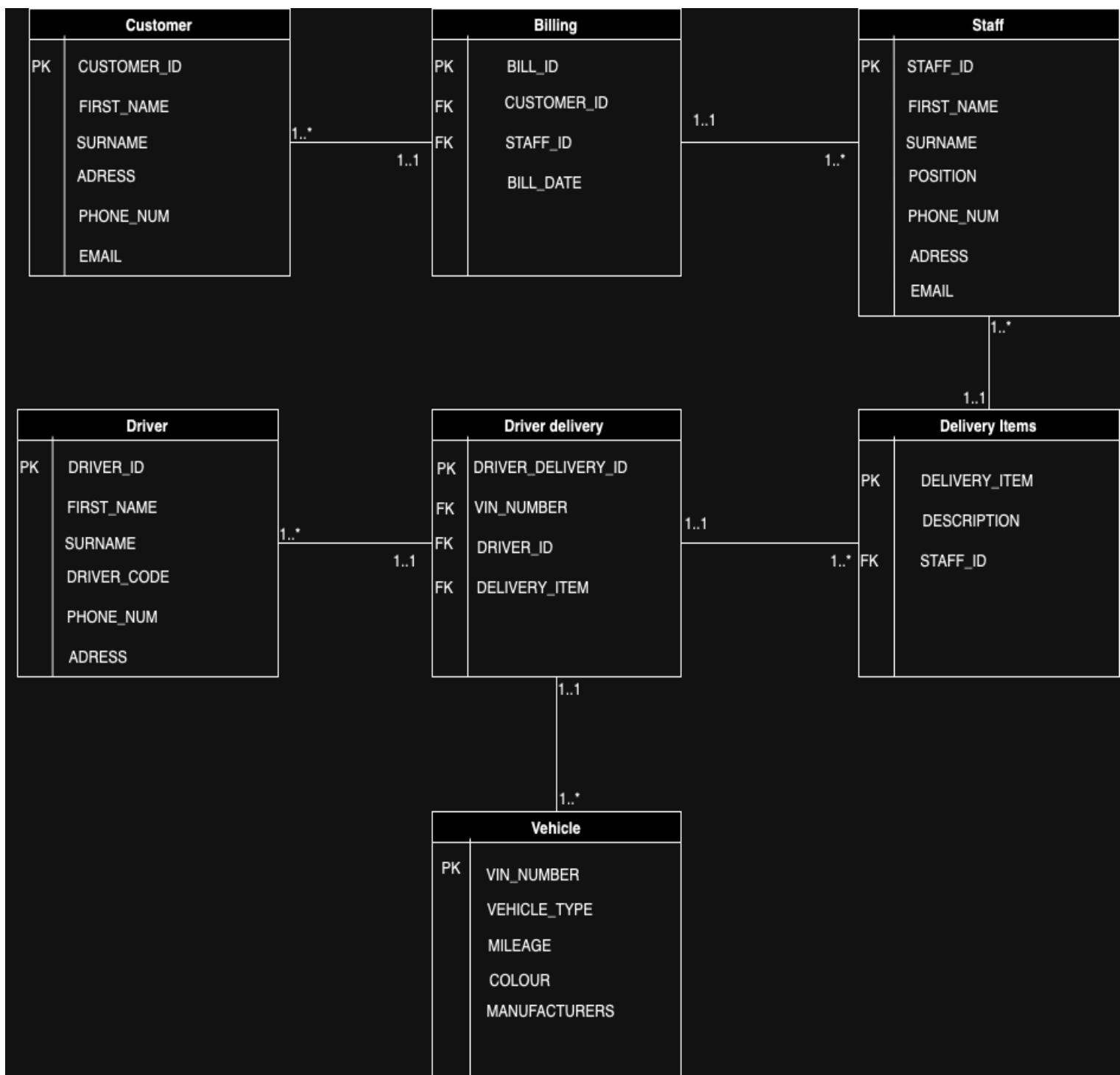


ABDD7311
Practical assignment 1

ST10359034
Emil fabel

1. ERD Diagram



Q2. Tables Creation

The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Oracle SQL Developer : C:\Users\lab_services_student\Desktop\ADDB7311_pa1_Q2.sql
- Menu Bar:** File Edit View Navigate Run Source Team Tools Window Help
- Toolbar:** Includes icons for New Connection, Open Connection, Save, Print, and others.
- Connections Sidebar:** Shows Oracle Connections (MyOradeConnection, Student) and Database Schema Service Connections.
- SQL Worksheet:** The main workspace contains the following SQL code:

```
-- Q2. Create the Customers table
CREATE TABLE Customers (
    CUSTOMER_ID NUMBER PRIMARY KEY,
    FIRST_NAME VARCHAR2(50),
    SURNAME VARCHAR2(50),
    ADDRESS VARCHAR2(100),
    PHONE_NUM VARCHAR2(15),
    EMAIL VARCHAR2(100)
);

-- 2. Create the Staff table
CREATE TABLE Staff (
    STAFF_ID NUMBER PRIMARY KEY,
    FIRST_NAME VARCHAR2(50),
    SURNAME VARCHAR2(50),
    POSITION VARCHAR2(50),
    PHONE_NUM VARCHAR2(15),
    ADDRESS VARCHAR2(100),
    EMAIL VARCHAR2(100)
);

-- 3. Create the Vehicle table
CREATE TABLE Vehicle (
    VIN_NUMBER VARCHAR2(50) PRIMARY KEY,
    VEHICLE_TYPE VARCHAR2(50),
    MILEAGE NUMBER,
    COLOUR VARCHAR2(50),
    MANUFACTURER VARCHAR2(50)
);

-- 4. Create the Delivery_Items table
CREATE TABLE Delivery_Items (
    DELIVERY_ITEM_ID NUMBER PRIMARY KEY,
    DESCRIPTION VARCHAR2(100),
    STAFF_ID NUMBER,
    CONSTRAINT fk_staff_delivery FOREIGN KEY (STAFF_ID) REFERENCES Staff(STAFF_ID)
);
```

The code is organized into four sections, each starting with a double-dash comment and followed by a CREATE TABLE statement with its respective columns and constraints. The Oracle connection 'MyOradeConnection' is selected in the connections sidebar.

Oracle SQL Developer : C:\Users\lab_services_student\Desktop\ADDB7311_pa1_Q2.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections MyOracleConnection Student Database Schema Service Connections

