2025

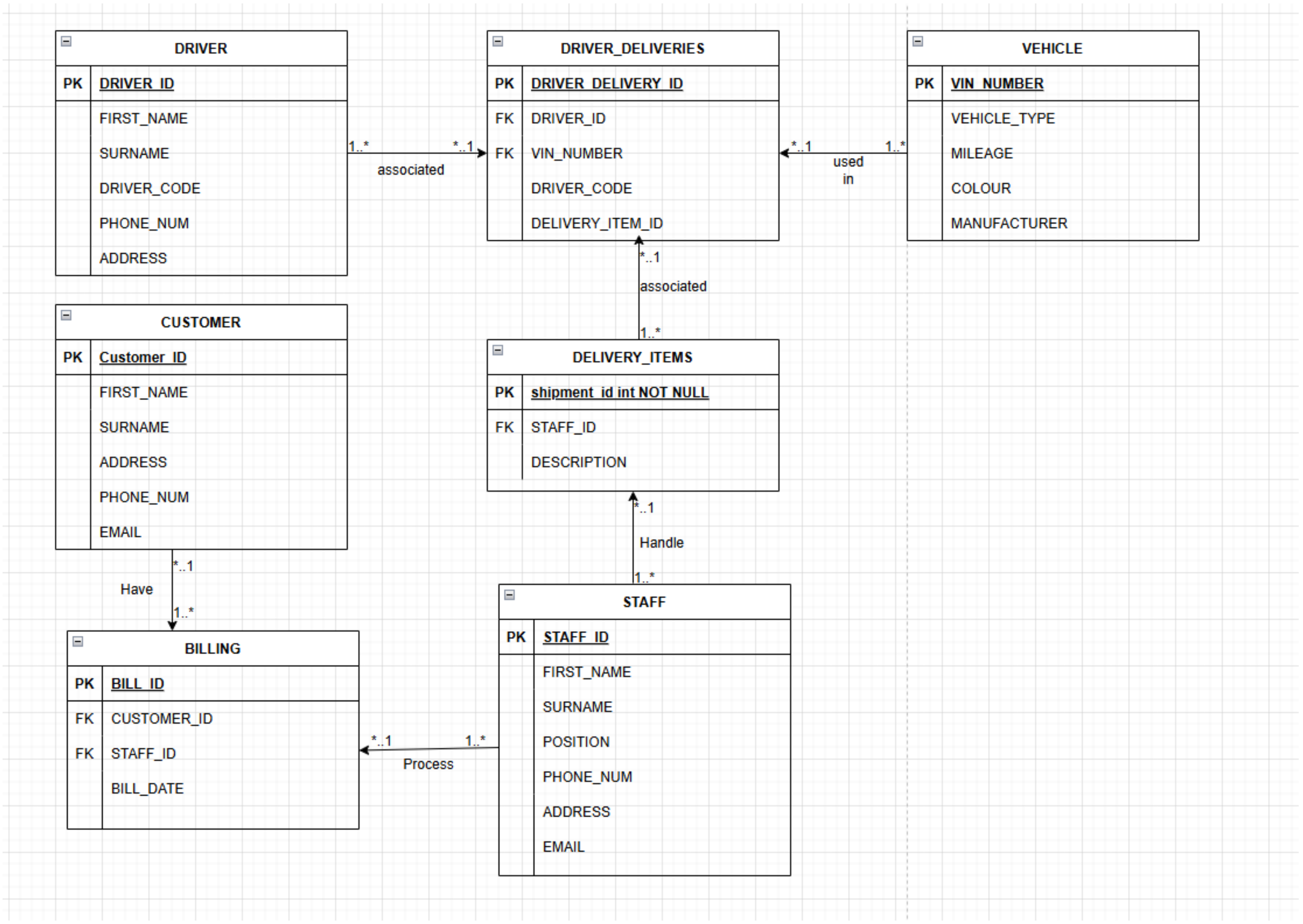
INSY7213

ASSIGNMENT 1

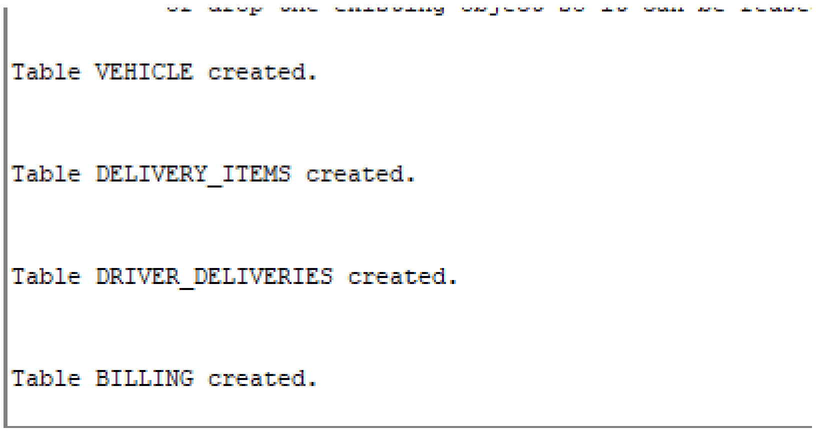
NEHAAR GOSAI

ST10359529

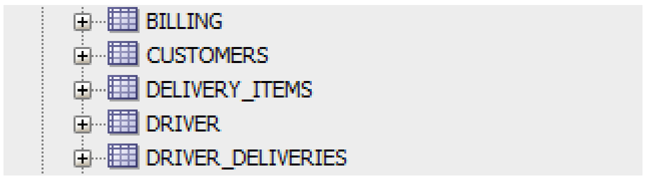
Question 1


DRIVER
PK DRIVER_ID
FIRST_NAME
SURNAME
DRIVER_CODE
PHONE_NUM
ADDRESS
DRIVER_DELIVERIES
PK DRIVER_DELIVERY_ID
FK DRIVER_ID
FK VIN_NUMBER
DRIVER_CODE
DELIVERY_ITEM_ID
VEHICLE
PK VIN_NUMBER
VEHICLE_TYPE
MILEAGE
COLOUR
MANUFACTURER
CUSTOMER
PK Customer_ID
FIRST_NAME
SURNAME
ADDRESS
PHONE_NUM
EMAIL
DELIVERY_ITEMS
PK shipment_id int NOT NULL
FK STAFF_ID
DESCRIPTION
STAFF
PK STAFF_ID
FIRST_NAME
SURNAME
POSITION
PHONE_NUM
ADDRESS
EMAIL
BILLING
PK BILL_ID
FK CUSTOMER_ID
FK STAFF_ID
BILL_DATE
1..*
*..1
associated
*..1
used in
1..*
*..1
associated
1..*
*..1
Handle
1..*
*..1
Have
1..*
*..1
1..*
Process

