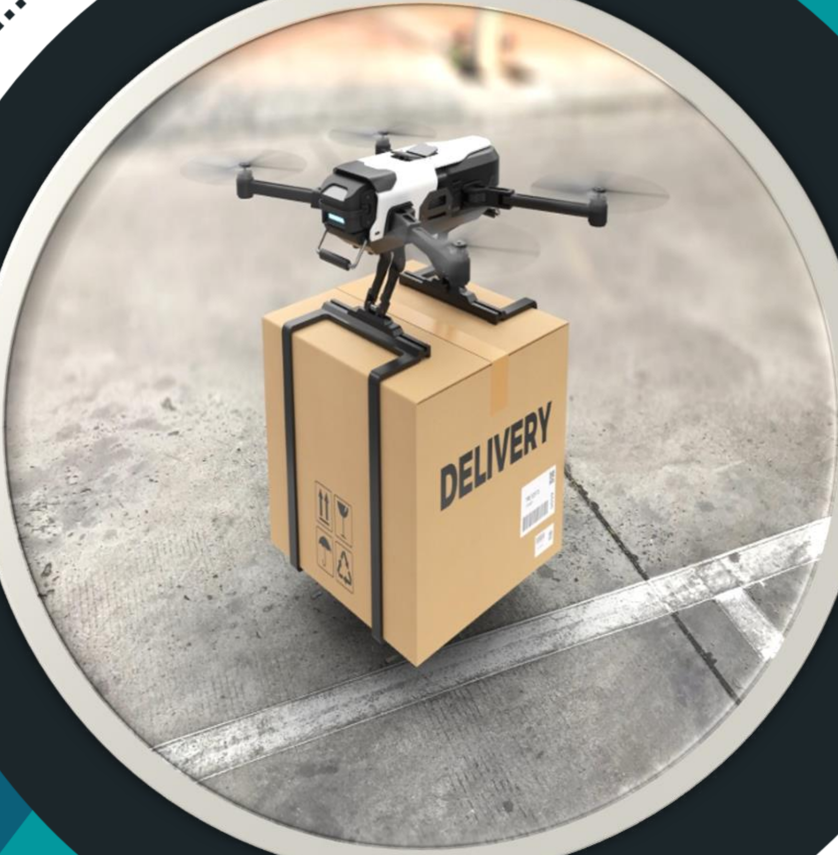
FINAL REPORT

2022



Prepared By:

2140520 قصي الثقفي

2140882 ريان الصبحي

2041533 عبدالرحيم فادر

2140469 رياض الشهري

PROJECT PLAN

IN INTRODUCTION TO DATABASE

PROJECT PROPOSAL

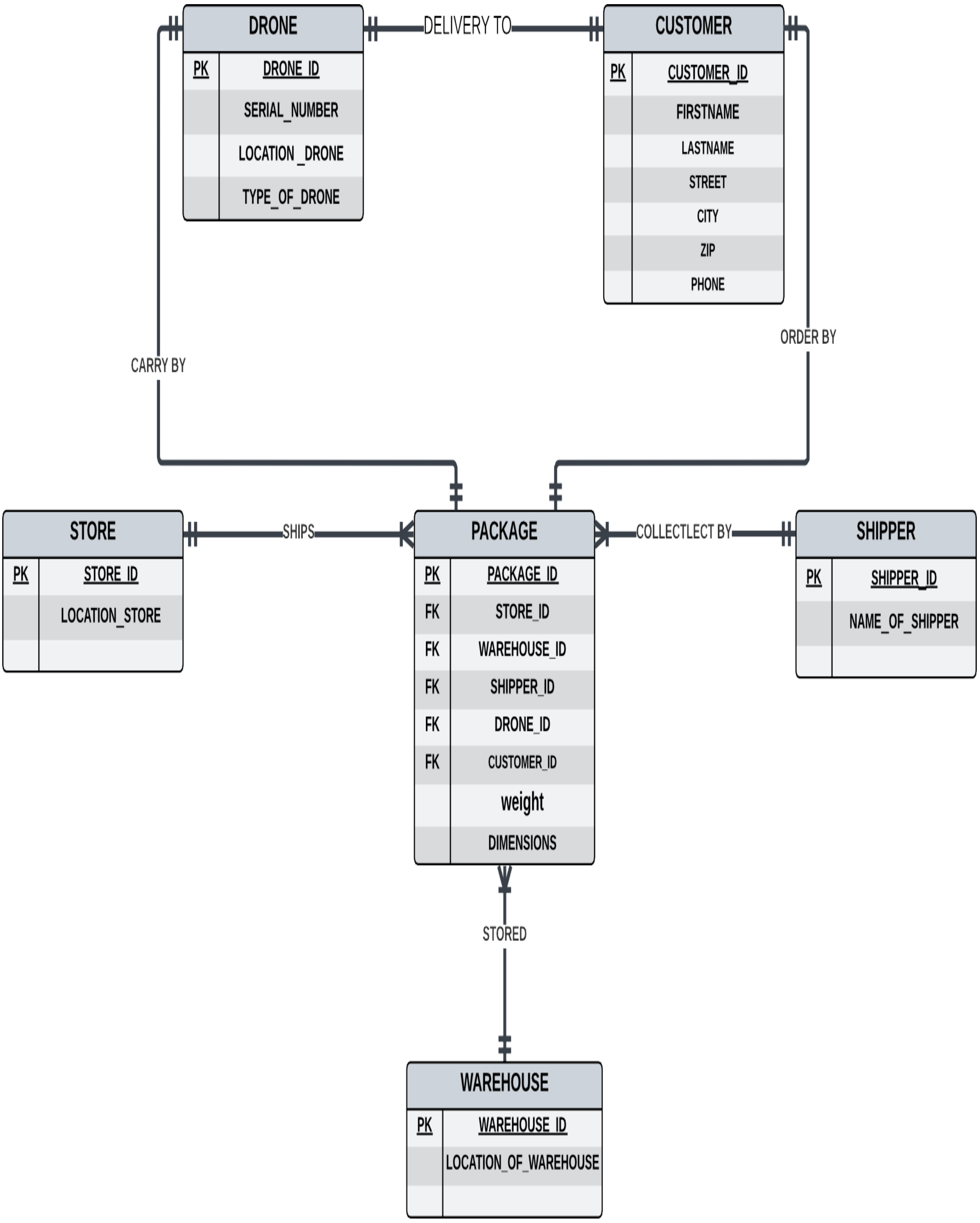
OUR PROJECT MAKES IT EASY FOR THE CUSTOMER TO ORDER ONLINE THROUGH THE DELIVERY OF ORDERS BY DRONE LIKE MEDICINES, CLOTHES OR ANY ITEM, AND WILL CONTRIBUTE TO SOLVING THE PROBLEMOF LATE OR NON ARRIVALOF ORDERS.

