

a) Add another 5 records in your order product table. Means you will have 15 records for that table.

InvNo	Cust ID	Cust Name	Cust Email	Cust No	Cust Add	Staff ID	Staff Name	Prod ID	Prod Name	Category	Price (RM)	Suppl ID	Suppl Name	Suppl No.	Qty
A01	C01	Kate	kate21@gmail.com	013-4245678	Kajang, Selangor	S01	Nur	P01	Bread	Fresh Food	3.55	G01	Whole Supply	03-90753876	1
A01	C01	Kate	kate21@gmail.com	013-4245678	Kajang, Selangor	S01	Nur	P02	Milk	Drinks	4.88	G02	Bulk Supply	03-87396854	2
A02	C02	Rue	rue99@gmail.com	013-6789528	Klang, Selangor	S02	Muhd.	P03	Apple Juice	Drinks	5.98	G03	Happy Supply	03-90108949	1
A02	C02	Rue	rue99@gmail.com	013-6789528	Klang, Selangor	S02	Muhd.	P04	Eggs	Fresh Food	5.95	G04	Fresh Supply	03-90753876	1
A02	C02	Rue	rue99@gmail.com	013-6789528	Klang, Selangor	S02	Muhd.	P05	Dish Sponge	Household	5.00	G05	Value Supply	03-87372331	1
A03	C03	Ann	ann45@gmail.com	016-7875421	PJ, Selangor	S03	Joe	P06	Ice cream	Frozen	6.29	G03	Happy Supply	03-90108949	1
A03	C03	Ann	ann45@gmail.com	016-7875421	PJ, Selangor	S03	Joe	P07	Paper towel	Household	12.90	G02	Bulk Supply	03-87396854	1
A04	C04	Jane	janedoe@gmail.com	015-6755281	Bangi, Selangor	S01	Nur	P08	Cat food	Pets	19.70	G05	Value Supply	03-87372331	1
A04	C04	Jane	janedoe@gmail.com	015-6755281	Bangi, Selangor	S01	Nur	P04	Eggs	Fresh Food	5.95	G04	Fresh Supply	03-90753876	2
A05	C05	Alex	alex24@gmail.com	016-2985501	Kuala Lumpur	S03	Joe	P01	Bread	Fresh Food	3.55	G01	Whole Supply	03-90753876	1

A05	C05	Alex	alex24 @gmail .com	016- 298 5501	Kuala Lumpur	S03	Joe	P06	Ice cream	Frozen	6.29	G03	Happy Supply	03- 9010 8949	1
A06	C02	Rue	rue99 @gmail .com	013- 678 9528	Klang, Selangor	S02	Muhd.	P01	Bread	Fresh Food	3.55	G01	Whole Supply	03- 9075 3876	2
A06	C02	Rue	rue99 @gmail .com	013- 678 9528	Klang, Selangor	S02	Muhd.	P03	Apple Juice	Drinks	5.98	G03	Happy Supply	03- 9010 8949	1
A06	C02	Rue	rue99 @gmail .com	013- 678 9528	Klang, Selangor	S02	Muhd.	P08	Cat food	Pets	19.70	G05	Value Supply	03- 8737 2331	1
A07	C03	Ann	ann45 @gmail .com	016- 787 5421	PJ, Selangor	S03	Joe	P04	Eggs	Fresh Food	5.95	G04	Fresh Supply	03- 9075 3876	1
A07	C03	Ann	ann45 @gmail .com	016- 787 5421	PJ, Selangor	S03	Joe	P05	Dish Sponge	Household	5.00	G05	Value Supply	03- 8737 2331	1

After adding 5 extra records, these are tables in 3NF.