SQL Worksheet History

Worksheet Query Builder

```
DELIVERY_ITEM_ID NUMBER PRIMARY KEY,
DESCRIPTION VARCHAR2(100),
STAFF_ID NUMBER,
CONSTRAINT fk_staff_delivery FOREIGN KEY (STAFF_ID) REFERENCES Staff(STAFF_ID)
);

-- 5. Create the Drivers table
CREATE TABLE Drivers (
    DRIVER_ID NUMBER PRIMARY KEY,
    FIRST_NAME VARCHAR2(50),
    SURNAME VARCHAR2(50),
    DRIVER_CODE VARCHAR2(10),
    PHONE_NUM VARCHAR2(15),
    ADDRESS VARCHAR2(100)
);

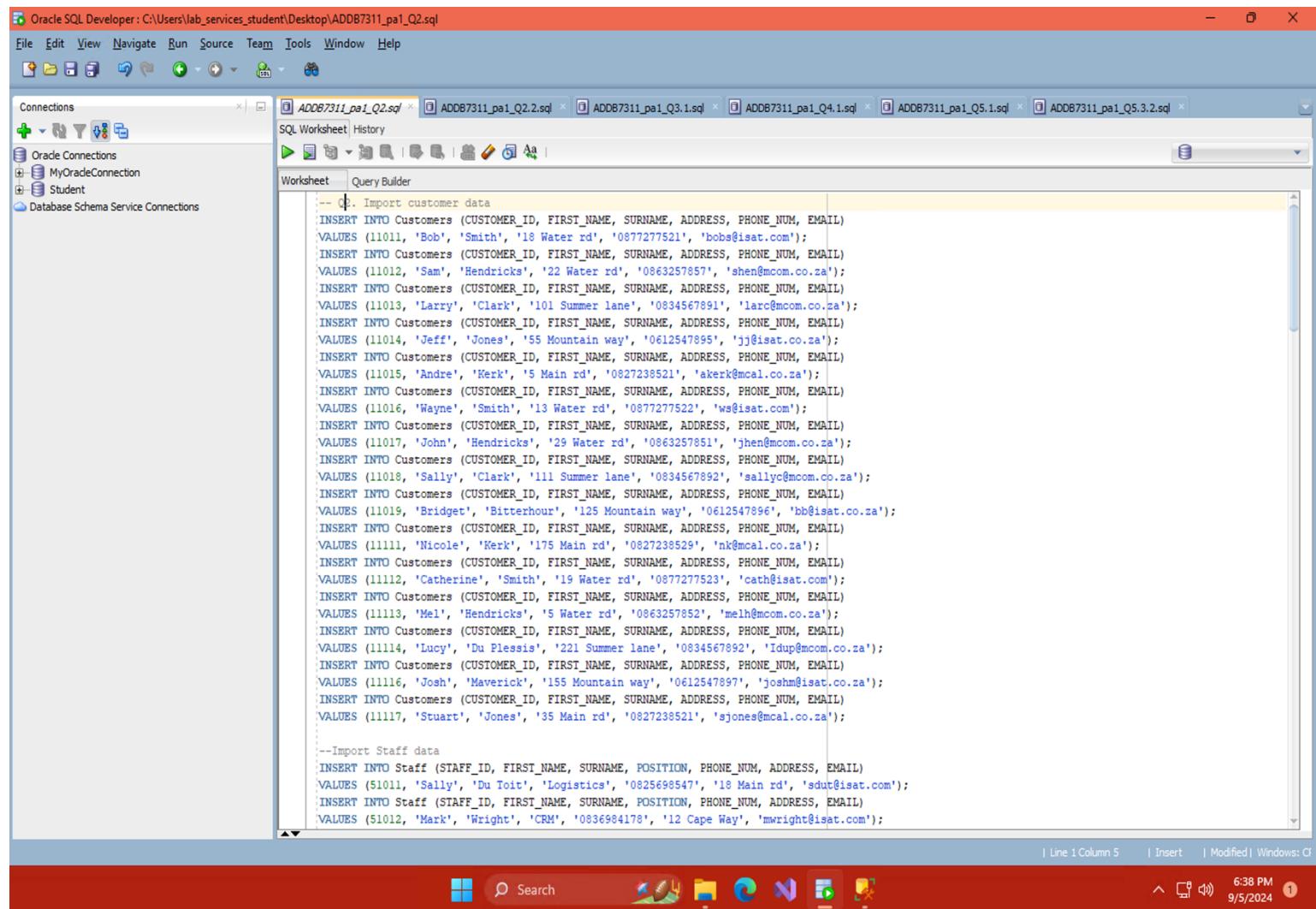
-- 6. Create the Driver_Deliveries table
CREATE TABLE Driver_Deliveries (
    DRIVER_DELIVERY_ID NUMBER PRIMARY KEY,
    VIN_NUMBER VARCHAR2(50),
    DRIVER_ID NUMBER,
    DELIVERY_ITEM_ID NUMBER,
    CONSTRAINT fk_vehicle FOREIGN KEY (VIN_NUMBER) REFERENCES Vehicle(VIN_NUMBER),
    CONSTRAINT fk_driver FOREIGN KEY (DRIVER_ID) REFERENCES Drivers(DRIVER_ID),
    CONSTRAINT fk_delivery_item FOREIGN KEY (DELIVERY_ITEM_ID) REFERENCES Delivery_Items(DELIVERY_ITEM_ID)
);

-- 7. Create the Billing table
CREATE TABLE Billing (
    BILL_ID NUMBER PRIMARY KEY,
    CUSTOMER_ID NUMBER,
    STAFF_ID NUMBER,
    BILL_DATE DATE,
    CONSTRAINT fk_customer FOREIGN KEY (CUSTOMER_ID) REFERENCES Customers(CUSTOMER_ID),
    CONSTRAINT fk_staff_billing FOREIGN KEY (STAFF_ID) REFERENCES Staff(STAFF_ID)
);
```

Line 1 Column 6 | Insert | Modified | Windows: C:\

Search 6:36 PM 9/5/2024

Import of Data Into Created Tables



The screenshot shows the Oracle SQL Developer interface. The title bar indicates the file path: C:\Users\lab_services_student\Desktop\ADDB7311_pa1_Q2.sql. The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The Connections sidebar shows 'MyOracleConnection' and 'Student'. The central workspace has tabs for ADDB7311_pa1_Q2.sql, ADDB7311_pa1_Q2.2.sql, ADDB7311_pa1_Q3.1.sql, ADDB7311_pa1_Q4.1.sql, ADDB7311_pa1_Q5.1.sql, and ADDB7311_pa1_Q5.3.sql. The current tab is ADDB7311_pa1_Q2.sql. The worksheet contains a large SQL script for importing customer and staff data into tables. The script uses the INSERT INTO statement to insert rows into the CUSTOMERS and STAFF tables. The customer data includes columns like CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, and EMAIL. The staff data includes columns like STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, and ADDRESS. The script is enclosed in two dashed boxes: one for customer data and one for staff data.

```
-- Q2. Import customer data
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11011, 'Bob', 'Smith', '18 Water rd', '0877277521', 'bobs@isat.com');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11012, 'Sam', 'Hendricks', '22 Water rd', '0863257857', 'shen@mcom.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11013, 'Larry', 'Clark', '101 Summer lane', '0834567891', 'larc@mcom.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11014, 'Jeff', 'Jones', '55 Mountain way', '0612547895', 'jjj@isat.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11015, 'Andre', 'Kerk', '5 Main rd', '0827238521', 'akerk@mcal.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11016, 'Wayne', 'Smith', '13 Water rd', '0877277522', 'ws@isat.com');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11017, 'John', 'Hendricks', '29 Water rd', '0863257851', 'jhen@mcom.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11018, 'Sally', 'Clark', '111 Summer lane', '0834567892', 'sallyc@mcom.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11019, 'Bridget', 'Bitterhour', '125 Mountain way', '0612547896', 'bb@isat.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11020, 'Nicole', 'Kerk', '175 Main rd', '0827238529', 'nk@mcal.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11021, 'Catherine', 'Smith', '19 Water rd', '0877277523', 'cath@isat.com');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11022, 'Mel', 'Hendricks', '5 Water rd', '0863257852', 'meli@mcom.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11023, 'Lucy', 'Du Plessis', '221 Summer lane', '0834567892', 'lup@mcom.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11024, 'Josh', 'Maverick', '155 Mountain way', '0612547897', 'joshm@isat.co.za');
INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
VALUES (11025, 'Stuart', 'Jones', '35 Main rd', '0827238521', 'sjones@mcal.co.za');

--Import Staff data
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51011, 'Sally', 'Du Toit', 'Logistics', '0825698547', 'sdu@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51012, 'Mark', 'Wright', 'CRM', '0836984178', 'mwright@isat.com');
```

Oracle SQL Developer : C:\Users\lab_services_student\Desktop\ADD87311_pa1_Q2.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

+ Oracle Connections
MyOracleConnection
Student
Database Schema Service Connections

