname of the staff

StaffDelivery.Surname,

-- Counting the number of deliveries made by each staff member

COUNT(DeliveryItems.StaffId) AS DeliveryCount

FROM

-- Specifying the 'StaffDelivery' table as the source of data

StaffDelivery

-- Joining 'StaffDelivery' with 'DeliveryItems' on 'StaffId'

INNER JOIN DeliveryItems ON StaffDelivery.StaffId = DeliveryItems.StaffId

--Joining 'DriverDelivery' with 'DeliveryItems' on 'DeliveryItemsID'

INNER JOIN DriverDelivery ON DeliveryItems.DeliveryItemId = DriverDelivery.DeliveryItemId

--Joining 'Vehicle' with 'DriverDelivery' on 'VinNumber'

INNER JOIN Vehicle ON Vehicle.VinNumber = DriverDelivery.VinNumber

-- Grouping the result by staff ID, first name, and surname

GROUP BY

StaffDelivery.StaffId,

StaffDelivery.FirstName,

StaffDelivery.Surname

-- Ordering the result by the count of deliveries in descending order

ORDER BY

DeliveryCount DESC;

## The output of Question 5.3.2



Figure : View Table Output

# Question 6

--QUESTION 6

SET SERVEROUTPUT ON;

DECLARE

name CustomersDelivery.FirstName %TYPE;

phone CustomersDelivery.PhoneNumber %TYPE;

email CustomersDelivery.Email %TYPE;

bills Billing.BillId %TYPE;

CURSOR rating IS

SELECT

CustomersDelivery.FirstName,

CustomersDelivery.Surname,

CustomersDelivery.PhoneNumber,

CustomersDelivery.Email,

COUNT(Billing.CustomerID ) AS numBills

FROM

CustomersDelivery

INNER JOIN Billing ON CustomersDelivery.CustomerId = Billing.CustomerID

GROUP BY

CustomersDelivery.FirstName,

CustomersDelivery.Surname,

CustomersDelivery.PhoneNumber,

CustomersDelivery.Email;

BEGIN

FOR eachRat IN rating

LOOP

name := eachRat.FirstName || ', ' || eachRat.Surname;

phone := eachRat.PhoneNumber;

email := eachRat.Email;

bills := eachRat.numBills;

IF(bills >=4) THEN

dbms\_output.put\_line('--------------------------------------------------') ;

dbms\_output.put\_line('CUSTOMER NAME :' || name);

dbms\_output.put\_line('CUSTOMER PHONE :' || phone) ;

dbms\_output.put\_line('CUSTOMER EMAIL :' || email) ;

dbms\_output.put\_line('NumberOfBills :' || bills || ('(ELITE)'));

dbms\_output.put\_line('Thank you for being our valued Customer');

dbms\_output.put\_line('--------------------------------------------------');

ELSE

dbms\_output.put\_line('--------------------------------------------------') ;

dbms\_output.put\_line('CUSTOMER NAME :' || name);

dbms\_output.put\_line('CUSTOMER PHONE :' || phone) ;

dbms\_output.put\_line('CUSTOMER EMAIL :' || email) ;

dbms\_output.put\_line('NumberOfBills :' || bills);

dbms\_output.put\_line('--------------------------------------------------');

END IF;

END LOOP;

END;

## The output of Question 6

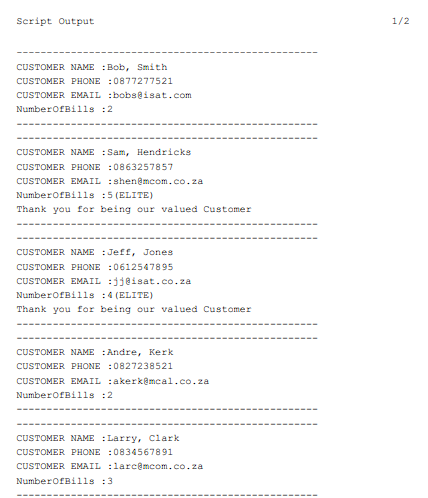


Figure : Output of Elite Bills Part 1

A screenshot of a computer

Description automatically generated

Figure : Output of Elite Bills Part 2

6.2

Here are two suggestions:

* Case Statement in PL/SQL (Tutorial Point Group, 2023)

You could use a CASE expression instead of an IF-THEN-ELSE statement to assign the customer rating based on the number of bills. This would make your code more concise and avoid repeating the dbms\_output statements for each rating .

* View Table in SQL (Navlani, 2019)

You could create a view that contains the customer name, phone, email, and number of bills, and then use a simple SELECT statement to query the view and display the results. This would make your code more modular and reusable, as you could access the view from other programs or queries without having to repeat the logic of joining the tables and counting the bills.