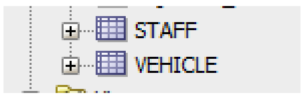
Question 2



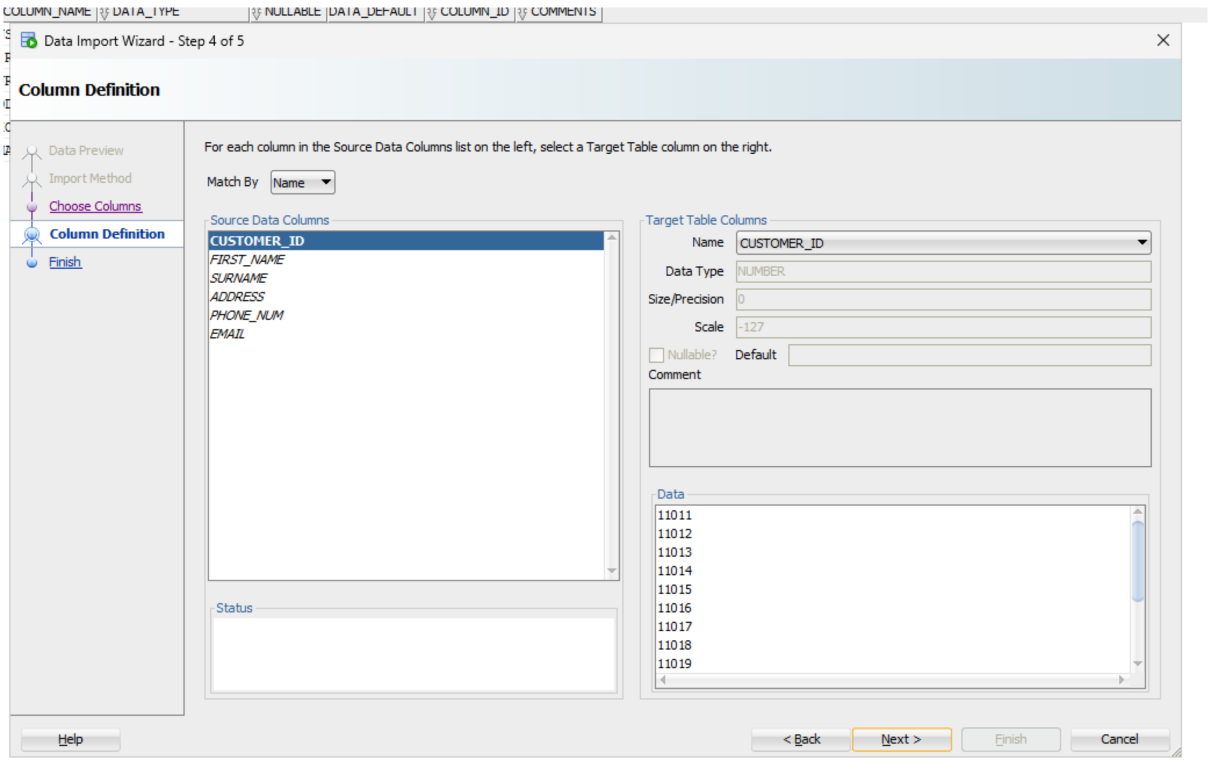
```
Table VEHICLE created.


Table DELIVERY_ITEMS created.


Table DRIVER_DELIVERIES created.


Table BILLING created.
```



- BILLING
- CUSTOMERS
- DELIVERY_ITEMS
- DRIVER
- DRIVER_DELIVERIES



- STAFF
- VEHICLE



COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS

Data Import Wizard - Step 4 of 5

Column Definition

For each column in the Source Data Columns list on the left, select a Target Table column on the right.

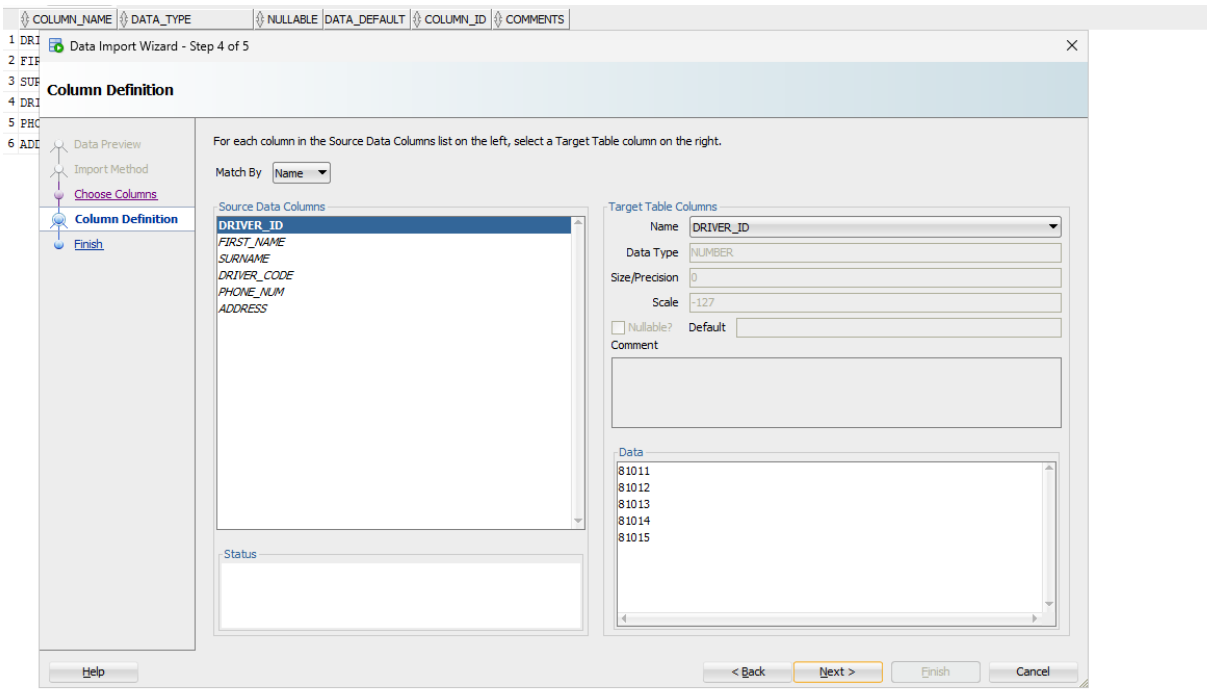
Match By: Name

Source Data Columns:
CUSTOMER_ID
FIRST_NAME
SURNAME
ADDRESS
PHONE_NUM
EMAIL

Target Table Columns:
Name: CUSTOMER_ID
Data Type: NUMBER
Size/Precision: 0
Scale: -127
Nullable? [ ]
Default:
Comment:

Data:
11011
11012
11013
11014
11015
11016
11017
11018
11019

Status


Data Import Wizard - Step 4 of 5

Column Definition

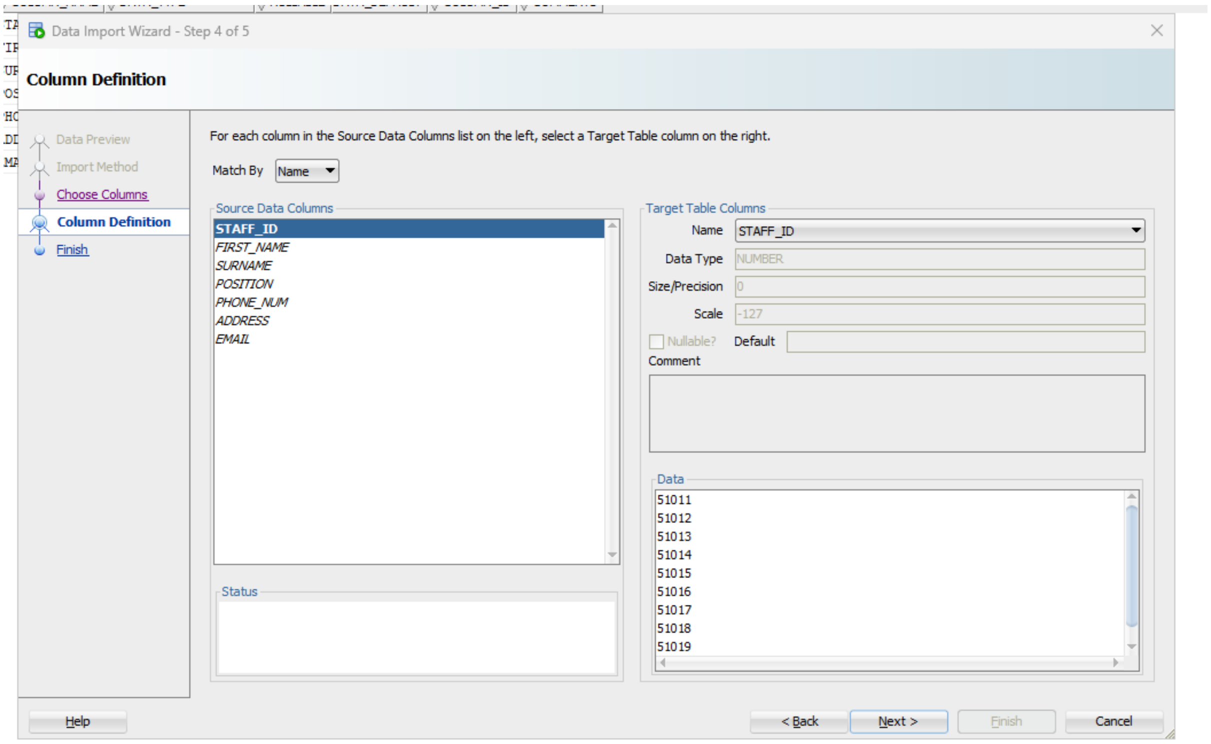
For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By: Name

Source Data Columns: DRIVER_ID, FIRST_NAME, SURNAME, DRIVER_CODE, PHONE_NUM, ADDRESS

Target Table Columns
Name: DRIVER_ID
Data Type: NUMBER
Size/Precision: 0
Scale: -127
Nullable? Default
Comment

Data
```
81011
81012
81013
81014
81015
```


Data Import Wizard - Step 4 of 5

Column Definition

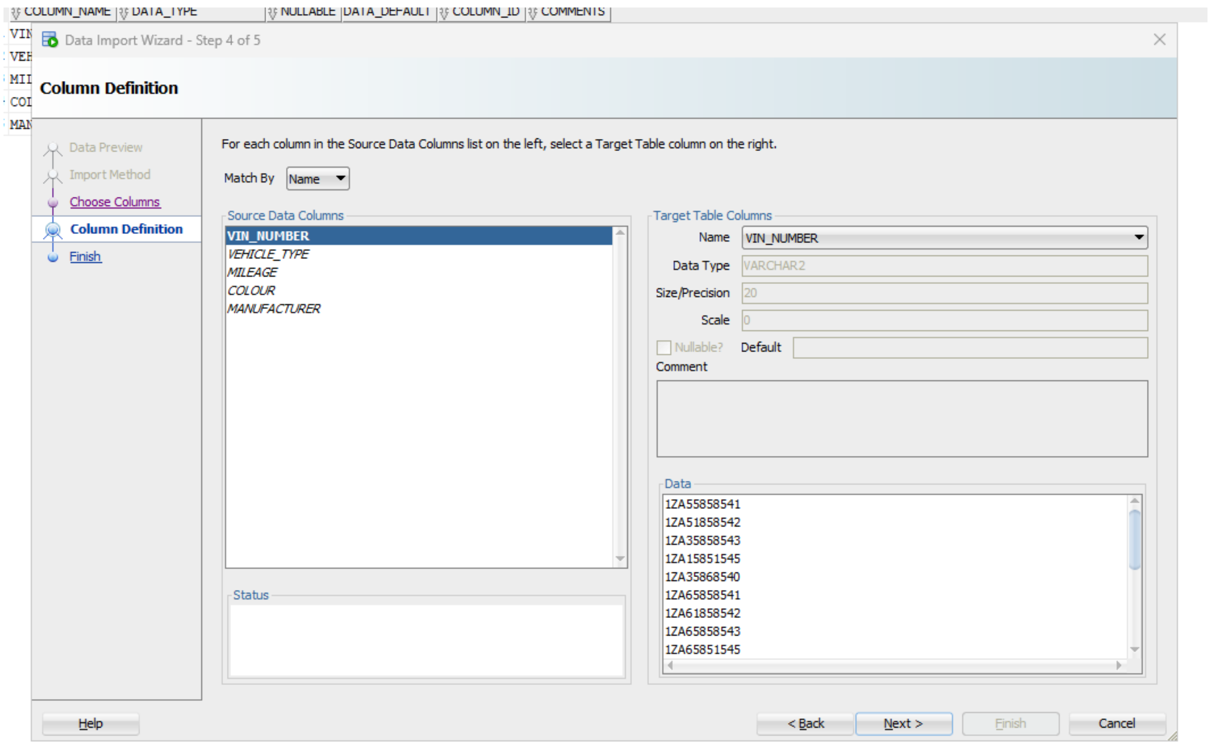
For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By: Name