Table: Order_Invoice

InvNo	ProdID	qty
A01	P01	1
A01	P02	2
A02	P03	1
A02	P04	1
A02	P05	1
A03	P06	1
A03	P07	1
A04	P08	1
A04	P04	2
A05	P01	1
A05	P06	1
A06	P01	2
A06	P03	1
A06	P08	1
A07	P04	1
A07	P05	1

Table: Customer_Staff

InvNo	CustID	StaffID
A01	C01	S01
A02	C02	S02
A03	C03	S03
A04	C04	S01
A05	C05	S03
A07	C02	S02
A08	C03	S03

Table: Product

ProdID	ProdName	Price (RM)	SupplID
P01	Bread	3.55	G01
P02	Milk	4.88	G02
P03	Apple Juice	5.98	G03
P04	Eggs	5.95	G04
P05	Dish Sponge	5.00	G05
P06	Ice cream	6.29	G03
P07	Paper towel	12.90	G02
P08	Cat food	19.70	G05

Table: Product_Category

ProdName	Category
Bread	Fresh Food
Milk	Drinks
Apple Juice	Drinks
Eggs	Fresh Food
Dish Sponge	Household
Ice cream	Frozen
Paper towel	Household
Cat food	Pets

Table: Staff

StaffID	StaffName
S01	Nur
S02	Muhd.
S03	Joe

Table: Customer

CustID	CustName	CustEmail	CustNo	CustAdd
C01	Kate	kate21@gmail.com	013-424 5678	Kajang, Selangor
C02	Rue	rue99@gmail.com	013-678 9528	Klang, Selangor
C03	Ann	ann45@gmail.com	016-787 5421	PJ, Selangor
C04	Jane	janedoe@gmail.com	015-675 5281	Bangi, Selangor
C05	Alex	alex24@gmail.com	016-298 5501	Kuala Lumpur

Table: Supplier

SupplID	SupplName	SupplNo.
G01	Whole Supply	03-9075 3876
G02	Bulk Supply	03-8739 6854
G03	Happy Supply	03-9010 8949
G04	Fresh Supply	03-9075 3876
G05	Value Supply	03-8737 2331

In Microsoft Access, we used SQL to create the tables. For table Invoice_Order, the composite primary key of InvNo and ProdID was done using the design view of the table.

```

Query Type
Order_Invoice-statement
CREATE TABLE Order_Invoice
(
InvNo varchar(3) NOT NULL,
ProdID varchar(3) NOT NULL,
qty int);

```

SQL: Order_Invoice

The screenshot shows the Microsoft Access 'Tables' window. On the left, a list of tables is displayed: Customer, Customer_Staff, Order_Invoice (which is selected and highlighted in pink), Product, Product_Category, Staff, and Supplier. On the right, a grid displays data for the Order_Invoice table. The columns are InvNo, ProdID, qty, and Click. The data consists of 14 rows, each with a unique combination of InvNo and ProdID.

InvNo	ProdID	qty	Click
A01	P01	1	
A01	P02	2	
A02	P03	1	
A02	P04	1	
A02	P05	1	
A03	P06	1	
A03	P07	1	
A04	P04	2	
A04	P08	1	
A05	P01	1	
A05	P06	1	
A06	P01	2	
A06	P03	1	
A06	P08	1	
A07	P04	1	
A07	P05	1	

Table: Order_Invoice

Query Type

customer_staff-statement

```
CREATE TABLE Customer_Staff
(
    InvNo varchar(3) NOT NULL PRIMARY KEY,
    CustID varchar(3) NOT NULL,
    StaffID varchar(3) NOT NULL);
```

SQL: Customer_Staff

Tables

InvNo	CustID	StaffID
A01	C01	S01
A02	C03	S02
A03	C03	S03
A04	C04	S01
A05	C05	S03
A06	C02	S02
A07	C03	S03
*		

Table: Customer_Staff

Query Type

product-statement

```
CREATE TABLE Product
(
    ProdID varchar(3) NOT NULL PRIMARY KEY,
    ProdName varchar(50),
    Price double,
    SupplID varchar(3) NOT NULL);
```

SQL: Product

Tables

ProdID	ProdName	Price	SupplID
P01	Bread	3.55	G01
P02	Milk	4.88	G02
P03	Apple Juice	5.98	G03
P04	Eggs	5.95	G04
P05	Dish Sponge	5	G05
P06	Ice cream	6.39	G03
P07	Paper towel	12.9	G02
P08	Cat food	19.7	G05
*			

Table: Product

customer-statement

```
CREATE TABLE Customer
(
CustID varchar(3) NOT NULL PRIMARY KEY,
CustName varchar(50),
CustEmail varchar,
CustNo varchar(12),
CustAdd varchar(250));
```