SQL Worksheet History

Worksheet Query Builder

```
| INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
| VALUES (11113, 'Mel', 'Hendricks', '5 Water rd', '0863257852', 'melh@mcom.co.za');
| INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
| VALUES (11114, 'Lucy', 'Du Plessis', '221 Summer lane', '0834567892', 'Idup@mcom.co.za');
| INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
| VALUES (11116, 'Josh', 'Maverick', '155 Mountain way', '0612547897', 'joshm@isat.co.za');
| INSERT INTO Customers (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, PHONE_NUM, EMAIL)
| VALUES (11117, 'Stuart', 'Jones', '35 Main rd', '0827238521', 'sjones@mcal.co.za');

--Import Staff data
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51011, 'Sally', 'Du Toit', 'Logistics', '0825698547', '18 Main rd', 'sdut@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51012, 'Mark', 'Wright', 'CRM', '0836984178', '12 Cape Way', 'mwright@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51013, 'Harry', 'Sheen', 'Logistics', '0725648965', '15 Water Street', 'hsheen@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51014, 'Jabu', 'Xolani', 'Logistics', '0823116598', '18 White Lane', 'jxo@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51015, 'Roberto', 'Henry', 'Packaging', '0783521451', '55 Cape Street', 'rhenry@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51016, 'Pat', 'Durant', 'Logistics', '0825698542', '1 Main rd', 'pd@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51017, 'Steve', 'Maritz', 'CRM', '0836984173', '2 Cape Way', 'sm@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51018, 'Maxwell', 'Dube', 'Logistics', '0725648964', '5 Water Street', 'max@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51019, 'Shane', 'Mane', 'Logistics', '0823116595', '8 White Lane', 'smane@isat.com');
INSERT INTO Staff (STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL)
VALUES (51111, 'Bob', 'Truth', 'Packaging', '0783521456', '35 Cape Street', 'btruth@isat.com');

--Import Vehicle data
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('12A55858541', 'Cutaway van chassis', 115352, 'RED', 'MAN');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('12A51858542', 'Flatbed truck', 315856, 'BLUE', 'ISUZU');
```

Line 1 Column 5 | Insert | Modified | Windows: C:\

6:38 PM 9/5/2024

Oracle SQL Developer : C:\Users\lab_services_student\Desktop\ADDB7311_pa1_Q2.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections My Oracle Connections My Oracle Connection Student Database Schema Service Connections

SQL Worksheet History

Worksheet Query Builder

```
--Import Vehicle data
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA55858541', 'Cutaway van chassis', 115352, 'RED', 'MAN');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA51858542', 'Flatbed truck', 315856, 'BLUE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('17A35858543', 'Medium Standard Truck', 789587, 'SILVER', 'MAN');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA15851545', 'Flatbed truck', 555050, 'WHITE', 'TATA');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA35868540', 'Cutaway van chassis', 79058, 'WHITE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA65858541', 'Cutaway van chassis', 215352, 'RED', 'MAN');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA61858542', 'Flatbed truck', 215856, 'BLUE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA65858543', 'Medium Standard Truck', 889587, 'SILVER', 'MERC');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA65851545', 'Flatbed truck', 155050, 'WHITE', 'MAN');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA65868540', 'Cutaway van chassis', 19058, 'WHITE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA75858541', 'Cutaway van chassis', 315352, 'RED', 'MAN');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA71858542', 'Flatbed truck', 115856, 'BLUE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA75858543', 'Medium Standard Truck', 989587, 'SILVER', 'MAN');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA17851545', 'Flatbed truck', 755050, 'WHITE', 'TATA');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA75868540', 'Cutaway van chassis', 29058, 'WHITE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA85858541', 'Cutaway van chassis', 515352, 'RED', 'MERC');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('1ZA81858542', 'Flatbed truck', 715856, 'BLUE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
```

| Line | Column 5 | Insert | Modified | Windows: C |

Search

6:38 PM 9/5/2024

Oracle SQL Developer : C:\Users\lab_services_student\Desktop\ADDB7311_pa1_Q2.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections MyOracleConnection Student Database Schema Service Connections

ADD87311_pa1_Q2.sql ADD87311_pa1_Q2.2.sql ADD87311_pa1_Q3.1.sql ADD87311_pa1_Q4.1.sql ADD87311_pa1_Q5.1.sql ADD87311_pa1_Q5.3.sql ADD87311_pa1_Q5.3.2.sql

SQL Worksheet History

Worksheet Query Builder

```
VIN ( 12A75068540 , Flatbed truck , 750000 , WHITE , ISUZU );
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('12A75068540', 'Cutaway van chassis', 29058, 'WHITE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('12A85858541', 'Cutaway van chassis', 515352, 'RED', 'MERC');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('12A85858542', 'Flatbed truck', 715856, 'BLUE', 'ISUZU');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('12A85858543', 'Medium Standard Truck', 789587, 'SILVER', 'MAN');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('12A85851545', 'Flatbed truck', 955050, 'WHITE', 'TATA');
INSERT INTO Vehicle (VIN_NUMBER, VEHICLE_TYPE, MILEAGE, COLOUR, MANUFACTURER)
VALUES ('12A85868540', 'Cutaway van chassis', 39058, 'WHITE', 'MERC');

--Import Delivery_Items data
INSERT INTO Delivery_Items (DELIVERY_ITEM_ID, DESCRIPTION, STAFF_ID)
VALUES (71011, 'House relocation', 51011);
INSERT INTO Delivery_Items (DELIVERY_ITEM_ID, DESCRIPTION, STAFF_ID)
VALUES (71012, 'Delivery of specialized consignments', 51017);
INSERT INTO Delivery_Items (DELIVERY_ITEM_ID, DESCRIPTION, STAFF_ID)
VALUES (71013, 'Delivery of specialized consignments', 51015);
INSERT INTO Delivery_Items (DELIVERY_ITEM_ID, DESCRIPTION, STAFF_ID)
VALUES (71014, 'Office relocation', 51012);
INSERT INTO Delivery_Items (DELIVERY_ITEM_ID, DESCRIPTION, STAFF_ID)
VALUES (71015, 'Delivery of specialized consignments', 51014);

--Import Drivers data
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');
```

| Line 1 Column 5 | Insert | Modified | Windows: C:\

Search         6:38 PM 9/5/2024

Oracle SQL Developer : C:\Users\lab_services_student\Desktop\ADD87311_pa1_Q2.sql

File Edit View Navigate Run Source Team Window Help

Connections ADD87311_pa1_Q2.sql ADD87311_pa1_Q2.2.sql ADD87311_pa1_Q3.1.sql ADD87311_pa1_Q4.1.sql ADD87311_pa1_Q5.1.sql ADD87311_pa1_Q5.3.2.sql

SQL Worksheet History

Worksheet Query Builder

```
--Import Drivers_Deliveries data
INSERT INTO Driver_Deliveries (DRIVER_DELIVERY_ID, VIN_NUMBER, DRIVER_ID, DELIVERY_ITEM_ID)
VALUES (91011, '1ZA55858541', 81011, 71011);
INSERT INTO Driver_Deliveries (DRIVER_DELIVERY_ID, VIN_NUMBER, DRIVER_ID, DELIVERY_ITEM_ID)
VALUES (91012, '1ZA35858543', 81012, 71013);
INSERT INTO Driver_Deliveries (DRIVER_DELIVERY_ID, VIN_NUMBER, DRIVER_ID, DELIVERY_ITEM_ID)
VALUES (91013, '1ZA17851545', 81011, 71015);
INSERT INTO Driver_Deliveries (DRIVER_DELIVERY_ID, VIN_NUMBER, DRIVER_ID, DELIVERY_ITEM_ID)
VALUES (91014, '1ZA35868540', 81013, 71015);
INSERT INTO Driver_Deliveries (DRIVER_DELIVERY_ID, VIN_NUMBER, DRIVER_ID, DELIVERY_ITEM_ID)
VALUES (91015, '1ZA15851545', 81014, 71012);

--Import Billing data
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (800, 11011, 51011, TO_DATE('06-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (801, 11012, 51013, TO_DATE('07-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (802, 11014, 51015, TO_DATE('10-Nov-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (803, 11015, 51012, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (804, 11013, 51014, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (805, 11111, 51011, TO_DATE('06-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (806, 11012, 51013, TO_DATE('07-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (807, 11014, 51015, TO_DATE('10-Nov-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (808, 11015, 51012, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (809, 11113, 51018, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (810, 11011, 51011, TO_DATE('06-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (811, 11012, 51013, TO_DATE('07-Sep-22', 'DD-Mon-YY'));
```