ريان الصبحي Leader + ER

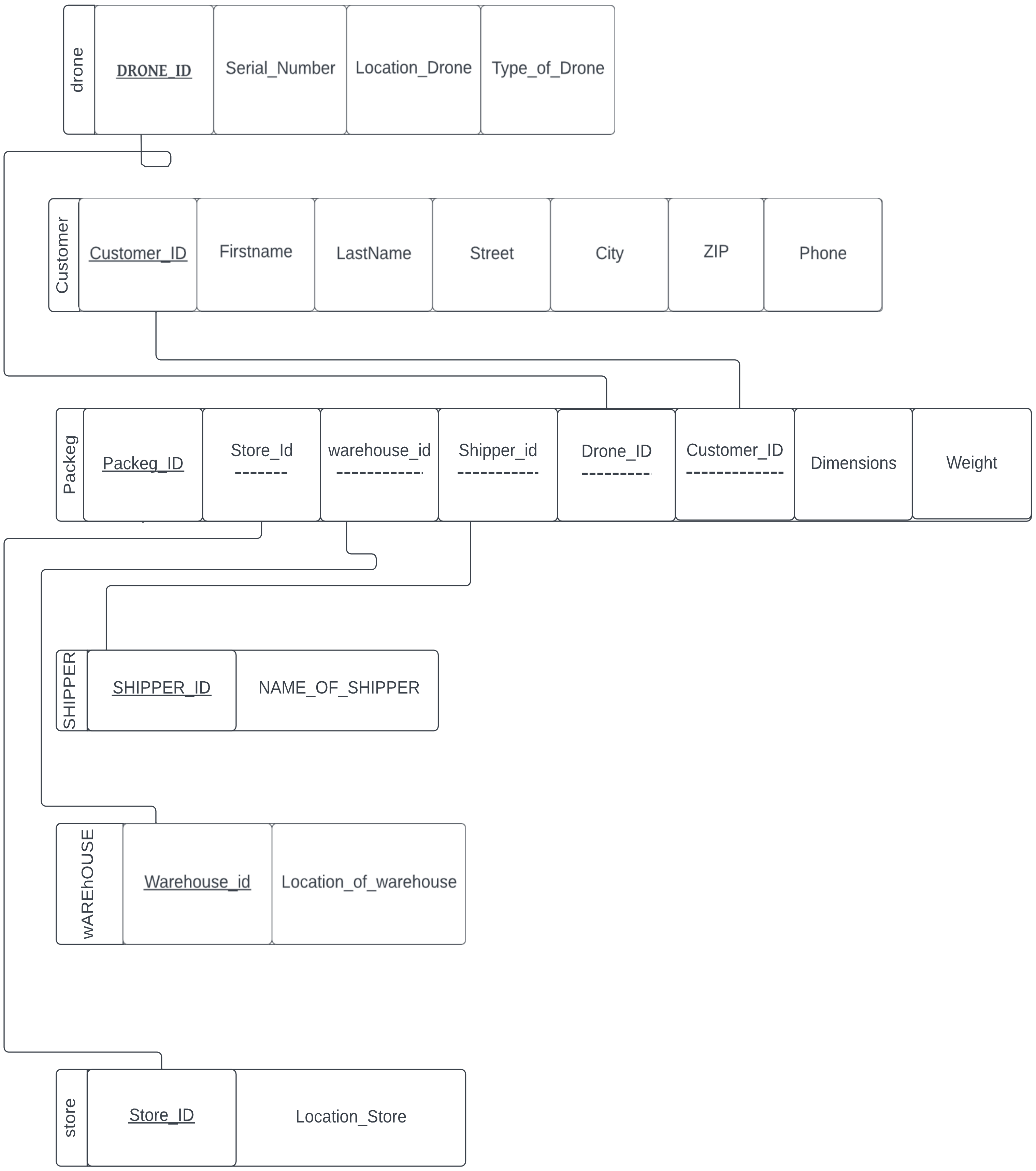
قصي الثقفي Convert ER Diagram to Relational schema + PowerPointer

عبدالرحيم فادن normalization + Create table insert

رياض الشهري Queris



DRONE
PK
DRONE_ID
SERIAL_NUMBER
LOCATION _DRONE
TYPE_OF_DRONE
DELIVERY TO
CUSTOMER
PK
CUSTOMER_ID
FIRSTNAME
LASTNAME
STREET
CITY
ZIP
PHONE
CARRY BY
ORDER BY
STORE
PK
STORE_ID
LOCATION_STORE
SHIPS
PACKAGE
PK
PACKAGE_ID
FK
STORE_ID
FK
WAREHOUSE_ID
FK
SHIPPER_ID
FK
DRONE_ID
FK
CUSTOMER_ID
weight
DIMENSIONS
COLLECTLECT BY
SHIPPER
PK
SHIPPER_ID
NAME_OF_SHIPPER
STORED
WAREHOUSE
PK
WAREHOUSE_ID
LOCATION_OF_WAREHOUSE


drone
DRONE_ID
Serial_Number
Location_Drone
Type_of_Drone
Customer
Customer_ID
Firstname
LastName
Street
City
ZIP
Phone
Packeg
Packeg_ID
Store_Id
warehouse_id
Shipper_id
Drone_ID
Customer_ID
Dimensions
Weight
SHIPPER
SHIPPER_ID
NAME_OF_SHIPPER
wAREhOUSE
Warehouse_id
Location_of_warehouse
store
Store_ID
Location_Store

The **drone** currently in NF (normalized form).