SQL: Customer

Tables

The screenshot shows the Microsoft Access 'Tables' view. On the left, a list of tables is shown: Customer, Customer_Staff, Order_Invoice, Product, Product_Category, Staff, and Supplier. The 'Customer' table is highlighted with a pink background. On the right, the 'Customer' table is displayed in a grid format with the following data:

CustID	CustName	CustEmail	CustNo	CustAdd
C01	Kate	kate21@gmail.com	013-4245678	Kajang, Selango
C02	Rue	rue99@gmail.com	013-6789528	Klang, Selangor
C03	Ann	ann45@gmail.com	016-7875421	PJ, Selangor
C04	Jane	janedoe@gmail.com	015-6755281	Bangi, Selangor
C05	Alex	alex24@gmail.com	016-2985501	Kuala Lumpur

Table: Customer

staff-statement

```
CREATE TABLE Staff
(
StaffID varchar(3) NOT NULL PRIMARY KEY,
StaffName varchar(50));
```

SQL: Staff

Tables

The screenshot shows the Microsoft Access 'Tables' view. On the left, a list of tables is shown: Customer, Customer_Staff, Order_Invoice, Product, Product_Category, Staff, and Supplier. The 'Staff' table is highlighted with a pink background. On the right, the 'Staff' table is displayed in a grid format with the following data:

StaffID	StaffName
S01	Nur
S02	Muhd.
S03	Joe

Table: Staff

supplier-statement

```
CREATE TABLE Supplier
(
SupplID varchar(3) NOT NULL PRIMARY KEY,
SupplName varchar(50),
SupplNo varchar(12));
```

SQL: Supplier

Tables

	SupplID	SupplName	SupplNo	Click
[+]	G01	Whole Supply	03-90753876	
[+]	G02	Bulk Supply	03-87396854	
[+]	G03	Happy Supply	03-90108949	
[+]	G04	Fresh Supply	03-90753876	
[+]	G05	Value Supply	03-87372331	
*				

Table: Supplier

product_category-statement

```
CREATE TABLE Product_Category
(
ProdName varchar(50) PRIMARY KEY,
Category varchar(50));
```