Source Data Columns: STAFF_ID, FIRST_NAME, SURNAME, POSITION, PHONE_NUM, ADDRESS, EMAIL

Target Table Columns
Name: STAFF_ID
Data Type: NUMBER
Size/Precision: 0
Scale: -127
Nullable? Default
Comment

Data
```
51011
51012
51013
51014
51015
51016
51017
51018
51019
```



COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS

Data Import Wizard - Step 4 of 5

Column Definition

For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By: Name

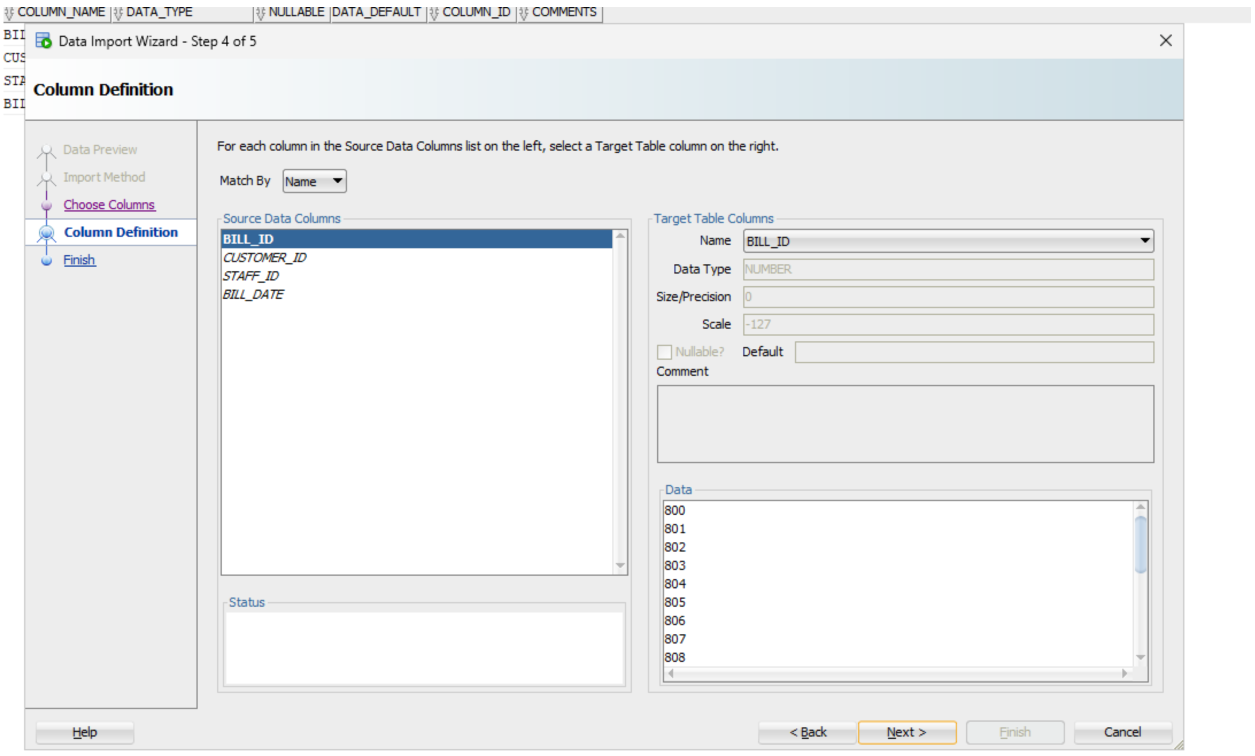
Source Data Columns:
- VIN_NUMBER
- *VEHICLE_TYPE*
- *MILEAGE*
- *COLOUR*
- *MANUFACTURER*

Target Table Columns

Name	VIN_NUMBER
Data Type	VARCHAR2
Size/Precision	20
Scale	0
Nullable?	Default
Comment	

Data:
```
1ZA55858541
1ZA51858542
1ZA35858543
1ZA15851545
1ZA35868540
1ZA65858541
1ZA61858542
1ZA65858543
1ZA65851545
```

Status



COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS

Data Import Wizard - Step 4 of 5

Column Definition

For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By: Name

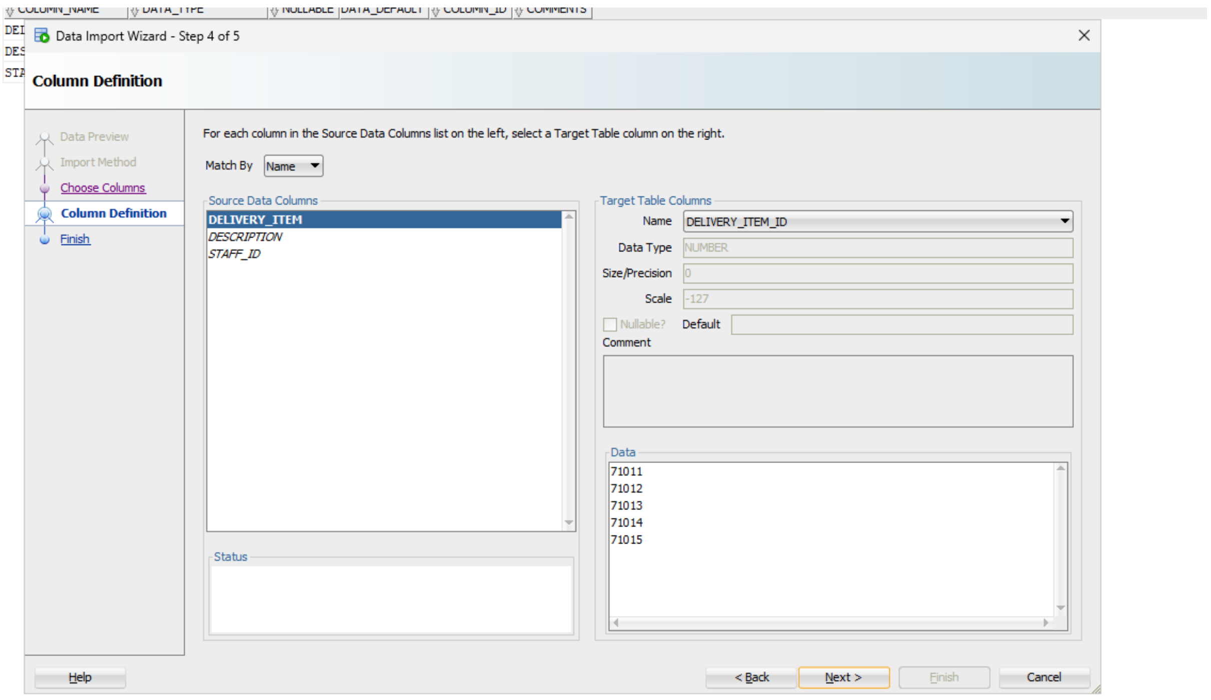
Source Data Columns:
- BILL_ID
- *CUSTOMER_ID*
- *STAFF_ID*
- *BILL_DATE*