--1NF:

The **drone** form is in 1NF because there is no repeating group.

--2NF:

The **drone** is in 2NF because all non-key attributes are fully dependent on the entire key.

--3NF:

The drone relation is in 3NF because there no transitive dependency.

The **customer** currently in NF (normalized form).

--1NF:

The **customer** form is in 1NF because there is no repeating group.

--2NF:

The **customer** is in 2NF because all non-key attributes are fully dependent on the entire key.

--3NF:

The **customer** relation is in 3NF because there no transitive dependency.

The **store** currently in NF (normalized form).

--1NF:

The **store** form is in 1NF because there is no repeating group.

--2NF:

The **store** is in 2NF because all non-key attributes are fully dependent on the entire key.

--3NF:

The **store** relation is in 3NF because there no transitive dependency.

The **shipper** currently in NF (normalized form).

--1NF:

The **shipper** form is in 1NF because there is no repeating group.

--2NF:

The **shipper** is in 2NF because all non-key attributes are fully dependent on the entire key.

--3NF:

The **shipper** relation is in 3NF because there no transitive dependency.

The warehouse currently in NF (normalized form).

--1NF:

The warehouse form is in 1NF because there is no repeating group.

--2NF:

The warehouse is in 2NF because all non-key attributes are fully dependent on the entire key.

--3NF:

The warehouse relation is in 3NF because there no transitive dependency.

-- 1NF:

The **package** form is in 1NF because there is no repeating group.

--2NF:

The **package** is in 2NF because all non-key attributes are fully dependent on the entire

--3NF:

The **package** relation is in 3NF because there no transitive dependency.

```
CREATE TABLE Drones (

Drone_ID number(5) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1
INCREMENT BY 1  NOT NULL,

DRONE_SERIAL_NUMBER number(10) NOT NULL,

DRONE_LOCATION varchar2(100) NOT NULL,

DRONE_TYPE varchar2(55) NOT NULL,

PRIMARY KEY (Drone_ID),

UNIQUE (DRONE_SERIAL_NUMBER),

UNIQUE (DRONE_TYPE)

);



CREATE TABLE Customers(


CUSTOMER_ID number(10) NOT NULL,

CUSTOMER_FIRST_NAME varchar2(55) NOT NULL,

CUSTOMER_LAST_NAME varchar2(55) NOT NULL,

STREET varchar2(25) NOT NULL,

CITY varchar2(25) NOT NULL,

ZIP_CODE number(25) NOT NULL,

CUSTOMER_PHONE VARCHAR2(15) NOT NULL,

PRIMARY KEY (CUSTOMER_ID),

UNIQUE (CUSTOMER_PHONE)

);



Create TABLE Shipper (


SHIPPER_ID number(10) NOT NULL,
SHIPPER_NAME varchar2(55) NOT NULL,

PRIMARY KEY (SHIPPER_ID)

);


Create TABLE Wharehouses(


WAREHOUSE_ID number(10) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1
INCREMENT BY 1  NOT NULL,

WAREHOUSE_LOCATION varchar2(100) NOT NULL,

PRIMARY KEY (WAREHOUSE_ID)


);


Create TABLE STORES(

STORES_ID number(10) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1
INCREMENT BY 1  NOT NULL,

STORES_LOCATION varchar2(100) NOT NULL,

PRIMARY Key (STORES_ID)

);


Create TABLE PACKAGES(

PACKAGE_ID number(10) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1
INCREMENT BY 1  NOT NULL,

DIMENSTIONS varchar2(20) NOT NULL,

WEIGHT varchar2(10) NOT NULL,

STORES_ID number(10) NOT NULL,

WAREHOUSE_ID number(10) NOT NULL,

SHIPPER_ID number(10) NOT NULL,

DRONE_ID number(5) NOT NULL,

CUSTOMER_ID number(10) NOT NULL,

PRIMARY Key (PACKAGE_ID),

foreign key (STORES_ID) REFERENCES STORES (STORES_ID),
```

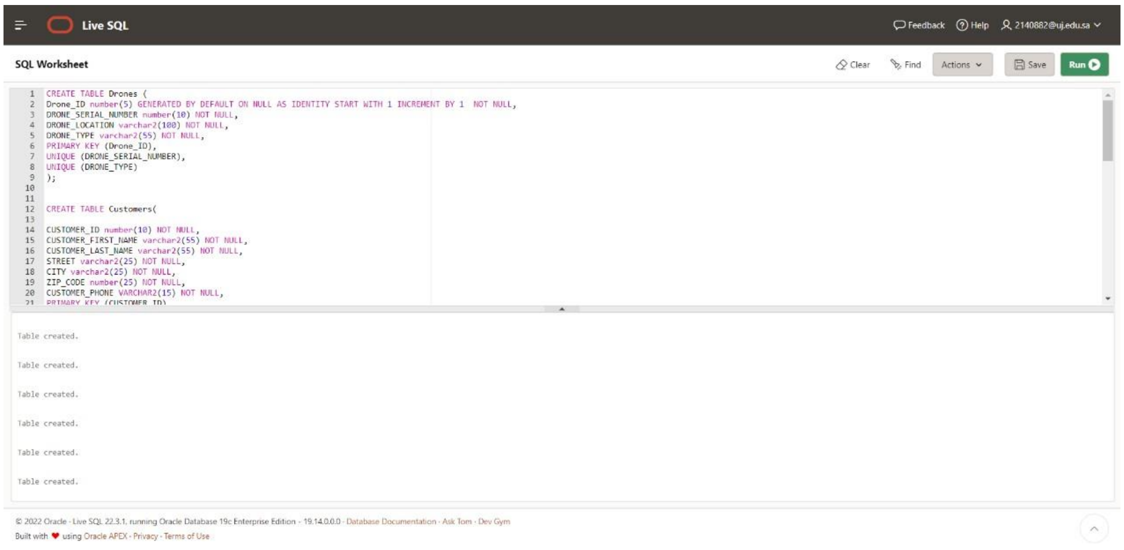
foreign key (WAREHOUSE_ID) REFERENCES Wharehouses (WAREHOUSE_ID),