Line 1 Column 5 | Insert | Modified | Windows: C

Search 6:38 PM 9/5/2024

Oracle SQL Developer : C:\Users\lab_services_student\Desktop\ADDB7311_pa1_Q2.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections ADDB7311_pa1_Q2.sql ADDB7311_pa1_Q2.2.sql ADDB7311_pa1_Q3.1.sql ADDB7311_pa1_Q4.1.sql ADDB7311_pa1_Q5.1.sql ADDB7311_pa1_Q5.3.2.sql

SQL Worksheet History

Worksheet Query Builder

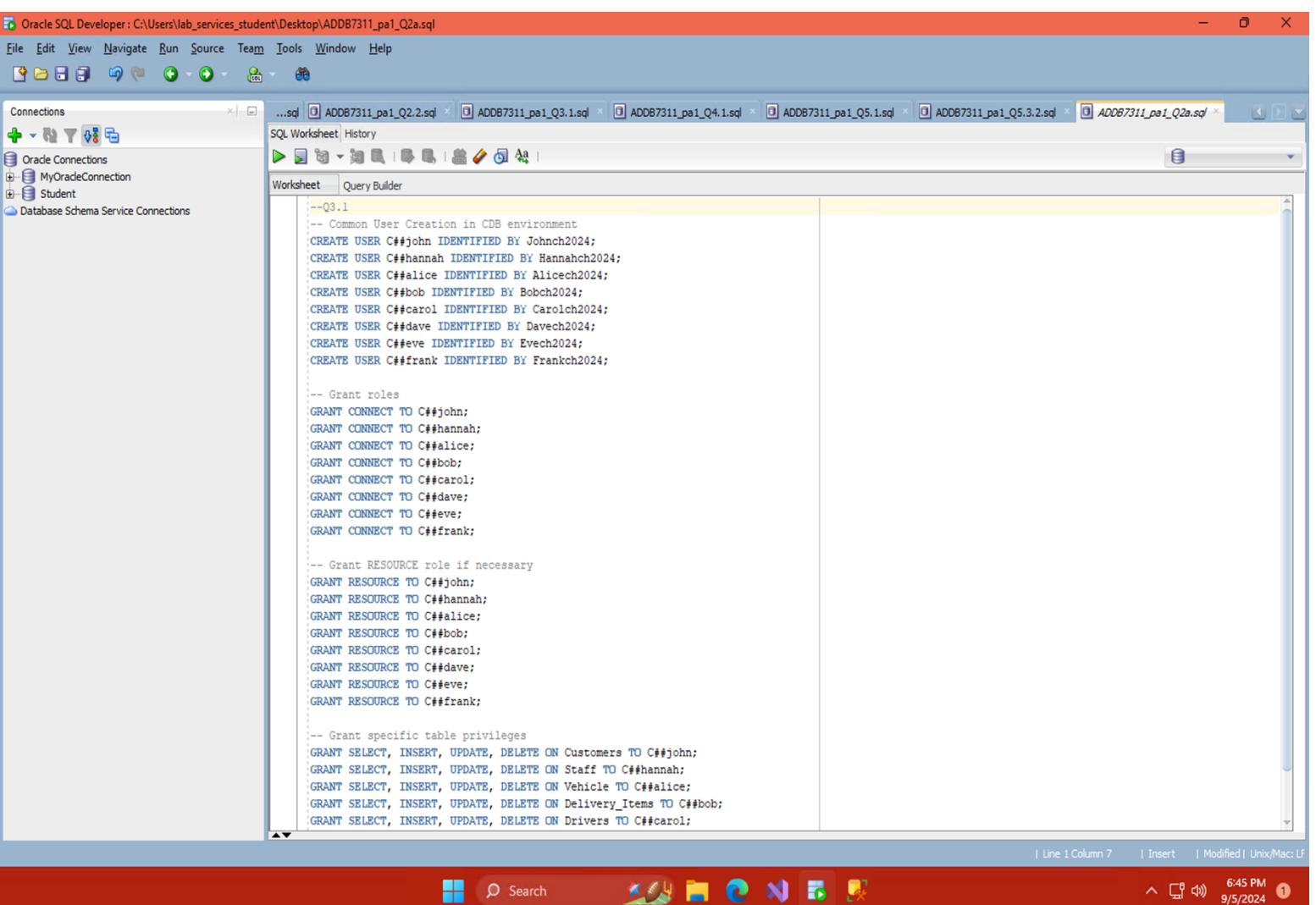
```
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (804, 11013, 51014, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (805, 11111, 51011, TO_DATE('06-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (806, 11012, 51013, TO_DATE('07-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (807, 11014, 51015, TO_DATE('10-Nov-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (808, 11015, 51012, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (809, 11113, 51018, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (810, 11011, 51011, TO_DATE('06-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (811, 11012, 51013, TO_DATE('07-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (812, 11014, 51016, TO_DATE('10-Nov-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (813, 11117, 51012, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (814, 11013, 51014, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (815, 11012, 51111, TO_DATE('06-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (816, 11012, 51019, TO_DATE('07-Sep-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (817, 11014, 51015, TO_DATE('10-Nov-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (818, 11112, 51012, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (819, 11013, 51014, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (820, 11116, 51019, TO_DATE('09-Dec-22', 'DD-Mon-YY'));
```

Line 1 Column 5 | Insert | Modified | Windows: C:\

Search

6:38 PM 9/5/2024

Q3.1 Creating and Granting Privileges To Users



The screenshot shows the Oracle SQL Developer interface with a worksheet containing SQL code. The code is as follows:

```
--Q3.1
-- Common User Creation in CDB environment
CREATE USER C##john IDENTIFIED BY Johnch2024;
CREATE USER C##hannah IDENTIFIED BY Hannahch2024;
CREATE USER C##alice IDENTIFIED BY Alicech2024;
CREATE USER C##bob IDENTIFIED BY Bobch2024;
CREATE USER C##carol IDENTIFIED BY Carolch2024;
CREATE USER C##dave IDENTIFIED BY Davech2024;
CREATE USER C##eve IDENTIFIED BY Evech2024;
CREATE USER C##frank IDENTIFIED BY Frankch2024;

-- Grant roles
GRANT CONNECT TO C##john;
GRANT CONNECT TO C##hannah;
GRANT CONNECT TO C##alice;
GRANT CONNECT TO C##bob;
GRANT CONNECT TO C##carol;
GRANT CONNECT TO C##dave;
GRANT CONNECT TO C##eve;
GRANT CONNECT TO C##frank;

-- Grant RESOURCE role if necessary
GRANT RESOURCE TO C##john;
GRANT RESOURCE TO C##hannah;
GRANT RESOURCE TO C##alice;
GRANT RESOURCE TO C##bob;
GRANT RESOURCE TO C##carol;
GRANT RESOURCE TO C##dave;
GRANT RESOURCE TO C##eve;
GRANT RESOURCE TO C##frank;

-- Grant specific table privileges
GRANT SELECT, INSERT, UPDATE, DELETE ON Customers TO C##john;
GRANT SELECT, INSERT, UPDATE, DELETE ON Staff TO C##hannah;
GRANT SELECT, INSERT, UPDATE, DELETE ON Vehicle TO C##alice;
GRANT SELECT, INSERT, UPDATE, DELETE ON Delivery_Items TO C##bob;
GRANT SELECT, INSERT, UPDATE, DELETE ON Drivers TO C##carol;
```