Target Table Columns

Name	BILL_ID
Data Type	NUMBER
Size/Precision	0
Scale	-127
Nullable?	Default
Comment	

Data:
```
800
801
802
803
804
805
806
807
808
```

Status


Data Import Wizard - Step 4 of 5

Column Definition

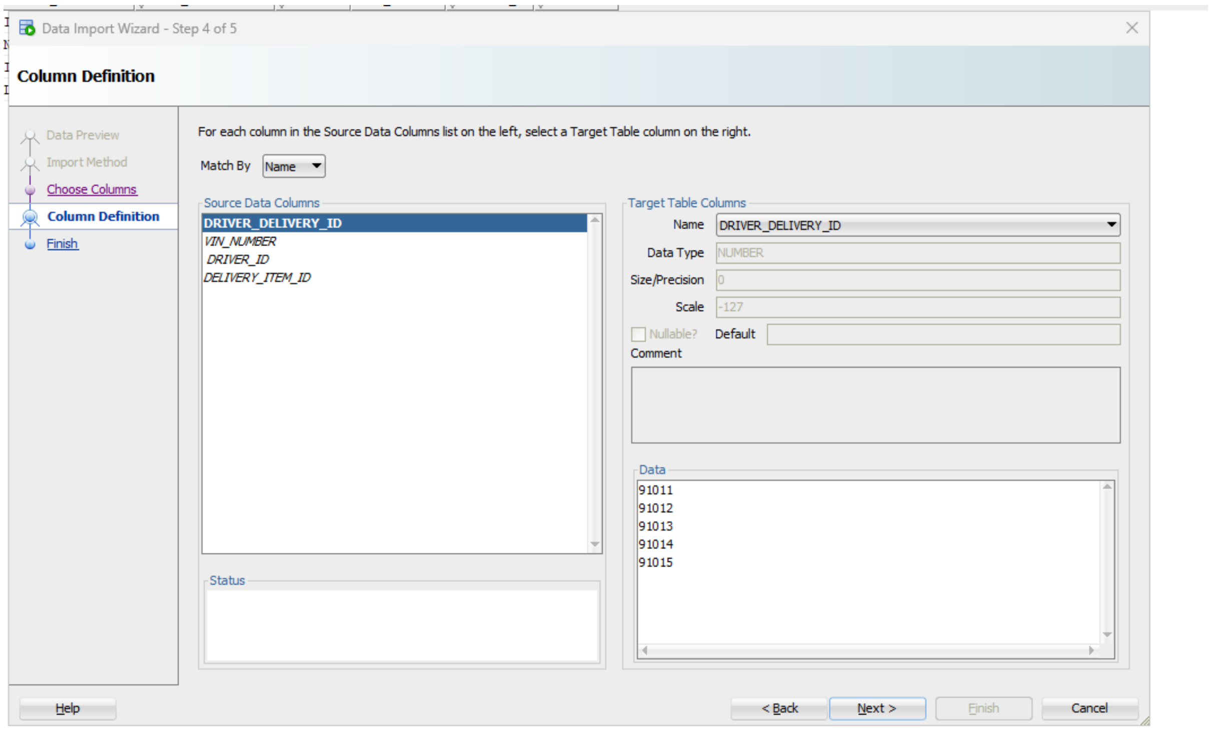
For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By: Name

Source Data Columns:
DELIVERY_ITEM
DESCRIPTION
STAFF_ID

Target Table Columns
Name: DELIVERY_ITEM_ID
Data Type: NUMBER
Size/Precision: 0
Scale: -127
Nullable? Default
Comment

Data
71011
71012
71013
71014
71015


Data Import Wizard - Step 4 of 5

Column Definition

For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By: Name

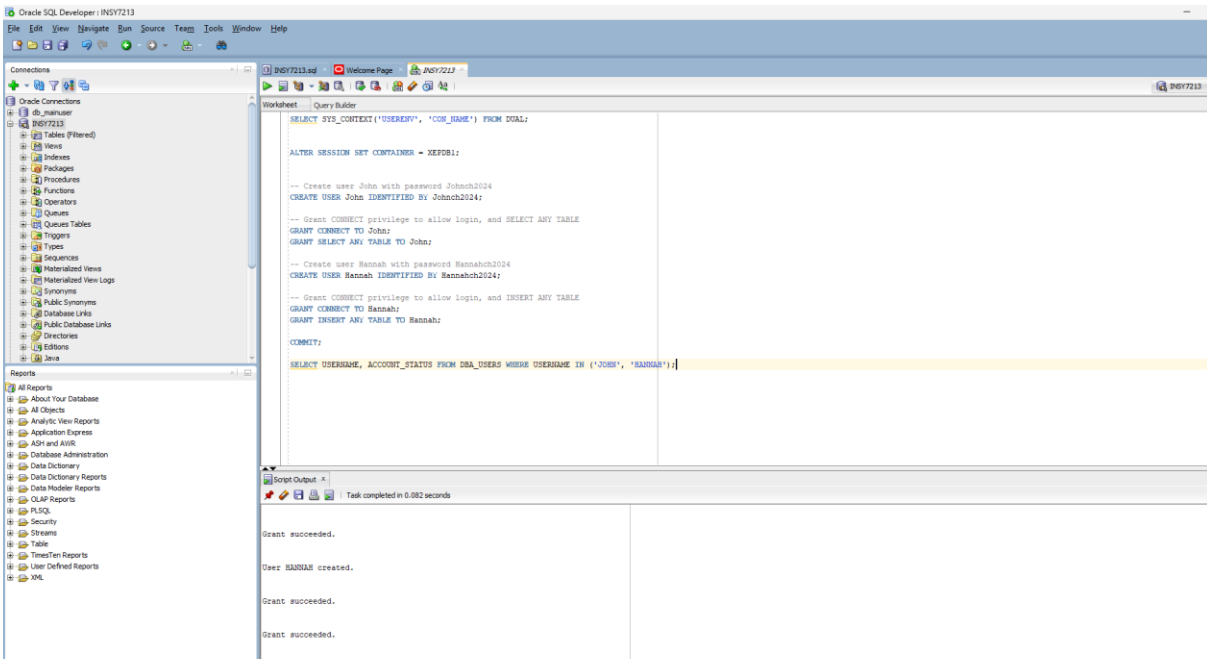
Source Data Columns:
DRIVER_DELIVERY_ID
VIN_NUMBER
DRIVER_ID
DELIVERY_ITEM_ID