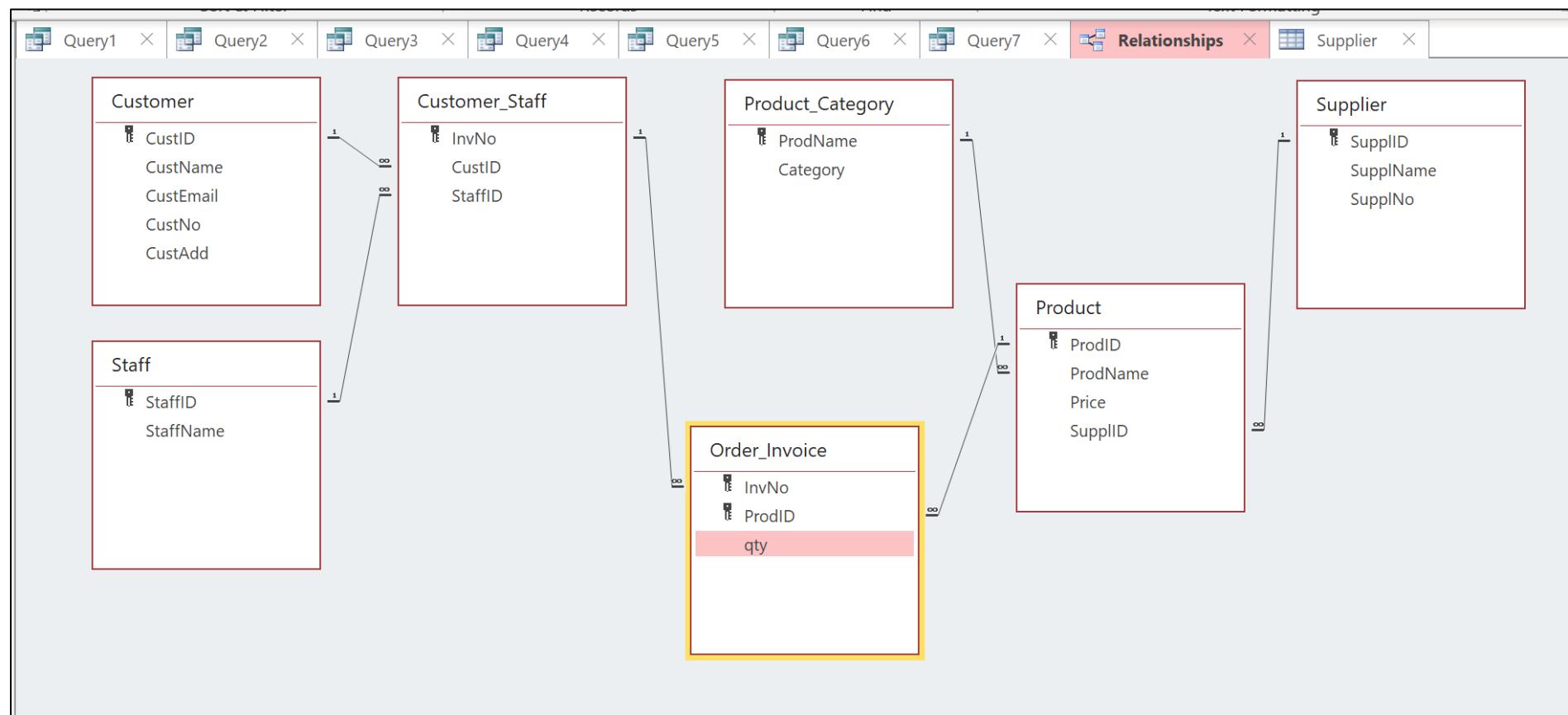
SQL: Product_Category

Tables

	ProdName	Category	Click
[+]	Apple Juice	Drinks	
[+]	Bread	Fresh Food	
[+]	Cat food	Pets	
[+]	Dish Sponge	Household	
[+]	Eggs	Fresh Food	
[+]	Ice cream	Frozen	
[+]	Milk	Drinks	
[+]	Paper towel	Household	
*			

Table: Product_Category

Relationships between the tables are linked by foreign keys which were done using the relationship query in MS Access.

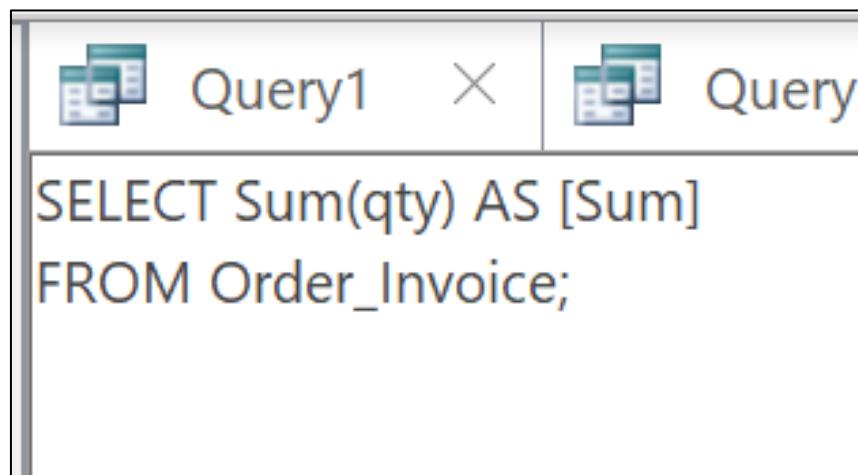


Relationships between tables

b) Create 10 SQL statements that MUST include SQL below:

1. Count/sum/average

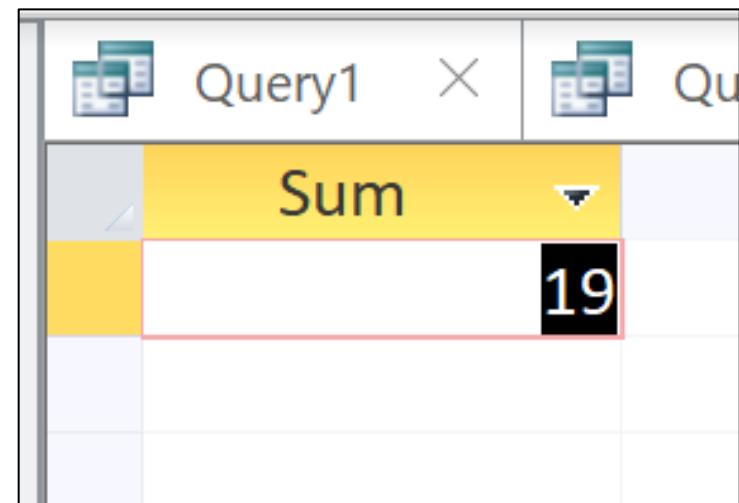
Sum the number of products sold.