Oracle SQL Developer : C:\Users\lab_services_student\Desktop\ADDB7311_pa1_Q2a.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections MyOracleConnection Student Database Schema Service Connections

SQL Worksheet History Worksheet Query Builder

```
CREATE USER C##hannah IDENTIFIED BY Hannahch2024;
CREATE USER C##alice IDENTIFIED BY Alicech2024;
CREATE USER C##bob IDENTIFIED BY Bobch2024;
CREATE USER C##carol IDENTIFIED BY Carolch2024;
CREATE USER C##dave IDENTIFIED BY Davech2024;
CREATE USER C##eve IDENTIFIED BY Evech2024;
CREATE USER C##frank IDENTIFIED BY Frankch2024;

-- Grant roles
GRANT CONNECT TO C##john;
GRANT CONNECT TO C##hannah;
GRANT CONNECT TO C##alice;
GRANT CONNECT TO C##bob;
GRANT CONNECT TO C##carol;
GRANT CONNECT TO C##dave;
GRANT CONNECT TO C##eve;
GRANT CONNECT TO C##frank;

-- Grant RESOURCE role if necessary
GRANT RESOURCE TO C##john;
GRANT RESOURCE TO C##hannah;
GRANT RESOURCE TO C##alice;
GRANT RESOURCE TO C##bob;
GRANT RESOURCE TO C##carol;
GRANT RESOURCE TO C##dave;
GRANT RESOURCE TO C##eve;
GRANT RESOURCE TO C##frank;

-- Grant specific table privileges
GRANT SELECT, INSERT, UPDATE, DELETE ON Customers TO C##john;
GRANT SELECT, INSERT, UPDATE, DELETE ON Staff TO C##hannah;
GRANT SELECT, INSERT, UPDATE, DELETE ON Vehicle TO C##alice;
GRANT SELECT, INSERT, UPDATE, DELETE ON Delivery_Items TO C##bob;
GRANT SELECT, INSERT, UPDATE, DELETE ON Drivers TO C##carol;
GRANT SELECT, INSERT, UPDATE, DELETE ON Driver_Deliveries TO C##dave;
GRANT SELECT, INSERT, UPDATE, DELETE ON Billing TO C##eve;
GRANT SELECT, INSERT, UPDATE, DELETE ON Billing TO C##frank;
```

Line 1 Column 7 | Insert | Modified | Unix/Mac: LF

Search 9/5/2024 6:45 PM

Q3.2

Importance of Separation of Duties

Separation of duties is crucial for security and data integrity. It lowers the possibility of accidental or deliberate data breaches or misuse by guaranteeing that no single person has complete control over a system's functions. To provide checks and balances in database usage, "John" may be in charge of data retrieval and "Hannah" of data input.

(Imperva, 2024)

Q4.1 Driver Report of Vehicle with Lower than 80000 Miles

The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** MyOracleConnection is selected.
- Worksheet:** The code for Q4.1 is pasted into the worksheet. It includes a declaration section with variables for driver first name, surname, driver code, vehicle VIN number, and mileage. The main logic selects the driver with the highest mileage vehicle under 80,000 miles, joining the Driver table with Driver_Deliveries and Vehicle tables, and ordering by mileage descending. It also includes a section to display results in the same format as Q5.1 using DBMS_OUTPUT.PUT_LINE. The code ends with a fetch first row only and a slash to indicate the end of the procedure.
- Query Result:** The output shows the results for the driver Jono Mvuyisi, with driver code E1, VIN number 1ZA35060540, and mileage 79058.
- Status:** The message "PL/SQL procedure successfully completed." is displayed at the bottom of the query result pane.
- System Bar:** The system bar at the bottom right shows the date and time as 7:29 PM on 9/5/2024.

```
--Q4.1
SET SERVEROUTPUT ON;

DECLARE
    v_first_name Driver.FIRST_NAME%TYPE;
    v_surname Driver.SURNAME%TYPE;
    v_driver_code Driver.DRIVER_CODE%TYPE;
    v_vin_number Vehicle.VIN_NUMBER%TYPE;
    v_mileage Vehicle.MILEAGE%TYPE;
BEGIN
    -- Select the driver with the highest mileage vehicle under 80,000 miles
    SELECT d.FIRST_NAME, d.SURNAME, d.DRIVER_CODE, v.VIN_NUMBER, v.MILEAGE
    INTO v_first_name, v_surname, v_driver_code, v_vin_number, v_mileage
    FROM Driver d
    JOIN Driver_Deliveries dd ON d.DRIVER_ID = dd.DRIVER_ID
    JOIN Vehicle v ON dd.VIN_NUMBER = v.VIN_NUMBER
    WHERE v.MILEAGE < 80000
    ORDER BY v.MILEAGE DESC
    FETCH FIRST ROW ONLY;

    -- Display the results in the same format as Q5.1
    DBMS_OUTPUT.PUT_LINE('DRIVER: ' || v_first_name || ', ' || v_surname);
    DBMS_OUTPUT.PUT_LINE('CODE: ' || v_driver_code);
    DBMS_OUTPUT.PUT_LINE('VIN NUMBER: ' || v_vin_number);
    DBMS_OUTPUT.PUT_LINE('MILEAGE: ' || v_mileage);
END;
/

```

DRIVER: Jono, Mvuyisi
CODE: E1
VIN NUMBER: 1ZA35060540
MILEAGE: 79058

PL/SQL procedure successfully completed.

Q4.2

Flat File Database Model:

Simplicity: For small applications, setting up a single table or file containing all the data is simple.

Data redundancy: Since everything is stored in one location, there is a high chance of duplicate data.

Limited Query Capabilities: It can be hard to manage relationships or run sophisticated queries.

Data Integrity: Because there are no restrictions, it is difficult to keep correct and consistent data.

Relational Database Model:

With structured data management, redundant information is cut down on and consistency is enhanced by grouping data into related tables.

Complex Queries: This feature is essential for evaluating driver performance or customer interactions as it allows complex queries to link data and obtain detailed insights.

Data integrity reduces anomalies and ensures data accuracy by enforcing restrictions and connections.

Scalability: The ability to effectively handle expanding volumes of data and adjust to changing business requirements.

(Guldenpfennig, 2023) (IBM, 2022) (Hughes, et al., 2024)

Q5.1 Report of Staff Who Has Processed Most Deliveries

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : MyOracleConnection". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for file operations. The Connections sidebar shows "MyOracleConnection" and "Student". The Worksheet tab is active, displaying a PL/SQL block:

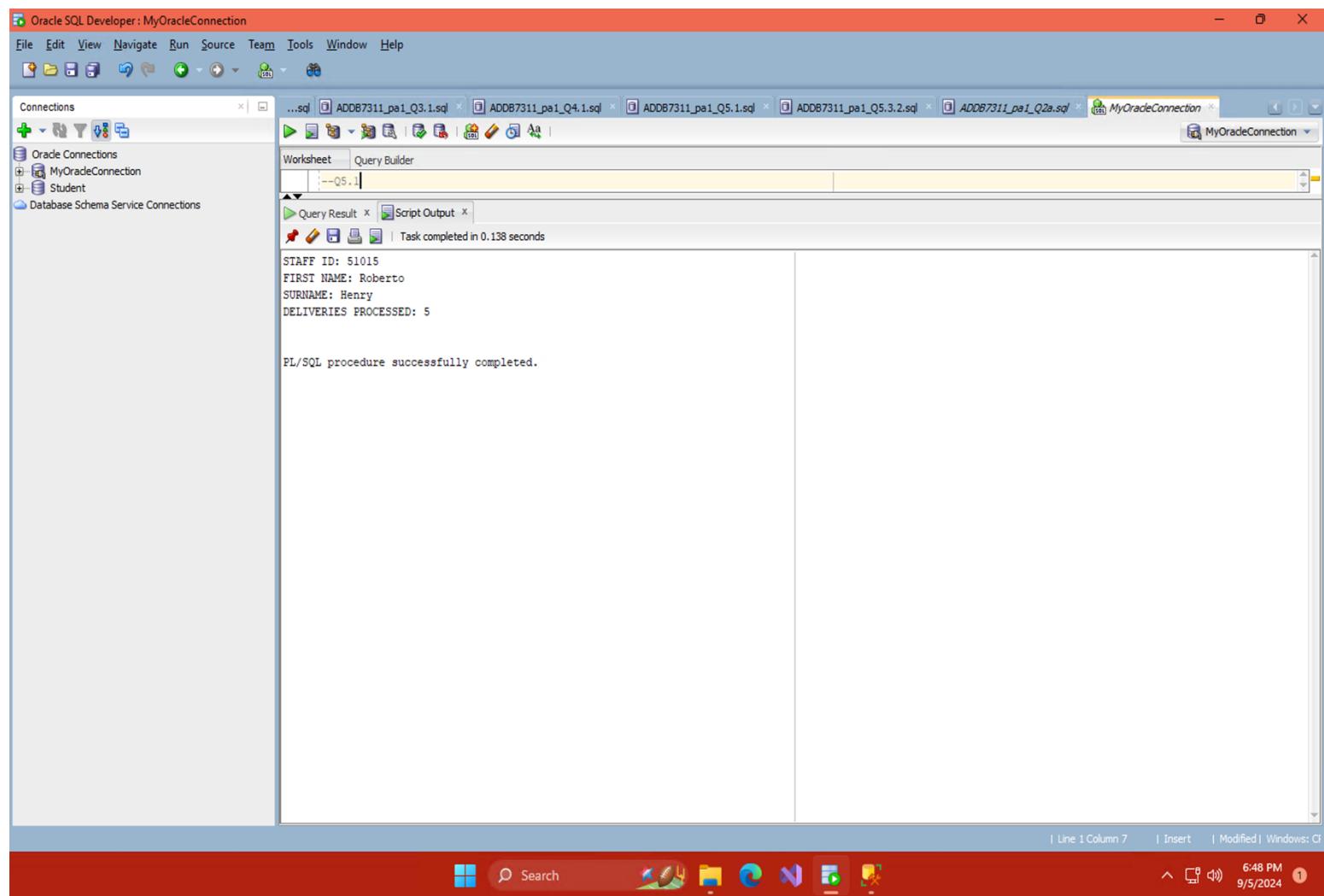
```
--Q5.1

SET SERVEROUTPUT ON;

DECLARE
    v_staff_id Staff.STAFF_ID%TYPE;
    v_first_name Staff.FIRST_NAME%TYPE;
    v_surname Staff.SURNAME%TYPE;
    v_delivery_count NUMBER;
BEGIN
    -- Select the staff member with the most deliveries
    SELECT s.STAFF_ID, s.FIRST_NAME, s.SURNAME, COUNT(b.STAFF_ID) AS DELIVERY_COUNT
    INTO v_staff_id, v_first_name, v_surname, v_delivery_count
    FROM Billing b
    JOIN Staff s ON b.STAFF_ID = s.STAFF_ID
    GROUP BY s.STAFF_ID, s.FIRST_NAME, s.SURNAME
    ORDER BY DELIVERY_COUNT DESC
    FETCH FIRST ROW ONLY;

    DBMS_OUTPUT.PUT_LINE('STAFF ID: ' || v_staff_id);
    DBMS_OUTPUT.PUT_LINE('FIRST NAME: ' || v_first_name);
    DBMS_OUTPUT.PUT_LINE('SURNAME: ' || v_surname);
    DBMS_OUTPUT.PUT_LINE('DELIVERIES PROCESSED: ' || v_delivery_count);
END;
/
```

The Query Result window below shows the output: "STAFF ID: 51015". The status bar at the bottom indicates "Task completed in 0.138 seconds". The system tray shows icons for Task View, Search, File Explorer, Control Panel, Network, and Task Manager. The date and time are 9/5/2024 6:48 PM.



Q5.2

Explanation of PL/SQL Components

Declaration:

Declares variables to store data fetched from the database.
As shown under "DECLARE"

Execution:

Contains the SQL query and logic to process data. As
shown under "BEGIN"

Exception Handling:

Handles errors, though not shown here in the code but can
be implemented with an "EXCEPTION" block

(KHAN, 2023)

Q5.3.1

How a View Works:

In a database, a view is a hypothetical table that is produced by a saved query. It specifies how data from one or more tables should be presented; it does not store data in and of itself. The database runs the underlying query when you query a view in order to get and display the data as though it were a table. Without having to comprehend the complexities of the underlying tables and joins, users can interact with complicated data structures through an easier-to-use interface thanks to this abstraction.

Benefits:

By combining complicated queries into a single, reusable structure, views streamline data access. This covers up the difficulty of data retrieval while guaranteeing reliable and safe access to data. You can restrict user access to sensitive data, display particular subsets of data, and enhance usability by utilising views instead of complicated SQL statements. All things considered, perspectives improve productivity and security by streamlining data management and access.

(Navlani, 2019)

Q5.3.2 Modified Code That Uses a View and Added Comments

The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** MyOracleConnection is selected.
- Worksheet:** The code is being run in the Worksheet tab.
- Code Content:**

```
--Q5.3.2
-- Create or replace the view named Staff_Delivery_Count
-- This view aggregates the delivery count for each staff member based on the Billing table
CREATE OR REPLACE VIEW Staff_Delivery_Count AS
-- Select the staff ID, first name, surname, and the count of deliveries
SELECT s.STAFF_ID, -- Staff ID from the Staff table
       s.FIRST_NAME, -- First name of the staff member
       s.SURNAME, -- Surname of the staff member
       COUNT(b.STAFF_ID) AS DELIVERY_COUNT -- Count of deliveries processed by the staff member
  FROM Billing b -- Billing table which records each delivery
 JOIN Staff s ON b.STAFF_ID = s.STAFF_ID -- Join with the Staff table to get staff details
 GROUP BY s.STAFF_ID, s.FIRST_NAME, s.SURNAME; -- Group by staff ID, first name, and surname to aggregate delivery counts
/

```
- Script Output:** The output shows the view was created successfully.

```
View STAFF_DELIVERY_COUNT created.
```
- System Bar:** Shows the task completed in 0.086 seconds, the date (9/5/2024), and the time (7:40 PM).

Q6.1

Implicit Cursor Attributes

Motivation:

Oracle uses implicit cursor properties automatically when a SQL statement is run. Without explicitly specifying a cursor, they are helpful for rapidly receiving information about the outcome of SQL operations. This is especially useful for simple actions such as DELETE, UPDATE, and INSERT.

Why Make Use of:

Efficiency: They offer a rapid, non-code method of obtaining information on the execution of SQL statements.

Simplicity: Perfect for basic SQL operations that don't require intricate cursor management.

Automatic Management: Oracle manages cursors automatically, eliminating the need for human intervention.

Explicit Cursor Attributes

Motivation:

When a developer has to handle more complex searches that return numerous rows and needs more control over the result processing, they employ explicit cursors. With explicit cursors, you can control the cursor's lifecycle, which includes opening, fetching, and closing, and get rows one at a time.

