LAB # 07

Task 01 :

Find the company's name that placed order 10290. (Tables : Customers & Orders)

(Description)

Inner Join clause in SQL Server creates a new table (not physical) by combining rows that have matching values in two or more tables. This join is based on a logical relationship (or a common field) between the tables and is used to retrieve data that appears in both tables.

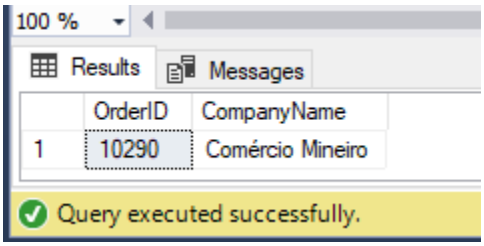
The WHERE clause is used to filter records.It is used to extract only those records that fulfill a specified condition.

The ON Clause makes code easy to understand.ON Clause can be used to join columns that have different names.We use ON clause to specify a join condition.

(Query Text)

```
select orders.OrderID,Customers.CompanyName from Orders inner join Customers on
orders.CustomerID=Customers.CustomerID where orders.OrderID = 10290
```

(Query Output)



××××-------- ××××--------××××--------××××

Task 02 :

Find the Companies that placed orders in 1997 (Tables : Customers & Orders)

(Description)

Inner Join clause in SQL Server creates a new table (not physical) by combining rows that have matching values in two or more tables. This join is based on a logical relationship (or a common field) between the tables and is used to retrieve data that appears in both tables.

The WHERE clause is used to filter records.It is used to extract only those records that fulfill a specified condition.

The ON Clause makes code easy to understand.ON Clause can be used to join columns that have different names.We use ON clause to specify a join condition.

(Query text)

```
select orders.OrderID,Customers.CompanyName from Orders inner join Customers on
orders.CustomerID=Customers.CustomerID where orders.OrderDate = '1997'
```

(Query Output)

100 %

Results | Messages

	OrderID	CompanyName
1	10400	Eastern Connection
2	10401	Rattlesnake Canyon Grocery

Query executed successfully.

××××-------- ××××--------××××--------××××

Task 03 :

Create a report that shows the product name and supplier id for all products supplied by Exotic Liquids, Grandma Kelly's Homestead, and Tokyo Traders. (Tables : Products & Suppliers)

HINT: You will need to escape the apostrophe in "Grandma Kelly's Homestead." To do so, place another apostrophe in front of it. For example,

SELECT *
FROM Suppliers
WHERE CompanyName='Grandma Kelly''s Homestead';

(Description)

Inner Join clause in SQL Server creates a new table (not physical) by combining rows that have matching values in two or more tables. This join is based on a logical relationship (or a common field) between the tables and is used to retrieve data that appears in both tables.

The WHERE clause is used to filter records.It is used to extract only those records that fulfill a specified condition.

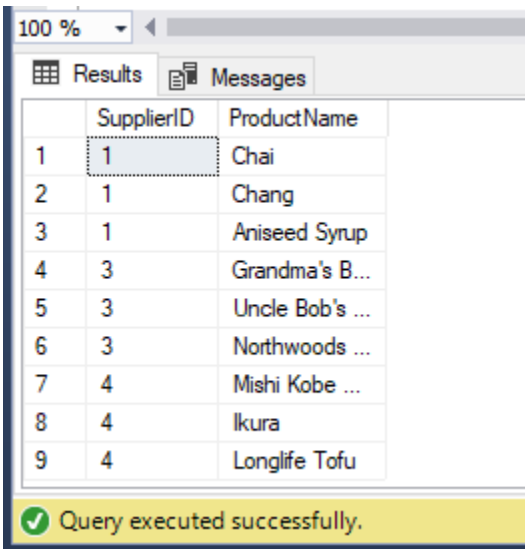
The ON Clause makes code easy to understand.ON Clause can be used to join columns that have different names.We use ON clause to specify a join condition.

The OR operator displays a record if any of the conditions separated by OR is TRUE.

(Query Text)

```
select Suppliers.SupplierID,Products.ProductName from Products inner join Suppliers on
Products.SupplierID=Suppliers.SupplierID where Suppliers.CompanyName = 'exotic liquids'
or Suppliers.CompanyName = 'Grandma Kelly''s Homestead' or Suppliers.CompanyName = 'Tokyo
Traders'
```

(Query Output)



100 %

Results | Messages

	SupplierID	ProductName
1	1	Chai
2	1	Chang
3	1	Aniseed Syrup
4	3	Grandma's B...
5	3	Uncle Bob's ...
6	3	Northwoods ...
7	4	Mishi Kobe ...
8	4	Ikura
9	4	Longlife Tofu

Query executed successfully.

××××-------- ××××--------××××--------××××

Task 04 :

Create a report that shows all products by name that are in the Seafood category. (Tables : Products & Categories)

(Description)

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

The ON Clause makes code easy to understand.ON Clause can be used to join columns that have different names.We use ON clause to specify a join condition.

The AND operator displays a record if all the conditions separated by AND are TRUE

(Query Text)

```
select companyname from Suppliers where SupplierID in (select Products.SupplierID from
Products join Categories on (Categories.CategoryID=Products.CategoryID) and
Categories.CategoryName = 'seafood')
```

(Query Output)

100 %

Results | Messages

	companyname
1	Tokyo Traders
2	Mayumi's
3	Pavlova, Ltd.
4	Nord-Ost-Fisch Handelsgesellschaft mbH
5	Svensk Sjöföda AB
6	New England Seafood Cannery
7	Lyngbysild
8	Escargots Nouveaux

Query executed successfully.