Target Table Columns
Name: DRIVER_DELIVERY_ID
Data Type: NUMBER
Size/Precision: 0
Scale: -127
Nullable? Default
Comment

Data
91011
91012
91013
91014
91015

Question 3

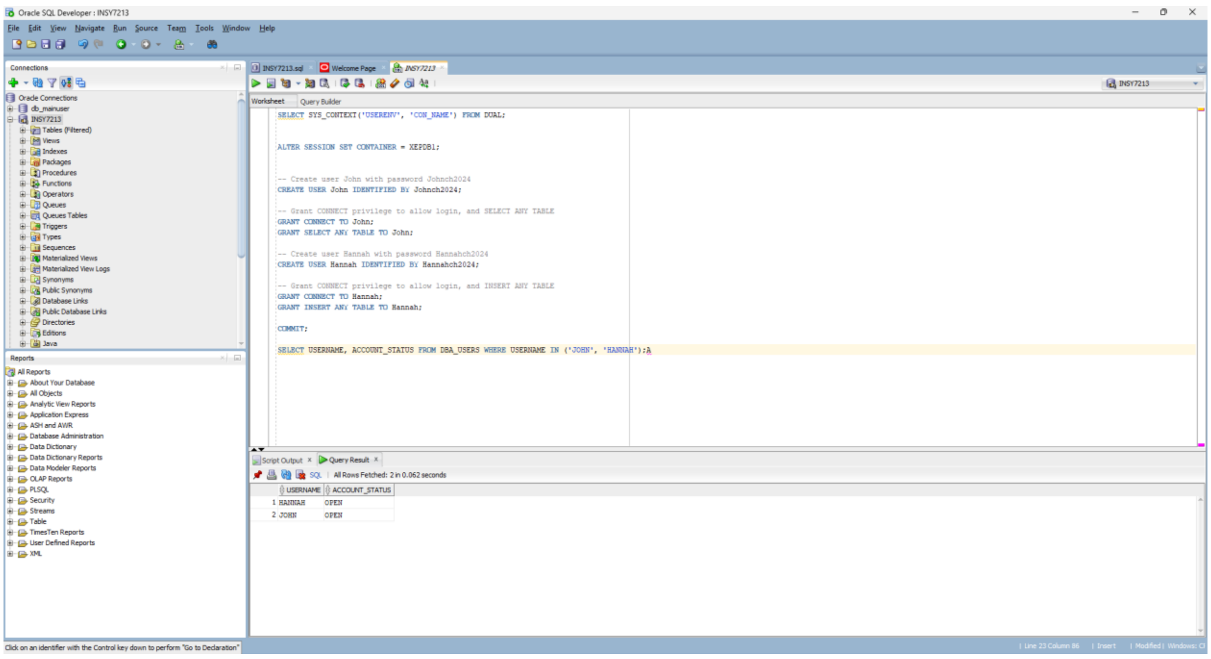
3.1



```sql
SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM DUAL;


ALTER SESSION SET CONTAINER = XEPDB1;


-- Create user John with password Johnch2024
CREATE USER John IDENTIFIED BY Johnch2024;

-- Grant CONNECT privilege to allow login, and SELECT ANY TABLE
GRANT CONNECT TO John;
GRANT SELECT ANY TABLE TO John;

-- Create user Hannah with password Hannahch2024
CREATE USER Hannah IDENTIFIED BY Hannahch2024;

-- Grant CONNECT privilege to allow login, and INSERT ANY TABLE
GRANT CONNECT TO Hannah;
GRANT INSERT ANY TABLE TO Hannah;

COMMIT;

SELECT USERNAME, ACCOUNT_STATUS FROM DBA_USERS WHERE USERNAME IN ('JOHN', 'HANNAH');
```

Task completed in 0.082 seconds

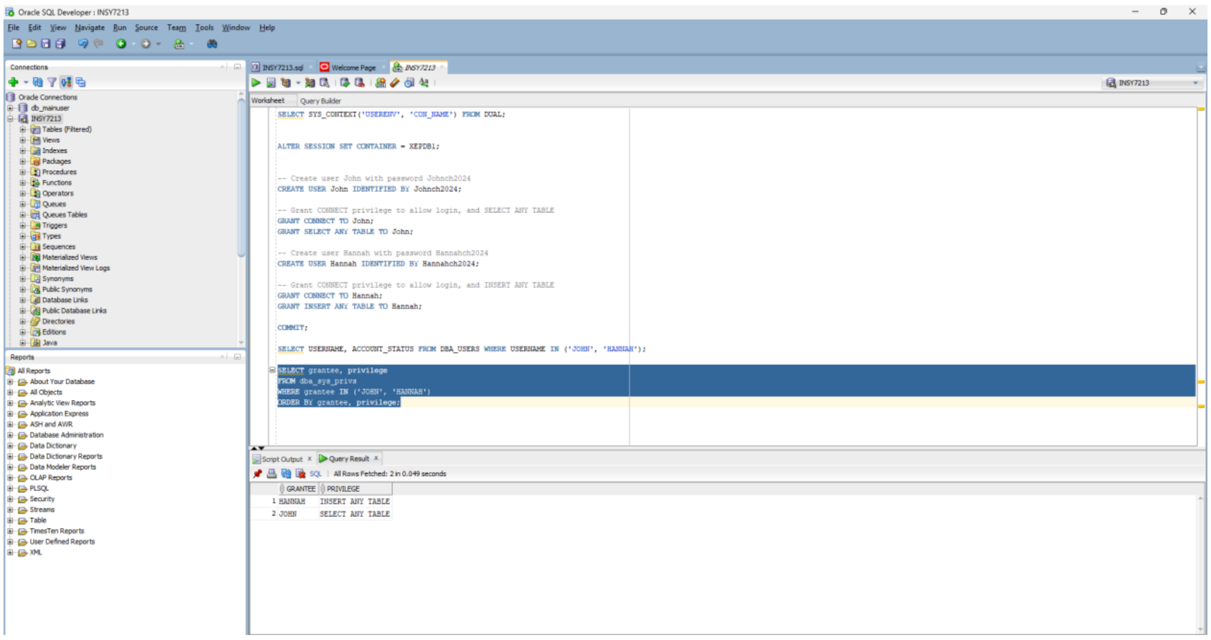
```
Grant succeeded.


User HANNAH created.


Grant succeeded.


Grant succeeded.
```



```sql
SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM DUAL;


ALTER SESSION SET CONTAINER = XEPDB1;


-- Create user John with password Johnch2024
CREATE USER John IDENTIFIED BY Johnch2024;

-- Grant CONNECT privilege to allow login, and SELECT ANY TABLE
GRANT CONNECT TO John;
GRANT SELECT ANY TABLE TO John;

-- Create user Hannah with password Hannahch2024
CREATE USER Hannah IDENTIFIED BY Hannahch2024;

-- Grant CONNECT privilege to allow login, and INSERT ANY TABLE
GRANT CONNECT TO Hannah;
GRANT INSERT ANY TABLE TO Hannah;

COMMIT;

SELECT USERNAME, ACCOUNT_STATUS FROM DBA_USERS WHERE USERNAME IN ('JOHN', 'HANNAH');
```