The screenshot shows the Microsoft Access interface with two query windows open. The left window is titled "Query1" and contains the following SQL code:

```
SELECT Sum(qty) AS [Sum]
FROM Order_Invoice;
```

SQL: Sum



The screenshot shows the Microsoft Access interface with two query windows open. The right window is titled "Query1" and displays the results of the previous SQL query. The result set has one row with the value "19".

Sum
19

Output: Sum

2. Altering the table

Change the data type of price from the table Product from double to currency.

```
ALTER TABLE Product  
ALTER COLUMN Price currency;
```

SQL: Alter table

ProdID	ProdName	Price	SupplID	Cli
P01	Bread	RM3.55 G01		
P02	Milk	RM4.88 G02		
P03	Apple Juice	RM5.98 G03		
P04	Eggs	RM5.95 G04		
P05	Dish Sponge	RM5.00 G05		
P06	Ice cream	RM6.39 G03		
P07	Paper towel	RM12.90 G02		
P08	Cat food	RM19.70 G05		
*				

Output: Alter table

3. Order by

List the customer ID and customer Name in ascending order of the customer ID.

```
SELECT CustID, CustName  
FROM Customer  
ORDER BY CustID;
```

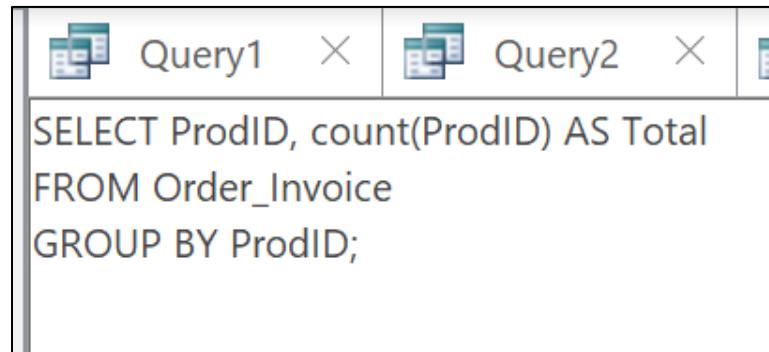
SQL: Order By

CustID	CustName
C01	Kate
C02	Rue
C03	Ann
C04	Jane
C05	Alex
*	

Output: Order By

4. Group by

List the number of products sold for each product ID.



```
SELECT ProdID, count(ProdID) AS Total
FROM Order_Invoice
GROUP BY ProdID;
```

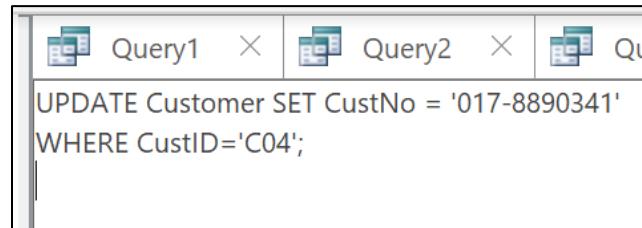
SQL: Group By

ProdID	Total
P01	3
P02	1
P03	2
P04	3
P05	2
P06	2
P07	1
P08	2

Output: Group By

5. Update table

A customer with customer ID = C04 has changed her phone number. Update her new phone number in the table which is 017-8890341.



```
UPDATE Customer SET CustNo = '017-8890341'
WHERE CustID='C04';
```