foreign key (SHIPPER_id) REFERENCES Shipper (SHIPPER_id),

foreign key (DRONE_ID) REFERENCES DRONES (DRONE_ID),

foreign key (CUSTOMER_ID) REFERENCES CUSTOMERS (CUSTOMER_ID)

);



```
CREATE TABLE Drones (
Drone_ID number(5) GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 1 INCREMENT BY 1  NOT NULL,
DRONE_SERIAL_NUMBER number(10) NOT NULL,
DRONE_LOCATION varchar2(100) NOT NULL,
DRONE_TYPE varchar2(55) NOT NULL,
PRIMARY KEY (Drone_ID),
UNIQUE (DRONE_SERIAL_NUMBER),
UNIQUE (DRONE_TYPE)
);


CREATE TABLE Customers(

CUSTOMER_ID number(10) NOT NULL,
CUSTOMER_FIRST_NAME varchar2(55) NOT NULL,
CUSTOMER_LAST_NAME varchar2(55) NOT NULL,
STREET varchar2(25) NOT NULL,
CITY varchar2(25) NOT NULL,
ZIP_CODE number(25) NOT NULL,
CUSTOMER_PHONE VARCHAR2(15) NOT NULL,
PRIMARY KEY (CUSTOMER_ID)
```

```
Table created.

Table created.

Table created.

Table created.

Table created.

Table created.
```

Live SQL
Feedback
Help
2140882@uj.edu.sa
SQL Worksheet
Clear
Find
Actions
Save
Run
© 2022 Oracle · Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

-- DRONES TABLE

INSERT INTO Drones (DRONE_SERIAL_NUMBER, DRONE_LOCATION, DRONE_TYPE) VALUES(11111, 'Jeddah warehouse', 'Single-Rotor');

INSERT INTO Drones (DRONE_SERIAL_NUMBER, DRONE_LOCATION, DRONE_TYPE) VALUES(11112, 'Riyadh warehouse', 'Multi-Rotor');

INSERT INTO Drones (DRONE_SERIAL_NUMBER, DRONE_LOCATION, DRONE_TYPE) VALUES(11113, 'Jeddah warehouse', 'Fixed-Wing');

INSERT INTO Drones (DRONE_SERIAL_NUMBER, DRONE_LOCATION, DRONE_TYPE) VALUES(11114, 'Dammam warehouse', 'Quadcopter');

INSERT INTO Drones (DRONE_SERIAL_NUMBER, DRONE_LOCATION, DRONE_TYPE) VALUES(11115, ' Jizan warehouse', 'Hybrid VTOL');

-- Customers Table

INSERT INTO Customers(CUSTOMER_ID, CUSTOMER_FIRST_NAME, CUSTOMER_LAST_NAME, STREET, CITY, ZIP_CODE, CUSTOMER_PHONE)

```sql
VALUES(1, 'Hassan', 'Adnan', 'Om alqora', 'Jeddah', '23456', '+966124530117');

INSERT INTO Customers(CUSTOMER_ID, CUSTOMER_FIRST_NAME,
CUSTOMER_LAST_NAME, STREET, CITY, ZIP_CODE, CUSTOMER_PHONE)
VALUES(2, 'Maya', 'Ahemd', 'Alsafa', 'Jeddah', '23455', '+966134530116');

INSERT INTO Customers(CUSTOMER_ID, CUSTOMER_FIRST_NAME,
CUSTOMER_LAST_NAME, STREET, CITY, ZIP_CODE, CUSTOMER_PHONE)
VALUES(3, 'Jana', 'Sliman', 'Mushrefa', 'Jeddah', '22233', '+966144500017');

INSERT INTO Customers(CUSTOMER_ID, CUSTOMER_FIRST_NAME,
CUSTOMER_LAST_NAME, STREET, CITY, ZIP_CODE, CUSTOMER_PHONE)
VALUES(4, 'Sliman', 'Nayf', 'Alrehab', 'Jeddah', '122001', '+966124530166');

INSERT INTO Customers(CUSTOMER_ID, CUSTOMER_FIRST_NAME,
CUSTOMER_LAST_NAME, STREET, CITY, ZIP_CODE, CUSTOMER_PHONE)
VALUES(5, 'Abdulrhman', 'Abdullah', 'Ali almortda', 'Jeddah', '62511', '+966110530133');



-- Shipper Table

INSERT INTO Shipper(SHIPPER_ID, SHIPPER_NAME) VALUES(11, 'Kinan');
INSERT INTO Shipper(SHIPPER_ID, SHIPPER_NAME) VALUES(22, 'Burhan');
INSERT INTO Shipper(SHIPPER_ID, SHIPPER_NAME) VALUES(33, 'Badi');
INSERT INTO Shipper(SHIPPER_ID, SHIPPER_NAME) VALUES(44, 'Salman');
INSERT INTO Shipper(SHIPPER_ID, SHIPPER_NAME) VALUES(55, 'Ziad');


-- Wharehouses Table

INSERT INTO Wharehouses(WAREHOUSE_LOCATION) VALUES('JARIR Tahlia, Jeddah, Saudi
Arabia');
```

```
INSERT INTO Wharehouses(WAREHOUSE_LOCATION) VALUES(' JARIR Rehab, Jeddah, Saudi
Arabia');

INSERT INTO Wharehouses(WAREHOUSE_LOCATION) VALUES('JARIR FAIHA, Jeddah, Saudi
Arabia');

INSERT INTO Wharehouses(WAREHOUSE_LOCATION) VALUES(' JARIR ASFAN, Jeddah, Saudi
Arabia');

INSERT INTO Wharehouses(WAREHOUSE_LOCATION) VALUES(' JARIR ALHMDANYHA,
Jeddah, Saudi Arabia');



-- STORES Table


INSERT INTO STORES(STORES_LOCATION) VALUES('J,Jeddah, Saudi Arabia');

INSERT INTO STORES(STORES_LOCATION) VALUES('Jeddah, Saudi Arabia');

INSERT INTO STORES(STORES_LOCATION) VALUES('Jeddah, Saudi Arabia');

INSERT INTO STORES(STORES_LOCATION) VALUES('Jeddah, Saudi Arabia');

INSERT INTO STORES(STORES_LOCATION) VALUES('Jeddah, Saudi Arabia');




-- Packages Table


INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID ,
SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )

VALUES( '30 X 50 X 12', '50 KG', 1, 2, 55, 1, 1);


INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID ,
SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )

VALUES('2 X 2 X 2', '10 KG', 1, 2, 55, 1, 2);


INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID ,
SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )

VALUES('3 X 4 X 15', '30 KG', 2, 4, 33, 1, 3);
```