Why Make Use of:

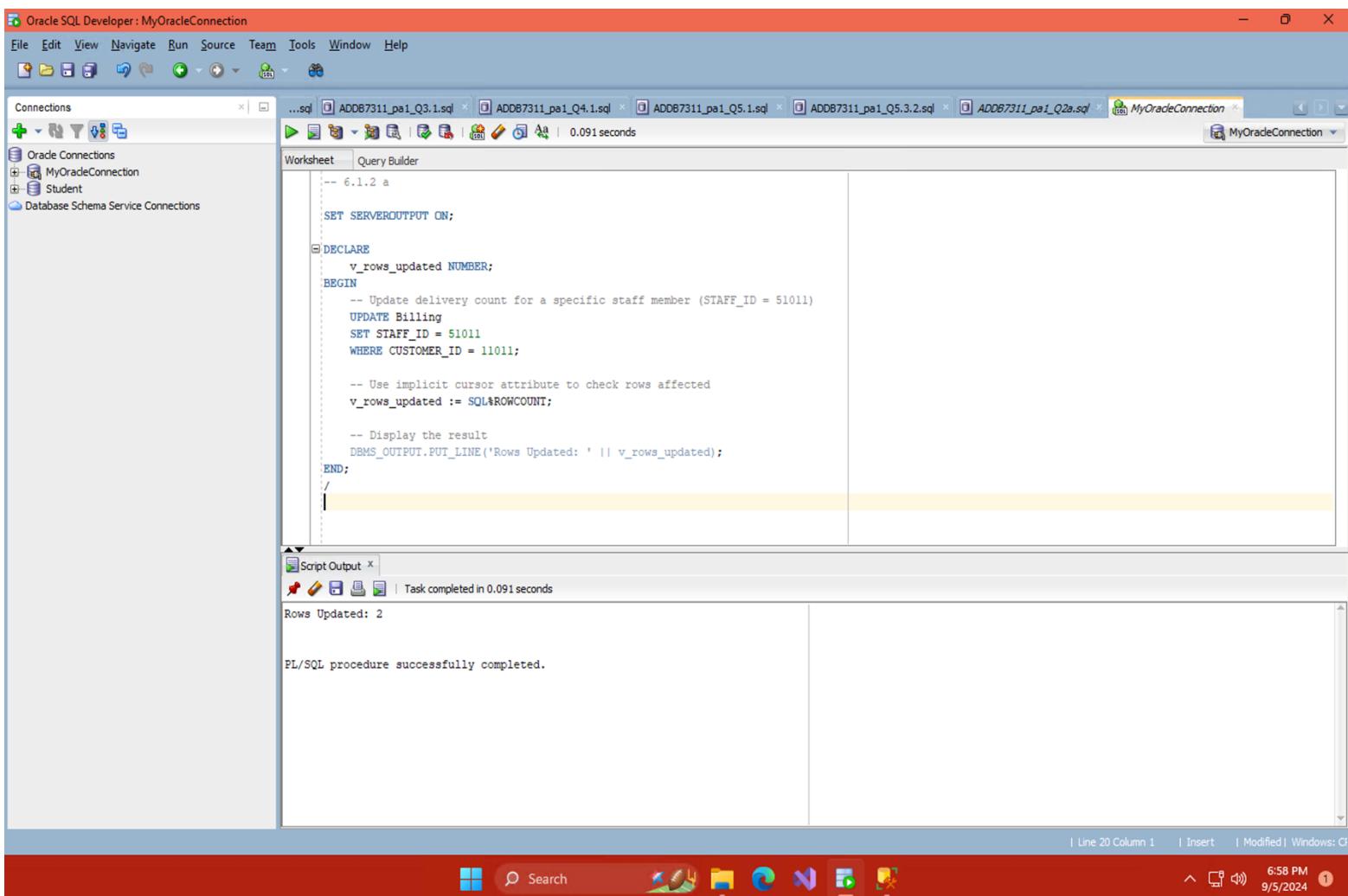
Control: Enables row-by-row processing and gives the user more control over how query results are processed.

Complex Queries: Crucial for managing queries that provide several rows and demand complex reasoning to be processed.

Flexibility: Enables management of cursor operations and query result determination by utilising cursor properties such as %FOUND, %NOTFOUND, and %ROWCOUNT.

(simranjenny84, 2022) (Javapoint.com, 2021)

Q6.2 Example Of Implicit Cursor Attributes



The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** Oracle Connections, MyOracleConnection, Student, Database Schema Service Connections.
- Worksheet:** The script being run is as follows:

```
-- 6.1.2 a
SET SERVEROUTPUT ON;

DECLARE
    v_rows_updated NUMBER;
BEGIN
    -- Update delivery count for a specific staff member (STAFF_ID = 51011)
    UPDATE Billing
    SET STAFF_ID = 51011
    WHERE CUSTOMER_ID = 11011;

    -- Use implicit cursor attribute to check rows affected
    v_rows_updated := SQL%ROWCOUNT;

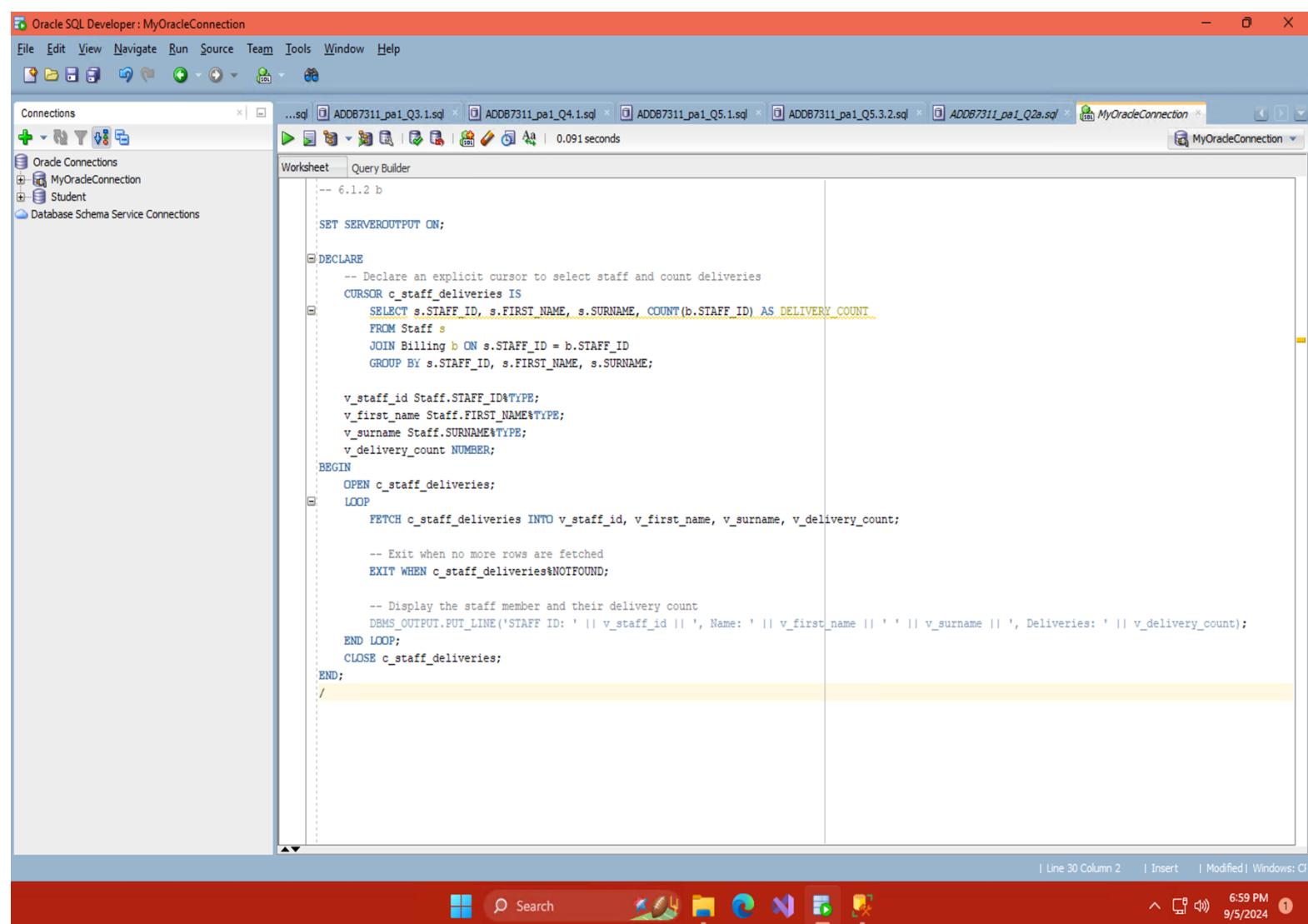
    -- Display the result
    DBMS_OUTPUT.PUT_LINE('Rows Updated: ' || v_rows_updated);
END;
/
```

- Script Output:** The output shows the result of the execution:

```
Rows Updated: 2
PL/SQL procedure successfully completed.
```

- Bottom Bar:** Shows system icons and the status bar indicating "Task completed in 0.091 seconds".
- Status Bar:** Line 20 Column 1, Insert, Modified, Windows: C:\
- System Taskbar:** Shows the date and time as 6:58 PM 9/5/2024.