SQL: Update table

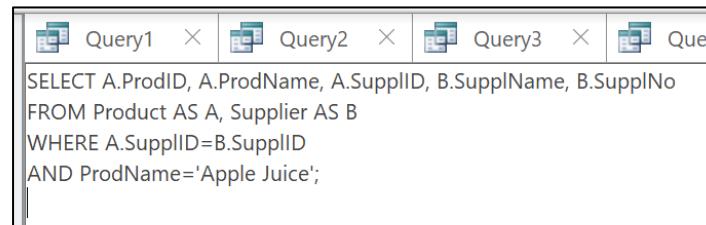
CustID	CustName	CustEmail	CustNo	CustAdd	Click
C01	Kate	kate21@gmail.com	013-4245678	Kajang, Selangor	
C02	Rue	rue99@gmail.com	013-6789528	Klang, Selangor	
C03	Ann	ann45@gmail.com	016-7875421	PJ, Selangor	
C04	Jane	janedoe@gmail.com	017-8890341	Bangi, Selangor	
C05	Alex	alex24@gmail.com	016-2985501	Kuala Lumpur	

Output: Update table

Customer C04 phone number is now updated.

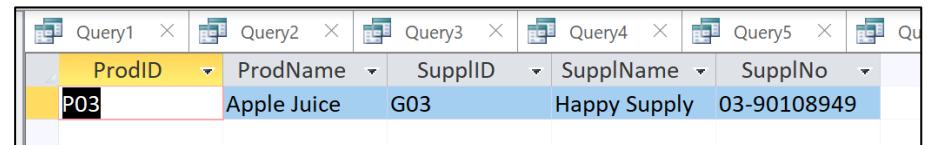
6. Select record/data from multiple tables

The store is running low on Apple Juice. To restock this item, find the supplier ID, name and phone number of Apple Juice from table Product and Supplier.



```
SELECT A.ProdID, A.ProdName, A.SupplID, B.SupplName, B.SupplNo  
FROM Product AS A, Supplier AS B  
WHERE A.SupplID=B.SupplID  
AND ProdName='Apple Juice';
```

SQL: Select record from multiple tables

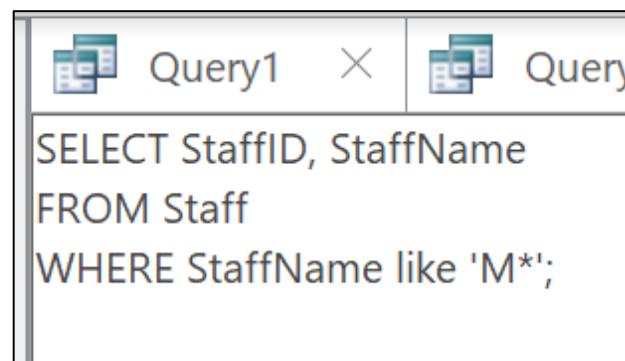


ProdID	ProdName	SupplID	SupplName	SupplNo
P03	Apple Juice	G03	Happy Supply	03-90108949

Output: Select record from multiple tables

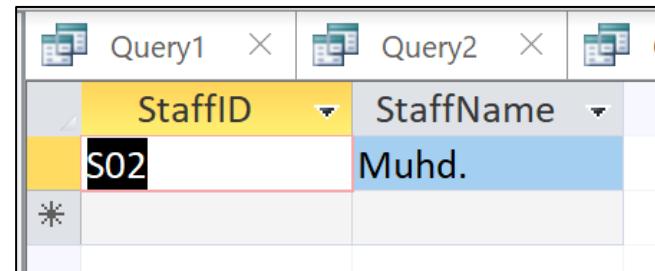
7. Display specific record (in/not in/like/not like)

Display records from the table Staff where the staff's name starts from 'M'.



```
SELECT StaffID, StaffName  
FROM Staff  
WHERE StaffName like 'M*';
```

SQL: Display specific record



StaffID	StaffName
S02	Muhd.
*	

Output: Display specific record

8. Display certain record that have mathematical process

Using SQL, multiply the price of product with the quantity bought by a customer with invoice number = ‘A01’ and display the results.

```
SELECT O.InvNo, O.ProdID, P.Price, O.qty, (Price*qty) AS Total  
FROM Order_Invoice AS O, Product AS P  
WHERE O.ProdID=P.ProdID  
AND O.InvNo='A01';
```

SQL: Display records with mathematical process

The results of Price x qty.