All Rows Fetched: 2 in 0.062 seconds

	USERNAME	ACCOUNT_STATUS
1	HANNAH	OPEN
2	JOHN	OPEN


Oracle SQL Developer : INSY7213

```sql
SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM DUAL;


ALTER SESSION SET CONTAINER = XEPDB1;


-- Create user John with password Johnch2024
CREATE USER John IDENTIFIED BY Johnch2024;

-- Grant CONNECT privilege to allow login, and SELECT ANY TABLE
GRANT CONNECT TO John;
GRANT SELECT ANY TABLE TO John;

-- Create user Hannah with password Hannahch2024
CREATE USER Hannah IDENTIFIED BY Hannahch2024;

-- Grant CONNECT privilege to allow login, and INSERT ANY TABLE
GRANT CONNECT TO Hannah;
GRANT INSERT ANY TABLE TO Hannah;

COMMIT;

SELECT USERNAME, ACCOUNT_STATUS FROM DBA_USERS WHERE USERNAME IN ('JOHN', 'HANNAH');

SELECT grantee, privilege
FROM dba_sys_privs
WHERE grantee IN ('JOHN', 'HANNAH')
ORDER BY grantee, privilege;
```

All Rows Fetched: 2 in 0.049 seconds

	GRANTEE	PRIVILEGE
1	HANNAH	INSERT ANY TABLE
2	JOHN	SELECT ANY TABLE

3.2.

A fundamental security principle in database administration is separation of duties (SOD), which assigns tasks to users in order to reduce risks such as fraud, mistakes, or illegal access. For Hannah (INSTALL ANY TABLE) and John (SELECT ANY TABLE):

- Prevents Single-User Control: Hannah can add new data but not view current entries, while John can read and view data but not edit it. This lowers the possibility of data manipulation or leakage by guaranteeing that no single user may access and change data unilaterally.

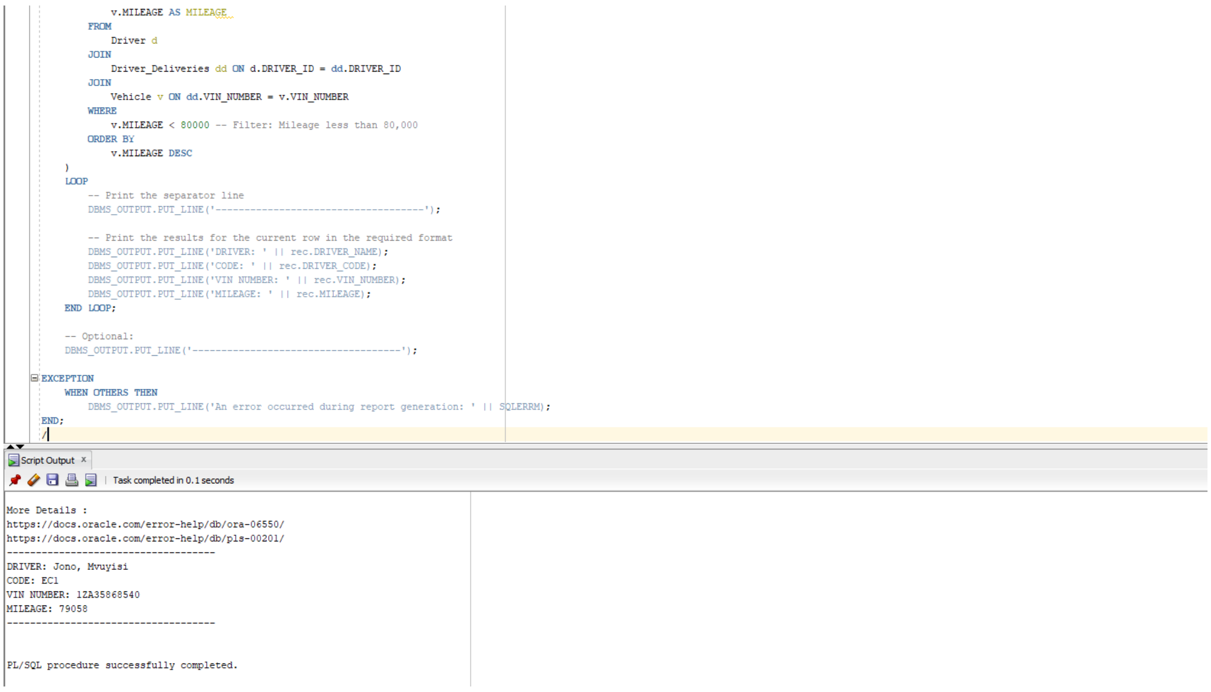
- Enhances Accountability: Actions can be linked to certain users, which facilitates auditing and discourages abuse.

- Complies with Security Best Practices: By restricting rights, SOD adheres to the least privilege principle and safeguards sensitive data (such as customer information) in companies like Cheath Deliveries.

- Mitigates Risks: Integrity and compliance are promoted by preventing situations in which a single user could add fictitious entries and later confirm them.