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID , DRONE_ID)

VALUES('10 X 5 X 6', '20 KG', 3, 4, 22, 2, 4);

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID , DRONE_ID)

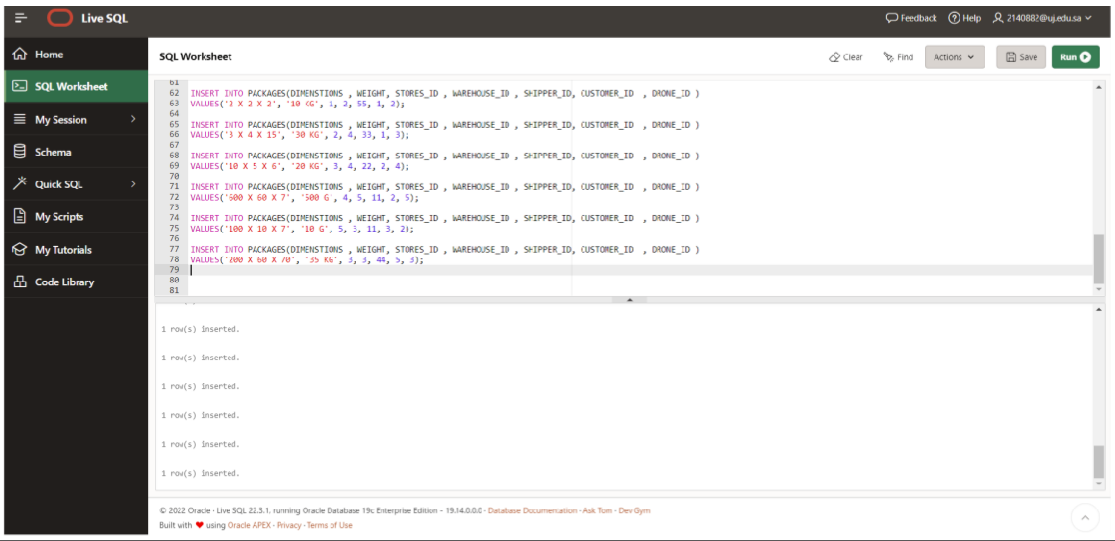
VALUES('600 X 60 X 7', '500 G', 4, 5, 11, 2, 5);

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID , DRONE_ID)

VALUES('100 X 10 X 7', '10 G', 5, 3, 11, 3, 2);

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID , DRONE_ID)

VALUES('200 X 60 X 70', '35 KG', 3, 3, 44, 5, 3);



```
INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )
VALUES('2 X 2 X 2', '10 KG', 1, 2, 55, 1, 2);

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )
VALUES('3 X 4 X 15', '30 KG', 2, 4, 33, 1, 3);

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )
VALUES('10 X 5 X 6', '20 KG', 3, 4, 22, 2, 4);

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )
VALUES('500 X 60 X 7', '500 G', 4, 5, 11, 2, 5);

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )
VALUES('100 X 10 X 7', '10 G', 5, 3, 11, 3, 2);

INSERT INTO PACKAGES(DIMENSTIONS , WEIGHT, STORES_ID , WAREHOUSE_ID , SHIPPER_ID, CUSTOMER_ID  , DRONE_ID )
VALUES('200 X 60 X 70', '35 KG', 3, 3, 44, 5, 3);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

© 2022 Oracle · Live SQL 22.5.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

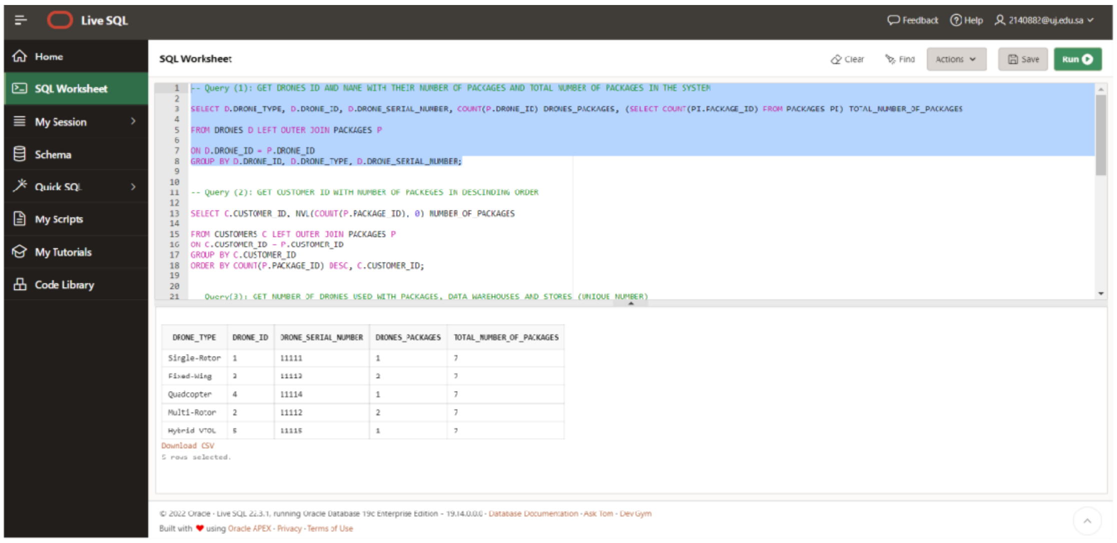
-- Query (1): GET DRONES ID AND NAME WITH THEIR NUMBER OF PACKAGES AND TOTAL NUMBER OF PACKAGES IN THE SYSTEM

SELECT D.DRONE_TYPE, D.DRONE_ID, D.DRONE_SERIAL_NUMBER, COUNT(P.DRONE_ID) DRONES_PACKAGES, (SELECT COUNT(PI.PACKAGE_ID) FROM PACKAGES PI) TOTAL_NUMBER_OF_PACKAGES

FROM DRONES D LEFT OUTER JOIN PACKAGES P

ON D.DRONE_ID = P.DRONE_ID

GROUP BY D.DRONE_ID, D.DRONE_TYPE, D.DRONE_SERIAL_NUMBER;