××××-------- ××××--------××××--------××××

Task 05 :

Create a report that shows all companies by name that sell products in CategoryID 8. (Tables : Supplier & Products)

(Description)

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

The ON Clause makes code easy to understand.ON Clause can be used to join columns that have different names.We use ON clause to specify a join condition.

The WHERE clause is used to filter records.It is used to extract only those records that fulfill a specified condition.

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

(Query Text)

```
select products.CategoryID,suppliers.CompanyName from Products join Suppliers on
(products.SupplierID= Suppliers.SupplierID) where CategoryID = 8 group by
suppliers.companyname,Products.CategoryID
```

(Query Output)

100 %

Results | Messages

	CategoryID	CompanyName
1	8	Escargots Nouveaux
2	8	Lyngbysild
3	8	Mayumi's
4	8	New England Seafood Cannery
5	8	Nord-Ost-Fisch Handelsgesellschaft mbH
6	8	Pavlova, Ltd.
7	8	Svensk Sjöföda AB
8	8	Tokyo Traders

Query executed successfully.

××××-------- ××××--------××××--------××××

Task 06 :

Create a report that shows all 5companies by name that sell products in the Seafood category.(Tables: Suppliers, Products & Categories)

(Description)

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

The ON Clause makes code easy to understand.ON Clause can be used to join columns that have different names.We use ON clause to specify a join condition.

The WHERE clause is used to filter records.It is used to extract only those records that fulfill a specified condition.

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

(Query Text)

```
select Suppliers.SupplierID,categories.CategoryID,suppliers.CompanyName
from Categories  inner join Products on Categories.CategoryID=products.CategoryID
inner join Suppliers on  products.SupplierID=Suppliers.SupplierID where
Categories.CategoryID=8 group by
Categories.CategoryID,Suppliers.SupplierID,suppliers.CompanyName
```

(Query Output)

100 %

Results | Messages

	SupplierID	CategoryID	CompanyName
1	4	8	Tokyo Traders
2	6	8	Mayumi's
3	7	8	Pavlova, Ltd.
4	13	8	Nord-Ost-Fisch Handelsgesellschaft mbH
5	17	8	Svensk Sjöföda AB
6	19	8	New England Seafood Cannery
7	21	8	Lyngbysild
8	27	8	Escargots Nouveaux

Query executed successfully.

××××-------- ××××--------××××--------××××

Task 07 :

Write query using a “sub query” to display which Customers were served by which Employee use Northwind

(Description)

The WHERE clause is used to filter records.It is used to extract only those records that fulfill a specified condition.

SQL aliases are used to give a table, or a column in a table, a temporary name.Aliases are often used to make column names more readable.An alias only exists for the duration of that query.An alias is created with the AS keyword.

(Query Text)

```
select Customers.CustomerID as customers ,Employees.FirstName as employee from
Customers,Employees where  Customers.ContactTitle=Employees.Title
```

(Query Output)

100 %

Results | Messages

	customers	employee
1	ALFKI	Nancy
2	ALFKI	Janet
3	ALFKI	Margaret
4	ALFKI	Michael
5	ALFKI	Robert
6	ALFKI	Anne
7	AROUT	Nancy
8	AROUT	Janet

Query executed successfully.

××××-------- ××××--------××××--------××××

Task 11:

Write query using a "sub query" to give the customer id and amount spent of the customer who spent the most using Northwind

(Description)

The SELECT DISTINCT statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

The AND operator displays a record if all the conditions separated by AND are TRUE

The WHERE clause is used to filter records.It is used to extract only those records that fulfill a specified condition.

(Query Text)

```
SELECT distinct Products.ProductName,Products.ProductID , [Order Details].UnitPrice
FROM orders,[Order Details],Products WHERE Products.ProductID=[Order Details].ProductID
and Orders.OrderID = [Order Details].OrderID  order by  [Order Details].UnitPrice desc
```

(Query Output)

100 %

Results | Messages

	ProductName	ProductID	UnitPrice
1	Côte de Blaye	38	263.50
2	Côte de Blaye	38	210.80
3	Thüringer Rostbratwurst	29	123.79
4	Thüringer Rostbratwurst	29	99.00
5	Mishi Kobe Niku	9	97.00
6	Sir Rodney's Marmalade	20	81.00
7	Mishi Kobe Niku	9	77.60
8	Sir Rodney's Marmalade	20	64.80
9	Carnarvon Tigers	18	62.50
10	Raclette Courdavault	59	55.00
11	Manjimup Dried Apples	51	53.00
12	Carnarvon Tigers	18	50.00

Query executed successfully.

××××-------- ××××--------××××--------××××

Task 12:

Write query using a "sub query" to list all Northwind customers who have not placed an order.

(Description)

(Query Text)

```
(select customerid,contactname from Customers) except (select customerid,contactname from
Customers where customerid in (select orders.customerid from orders))
```

(Query Output)

100 %

Results | Messages

	customerid	contactname
1	FISSA	Diego Roel
2	PARIS	Marie Bertrand

Query executed successfully.

××××-------- ××××--------××××--------××××