Example Of Explicit Cursor Attributes



The screenshot shows the Oracle SQL Developer interface with a PL/SQL script in the Worksheet tab. The script demonstrates the use of explicit cursor attributes to select staff members and their delivery counts.

```
-- 6.1.2 b

SET SERVEROUTPUT ON;

DECLARE
    -- Declare an explicit cursor to select staff and count deliveries
    CURSOR c_staff_deliveries IS
        SELECT s.STAFF_ID, s.FIRST_NAME, s.SURNAME, COUNT(b.STAFF_ID) AS DELIVERY_COUNT
        FROM Staff s
        JOIN Billing b ON s.STAFF_ID = b.STAFF_ID
        GROUP BY s.STAFF_ID, s.FIRST_NAME, s.SURNAME;

    v_staff_id Staff.STAFF_ID%TYPE;
    v_first_name Staff.FIRST_NAME%TYPE;
    v_surname Staff.SURNAME%TYPE;
    v_delivery_count NUMBER;

BEGIN
    OPEN c_staff_deliveries;
    LOOP
        FETCH c_staff_deliveries INTO v_staff_id, v_first_name, v_surname, v_delivery_count;
        -- Exit when no more rows are fetched
        EXIT WHEN c_staff_deliveries%NOTFOUND;

        -- Display the staff member and their delivery count
        DBMS_OUTPUT.PUT_LINE('STAFF ID: ' || v_staff_id || ', Name: ' || v_first_name || ' ' || v_surname || ', Deliveries: ' || v_delivery_count);
    END LOOP;
    CLOSE c_staff_deliveries;
END;
/
```

Oracle SQL Developer : MyOracleConnection

File Edit View Navigate Run Source Team Tools Window Help

Connections ...sql ADD87311_pa1_Q3.1.sql ADD87311_pa1_Q4.1.sql ADD87311_pa1_Q5.1.sql ADD87311_pa1_Q5.3.2.sql ADD87311_pa1_Q2a.sql MyOracleConnection MyOracleConnection

Worksheet Query Builder

```
FROM Staff s
JOIN Billing b ON s.STAFF_ID = b.STAFF_ID
GROUP BY s.STAFF_ID, s.FIRST_NAME, s.SURNAME;

v_staff_id Staff.STAFF_ID%TYPE;
v_first_name Staff.FIRST_NAME%TYPE;
v_surname Staff.SURNAME%TYPE;
v_delivery_count NUMBER;

BEGIN
    OPEN c_staff_deliveries;
    LOOP
        FETCH c_staff_deliveries INTO v_staff_id, v_first_name, v_surname, v_delivery_count;
        -- Exit when no more rows are fetched
        EXIT WHEN c_staff_deliveries%NOTFOUND;
        -- Display the staff member and their delivery count
        DBMS_OUTPUT.PUT_LINE('STAFF ID: ' || v_staff_id || ', Name: ' || v_first_name || ' ' || v_surname || ', Deliveries: ' || v_delivery_count);
    END LOOP;
    CLOSE c_staff_deliveries;
END;
/
```

Script Output X | Task completed in 0.09 seconds

```
STAFF ID: 51013, Name: Harry Sheen, Deliveries: 3
STAFF ID: 51015, Name: Roberto Henry, Deliveries: 5
STAFF ID: 51012, Name: Marl Wright, Deliveries: 4
STAFF ID: 51014, Name: Jabu Kolani, Deliveries: 3
STAFF ID: 51018, Name: Maxwell Dube, Deliveries: 1
STAFF ID: 51016, Name: Pat Durant, Deliveries: 1
STAFF ID: 51111, Name: Bob Truth, Deliveries: 1
STAFF ID: 51019, Name: Shane Mane, Deliveries: 2

PL/SQL procedure successfully completed.
```

| Line 30 Column 2 | Insert | Modified | Windows: C:\

Search       

6:59 PM 9/5/2024

Q6.2 Implementing The Usage Of A Sequence

The screenshot shows the Oracle SQL Developer interface. The left pane displays a tree view of database objects under 'Connections'. The central workspace contains a 'Worksheet' tab with the following SQL code:

```
--Q6.2
-- Create a sequence to generate unique BILL_ID values
CREATE SEQUENCE billing_seq
START WITH 821      -- Start sequence at 821
INCREMENT BY 1       -- Increment by 1 each time
NOCACHE;            -- Do not cache sequence values

-- Insert a new record into the Billing table
INSERT INTO Billing (BILL_ID, CUSTOMER_ID, STAFF_ID, BILL_DATE)
VALUES (billing_seq.NEXTVAL, 11117, 51015, SYSDATE);

-- Use sequence for BILL_ID and current date for BILL_DATE

-- Verify the newly inserted record
SELECT * FROM Billing
WHERE BILL_ID = (SELECT MAX(BILL_ID) FROM Billing);

-- Fetch the record with the highest BILL_ID (newly inserted)
```

The 'Script Output' tab shows the execution results:

BILL_ID	CUSTOMER_ID	STAFF_ID	BILL_DATE
1	823	11117	51015 05-SEP-24

The status bar at the bottom indicates: Line 2 Column 1 | Insert | Modified | Unix/Mac: LF | 5:15 PM | 9/5/2024.

Bibliography

- Imperva, 2024. *Separation of Duties*. [Online]
Available at: <https://www.imperva.com/learn/data-security/separation-of-duties/#:~:text=These%20separation%20of%20duties%20controls,data%20residing%20within%20production%20systems>.
- Guldenpfennig, M., 2023. *Study.com*. [Online]
Available at: <https://study.com/learn/lesson/what-is-flat-file-database.html#:~:text=A%20flat%20file%20database%20is%20basically%20a%20giant%20collection%20of,sales%20to%20orders%20to%20invoicedes>
- IBM, 2022. *What is a relational database?*. [Online]
Available at: <https://www.ibm.com/topics/relational-databases#:~:text=A%20relational%20database%20is%20a,key%20or%20a%20foreign%20key>
- Hughes, A., Wright, G. & Castagna, R., 2024. *flat file*. [Online]
Available at:
<https://www.techtarget.com/searchdatamanagement/definition/flat-file>
- KHAN, M. I., 2023. *PL/SQL Introduction*. [Online]
Available at: <https://www.geeksforgeeks.org/plsql-introduction/>
- Navlani, A., 2019. *Views (Virtual Tables) in SQL*. [Online]
Available at: https://www.datacamp.com/tutorial/views-in-sql?utm_source=google&utm_medium=paid_search&utm_campaignid=19589720824&utm_adgroupid=157156376591&utm_device=c&utm_keyword=&utm_matchtype=&utm_network=g&utm_adpostion=&utm_creative=711801244703&utm_targetid=dsa
- simranjenny84, 2022. *Difference between Implicit and Explicit Cursors*. [Online]
Available at: <https://www.geeksforgeeks.org/difference-between-implicit-and-explicit-cursors/>
- Javapoint.com, 2021. *PL/SQL Cursor*. [Online]
Available at: <https://www.javatpoint.com/pl-sql-cursor>