Question 4

4.1.



```
            v.MILEAGE AS 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 -- Filter: Mileage less than 80,000
        ORDER BY
            v.MILEAGE DESC
    )
    LOOP
        -- Print the separator line
        DBMS_OUTPUT.PUT_LINE('------------------------------------');

        -- Print the results for the current row in the required format
        DBMS_OUTPUT.PUT_LINE('DRIVER: ' || rec.DRIVER_NAME);
        DBMS_OUTPUT.PUT_LINE('CODE: ' || rec.DRIVER_CODE);
        DBMS_OUTPUT.PUT_LINE('VIN NUMBER: ' || rec.VIN_NUMBER);
        DBMS_OUTPUT.PUT_LINE('MILEAGE: ' || rec.MILEAGE);
    END LOOP;

    -- Optional:
    DBMS_OUTPUT.PUT_LINE('------------------------------------');

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred during report generation: ' || SQLERRM);
END;
/
```

Script Output

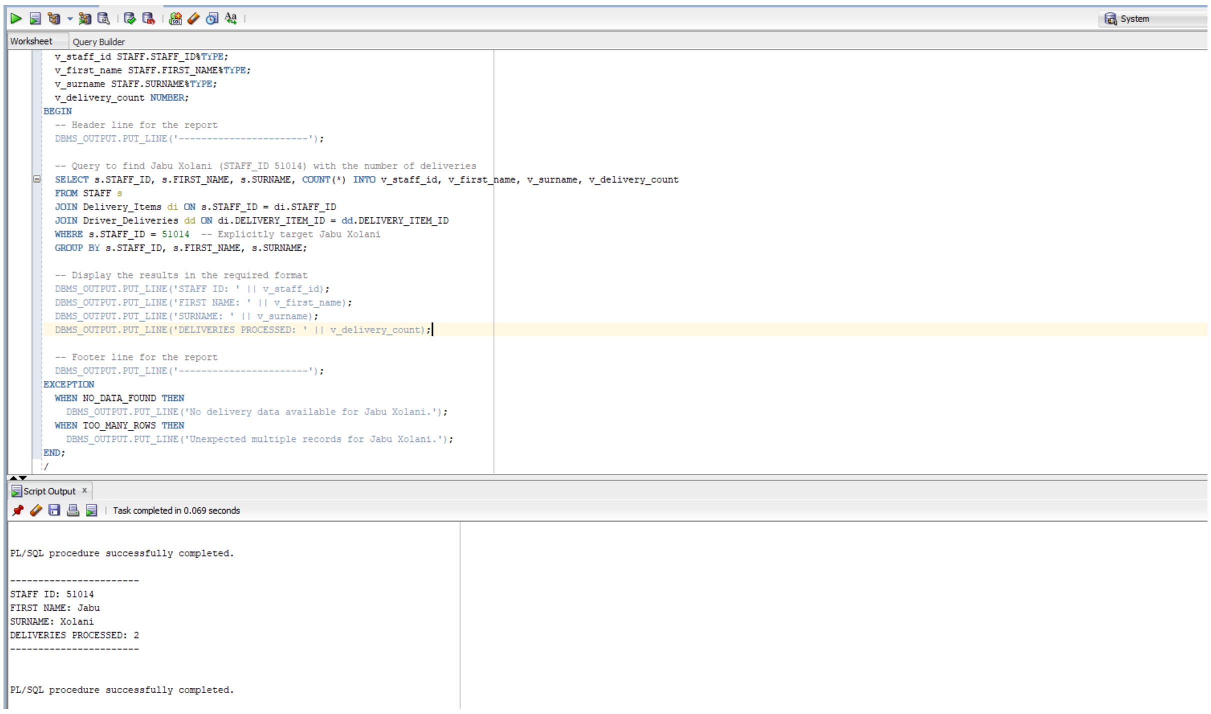
Task completed in 0.1 seconds

```
More Details :
https://docs.oracle.com/error-help/db/ora-06550/
https://docs.oracle.com/error-help/db/pls-00201/
-----------------------------------
DRIVER: Jono, Mvuyisi
CODE: EC1
VIN NUMBER: 1ZA35868540
MILEAGE: 79058
-----------------------------------


PL/SQL procedure successfully completed.
```

4.2. The SQL query provided for Question 4.1 joins the Driver, Driver_Deliveries, and Vehicle tables to generate a report listing drivers, their license codes, vehicle identification numbers, and mileage for vehicles with less than 80,000 miles. It uses the JOIN clause to link Driver and Driver_Deliveries on DRIVER_ID and Driver_Deliveries with Vehicle on VIN_NUMBER, ensuring only relevant records are included. The WHERE v.MILEAGE < 80000 condition filters for low-mileage vehicles, while the ORDER BY v.MILEAGE clause sorts the results in ascending order for clarity. For CHEETAH DELIVERIES, this query is highly relevant as it helps identify underutilized vehicles, such as the one with 79,058 miles driven by Jono Mvuyisi, enabling targeted maintenance or reallocation to optimize fleet efficiency. Additionally, it supports cost management by highlighting potential overstocked assets, ensures driver assignments align with vehicle usage, and aids in compliance by verifying driver codes, ultimately enhancing operational planning and customer service delivery.

Question 5

5.1.



```
  v_staff_id STAFF.STAFF_ID%TYPE;
  v_first_name STAFF.FIRST_NAME%TYPE;
  v_surname STAFF.SURNAME%TYPE;
  v_delivery_count NUMBER;
BEGIN
  -- Header line for the report
  DBMS_OUTPUT.PUT_LINE('----------------------');

  -- Query to find Jabu Xolani (STAFF_ID 51014) with the number of deliveries
  SELECT s.STAFF_ID, s.FIRST_NAME, s.SURNAME, COUNT(*) INTO v_staff_id, v_first_name, v_surname, v_delivery_count
  FROM STAFF s
  JOIN Delivery_Items di ON s.STAFF_ID = di.STAFF_ID
  JOIN Driver_Deliveries dd ON di.DELIVERY_ITEM_ID = dd.DELIVERY_ITEM_ID
  WHERE s.STAFF_ID = 51014  -- Explicitly target Jabu Xolani
  GROUP BY s.STAFF_ID, s.FIRST_NAME, s.SURNAME;

  -- Display the results in the required format
  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);

  -- Footer line for the report
  DBMS_OUTPUT.PUT_LINE('----------------------');
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No delivery data available for Jabu Xolani.');
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE('Unexpected multiple records for Jabu Xolani.');
END;
/
```

Task completed in 0.069 seconds

```
PL/SQL procedure successfully completed.

----------------------
STAFF ID: 51014
FIRST NAME: Jabu
SURNAME: Xolani
DELIVERIES PROCESSED: 2
----------------------


PL/SQL procedure successfully completed.
```