```
-- Query (1): GET DRONES ID AND NANE WITH THEIR NUMBER OF PACKAGES AND TOTAL NUMBER OF PACKAGES IN THE SYSTEM

SELECT D.DRONE_TYPE, D.DRONE_ID, D.DRONE_SERIAL_NUMBER, COUNT(P.DRONE_ID) DRONES_PACKAGES, (SELECT COUNT(PI.PACKAGE_ID) FROM PACKAGES PI) TOTAL_NUMBER_OF_PACKAGES

FROM DRONES D LEFT OUTER JOIN PACKAGES P

ON D.DRONE_ID = P.DRONE_ID
GROUP BY D.DRONE_ID, D.DRONE_TYPE, D.DRONE_SERIAL_NUMBER;


-- Query (2): GET CUSTOMER ID WITH NUMBER OF PACKEGES IN DESCINDING ORDER

SELECT C.CUSTOMER_ID, NVL(COUNT(P.PACKAGE_ID), 0) NUMBER_OF_PACKAGES

FROM CUSTOMERS C LEFT OUTER JOIN PACKAGES P
ON C.CUSTOMER_ID = P.CUSTOMER_ID
GROUP BY C.CUSTOMER_ID
ORDER BY COUNT(P.PACKAGE_ID) DESC, C.CUSTOMER_ID;


  Query(3): GET NUMBER OF DRONES USED WITH PACKAGES, DATA WAREHOUSES AND STORES (UNIQUE NUMBER)
```

DRONE_TYPE	DRONE_ID	DRONE_SERIAL_NUMBER	DRONES_PACKAGES	TOTAL_NUMBER_OF_PACKAGES
Single-Rotor	1	11111	1	7
Fixed-Wing	3	11113	2	7
Quadcopter	4	11114	1	7
Multi-Rotor	2	11112	2	7
Hybrid VTOL	5	11115	1	7

Download CSV
5 rows selected.

© 2022 Oracle · Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

-- Query (2): GET CUSTOMER ID WITH NUMBER OF PACKEGES IN DESCINDING ORDER

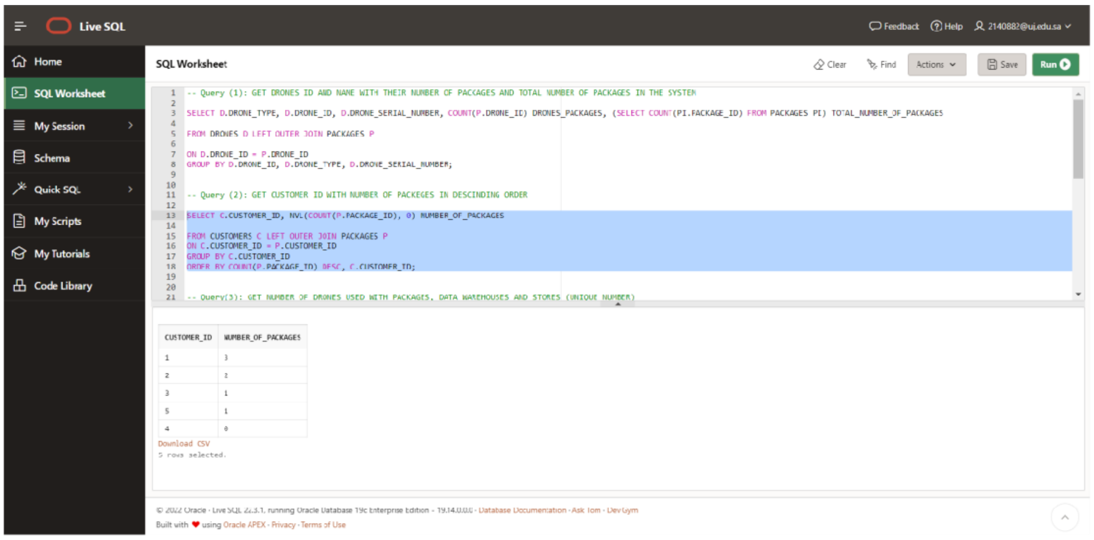
SELECT C.CUSTOMER_ID, NVL(COUNT(P.PACKAGE_ID), 0) NUMBER_OF_PACKAGES

FROM CUSTOMERS C LEFT OUTER JOIN PACKAGES P

ON C.CUSTOMER_ID = P.CUSTOMER_ID

GROUP BY C.CUSTOMER_ID

ORDER BY COUNT(P.PACKAGE_ID) DESC, C.CUSTOMER_ID;