InvNo	ProdID	Price	qty	Total
A01	P01	RM3.55	1	RM3.55
A01	P02	RM4.88	2	RM9.76

Output: Display records with mathematical process

9. Delete certain record

Customer with customer ID C03 is no longer a member. Delete their records from the table Customer.

```
DELETE *  
FROM Customer  
WHERE custID='C03';
```

SQL: Delete record

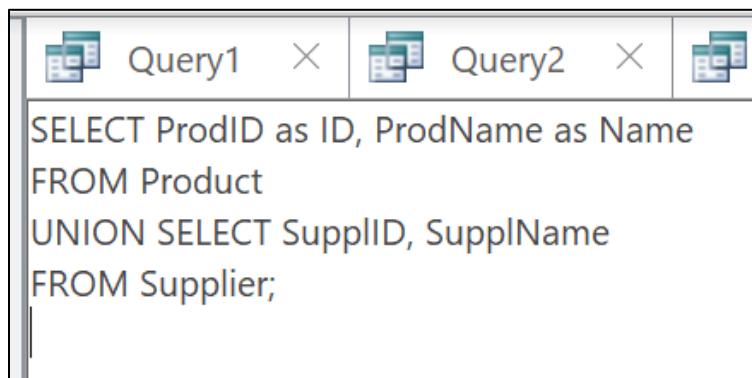
The records of the customer with customer ID C03 no longer exists.

CustID	CustName	CustEmail	CustNo	CustAdd	Click
C01	Kate	kate21@gmail.com	013-4245678	Kajang, Selangor	
C02	Rue	rue99@gmail.com	013-6789528	Klang, Selangor	
C04	Jane	janedoe@gmail.com	017-8890341	Bangi, Selangor	
C05	Alex	alex24@gmail.com	016-2985501	Kuala Lumpur	
*					

Output: Delete record

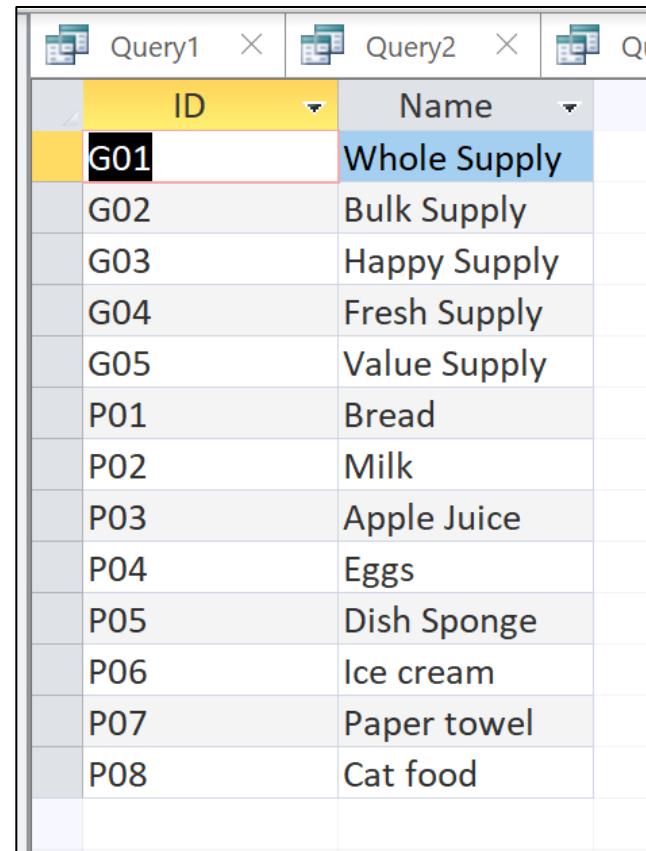
10. Intersect / union / different between 2 tables

List the IDs and Names of products and suppliers using the UNION operator.



```
SELECT ProdID as ID, ProdName as Name
FROM Product
UNION SELECT SupplID, SupplName
FROM Supplier;
```

SQL: Union between 2 tables



ID	Name
G01	Whole Supply
G02	Bulk Supply
G03	Happy Supply
G04	Fresh Supply
G05	Value Supply
P01	Bread
P02	Milk
P03	Apple Juice
P04	Eggs
P05	Dish Sponge
P06	Ice cream
P07	Paper towel
P08	Cat food

Output: Union between 2 tables