```
-- Query (1): GET DRONES ID AND NANE WITH THEIR NUMBER OF PACKAGES AND TOTAL NUMBER OF PACKAGES IN THE SYSTEM

SELECT D.DRONE_TYPE, D.DRONE_ID, D.DRONE_SERIAL_NUMBER, COUNT(P.DRONE_ID) DRONES_PACKAGES, (SELECT COUNT(PI.PACKAGE_ID) FROM PACKAGES PI) TOTAL_NUMBER_OF_PACKAGES

FROM DRONES D LEFT OUTER JOIN PACKAGES P

ON D.DRONE_ID = P.DRONE_ID
GROUP BY D.DRONE_ID, D.DRONE_TYPE, D.DRONE_SERIAL_NUMBER;


-- Query (2): GET CUSTOMER ID WITH NUMBER OF PACKEGES IN DESCINDING ORDER

SELECT C.CUSTOMER_ID, NVL(COUNT(P.PACKAGE_ID), 0) NUMBER_OF_PACKAGES

FROM CUSTOMERS C LEFT OUTER JOIN PACKAGES P
ON C.CUSTOMER_ID = P.CUSTOMER_ID
GROUP BY C.CUSTOMER_ID
ORDER BY COUNT(P.PACKAGE_ID) DESC, C.CUSTOMER_ID;


-- Query(3): GET NUMBER OF DRONES USED WITH PACKAGES, DATA WAREHOUSES AND STORES (UNIQUE NUMBER)
```

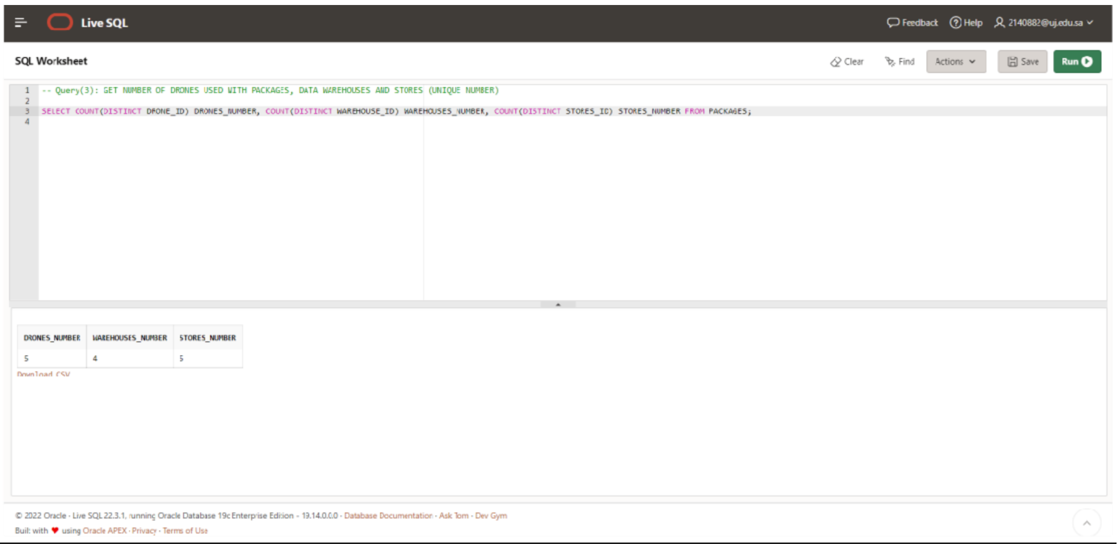
CUSTOMER_ID	NUMBER_OF_PACKAGES
1	3
2	2
3	1
5	1
4	0

Download CSV
5 rows selected.

© 2022 Oracle · Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

-- Query(3): GET NUMBER OF DRONES USED WITH PACKAGES, DATA WAREHOUSES AND STORES (UNIQUE NUMBER)

SELECT COUNT(DISTINCT DRONE_ID) DRONES_NUMBER, COUNT(DISTINCT WAREHOUSE_ID) WAREHOUSES_NUMBER, COUNT(DISTINCT STORES_ID) STORES_NUMBER FROM PACKAGES;



```
-- Query(3): GET NUMBER OF DRONES USED WITH PACKAGES, DATA WAREHOUSES AND STORES (UNIQUE NUMBER)

SELECT COUNT(DISTINCT DRONE_ID) DRONES_NUMBER, COUNT(DISTINCT WAREHOUSE_ID) WAREHOUSES_NUMBER, COUNT(DISTINCT STORES_ID) STORES_NUMBER FROM PACKAGES;
```

DRONES_NUMBER	WAREHOUSES_NUMBER	STORES_NUMBER
5	4	5

Download CSV

© 2022 Oracle · Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

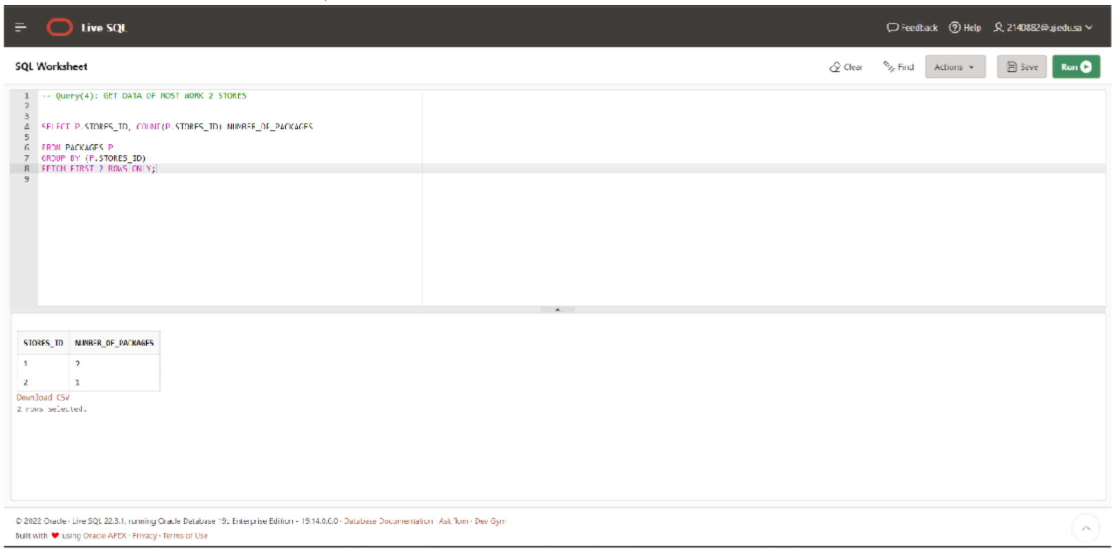
-- Query(4): GET DATA OF MOST WORK 2 STORES

SELECT P.STORES_ID, COUNT(P.STORES_ID) NUMBER_OF_PACKAGES

FROM PACKAGES P

GROUP BY (P.STORES_ID)

FETCH FIRST 2 ROWS ONLY;



```
-- Query(4): GET DATA OF MOST WORK 2 STORES


SELECT P.STORES_ID, COUNT(P.STORES_ID) NUMBER_OF_PACKAGES

FROM PACKAGES P
GROUP BY (P.STORES_ID)
FETCH FIRST 2 ROWS ONLY;
```

STORES_ID	NUMBER_OF_PACKAGES
1	2
2	1

Download CSV
2 rows selected.

© 2022 Oracle · Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 - Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

-- Query(5): GET THE CUSTOMER WITH THE MINIMUM NUMBER OF PACKAGES (NOT ZERO)

SELECT C.CUSTOMER_ID, C.CUSTOMER_FIRST_NAME || ' ' || C.CUSTOMER_LAST_NAME "Customer Full Name", C.STREET ||', ' || C.CITY || ', ' || C.ZIP_CODE "Cutsomer FullAddress", C.CUSTOMER_PHONE

FROM CUSTOMERS C,

(SELECT CUSTOMER_ID, COUNT(CUSTOMER_ID)

NUMBER_OF_PACKAGESFROM PACKAGES P

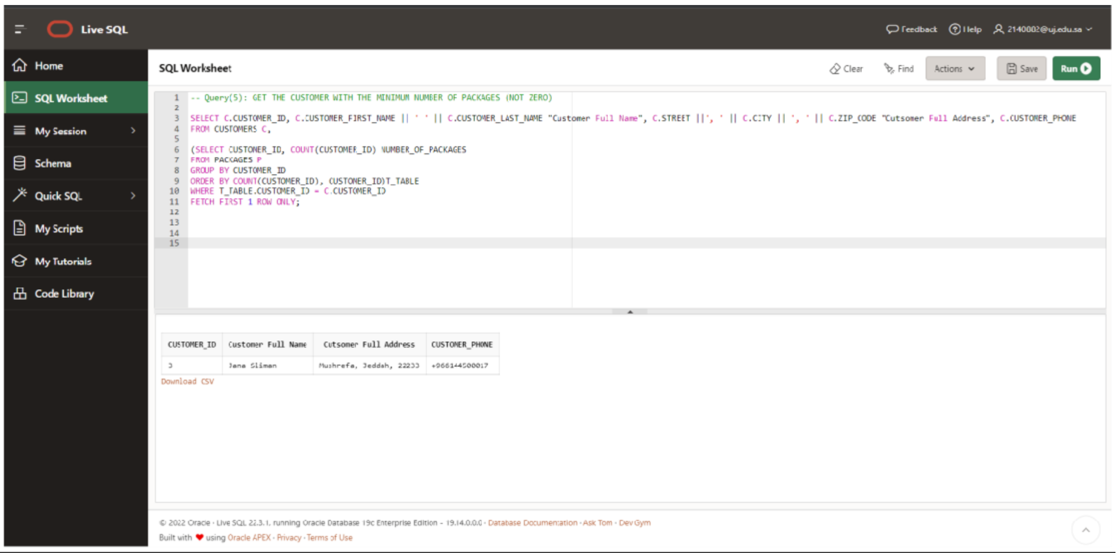
GROUP BY CUSTOMER_ID

ORDER BY COUNT(CUSTOMER_ID),

CUSTOMER_ID)T_TABLEWHERE T_TABLE.CUSTOMER_ID

= C.CUSTOMER_ID

FETCH FIRST 1 ROW ONLY;



```
-- Query(5): GET THE CUSTOMER WITH THE MINIMUM NUMBER OF PACKAGES (NOT ZERO)

SELECT C.CUSTOMER_ID, C.CUSTOMER_FIRST_NAME || ' ' || C.CUSTOMER_LAST_NAME "Customer Full Name", C.STREET ||', ' || C.CITY || ', ' || C.ZIP_CODE "Cutsomer Full Address", C.CUSTOMER_PHONE
FROM CUSTOMERS C,

(SELECT CUSTOMER_ID, COUNT(CUSTOMER_ID) NUMBER_OF_PACKAGES
FROM PACKAGES P
GROUP BY CUSTOMER_ID
ORDER BY COUNT(CUSTOMER_ID), CUSTOMER_ID)T_TABLE
WHERE T_TABLE.CUSTOMER_ID = C.CUSTOMER_ID
FETCH FIRST 1 ROW ONLY;
```

CUSTOMER_ID	Customer Full Name	Cutsomer Full Address	CUSTOMER_PHONE
3	Jana Sliman	Mushrefa, Jeddah, 22233	+966144500017

Download CSV

© 2022 Oracle · Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

Group

1