Assignment No: 4

Name: Pratik Wani

- SQL Queries
  - Here I created a database named as 'Practice' and table named as 'Products' and 'OrderDetails' performed all the Join types.

- Create database, Create tables and Inserted Data:
  - Create is a DDL command
  - Creating a database doesn't automatically set it as the active database, so we need USE command
  - While Creating Table we need to give all the column names there data types and constraints

SQLQuery1.sql - PRA...(PRATIK\dell (51))*

```sql
----Assignment 3

use practice;
-- Create the Products table
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName NVARCHAR(255),
    SupplierID INT,
    CategoryID INT,
    Unit NVARCHAR(50),
    Price DECIMAL(10, 2)
);

-- Insert data into the Products table
INSERT INTO Products (ProductID, ProductName, SupplierID, CategoryID, Unit, Price)
VALUES
(1, 'Chais', 1, 1, '10 boxes x 20 bags', 18),
(2, 'Chang', 1, 1, '24 - 12 oz bottles', 19),
(3, 'Aniseed Syrup', 1, 2, '12 - 550 ml bottles', 10),
(4, 'Chef Anton''s Cajun Seasoning', 2, 2, '48 - 6 oz jars', 22),
(5, 'Chef Anton''s Gumbo Mix', 2, 2, '36 boxes', 21.35),
(6, 'Grandma''s Boysenberry Spread', 3, 2, '12 - 8 oz jars', 25),
(7, 'Uncle Bob''s Organic Dried Pears', 3, 7, '12 - 1 lb pkgs.', 30),
(8, 'Northwoods Cranberry Sauce', 3, 2, '12 - 12 oz jars', 40),
(9, 'Mishi Kobe Niku', 4, 6, '18 - 500 g pkgs.', 97);
```

SQLQuery1.sql - PRA...(PRATIK\dell (51))*

```sql
-- Create the OrderDetails table
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT references products(productid),
    Quantity INT
);


-- Insert data into the OrderDetails table
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
VALUES
(1, 10248, 1, 12),
(2, 10248, 9, 10),
(3, 10248, 7, 5),
(4, 10249, 6, 9),
(5, 10249, 5, 40),
(6, 10250, 4, 10),
(7, 10250, 5, 35),
(8, 10250, 6, 15),
(9, 10251, 2, 6),
(10, 10251, 7, 15);
```

- Select:
  - Select * gives all records

SQLQuery1.sql - PRA...(PRATIK\dell (51))*

```
(9, 10251, 2, 6),
(10, 10251, 7, 15);

select * from Products;
select * from OrderDetails;
```

142 %

Results | Messages

	ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	1	Chais	1	1	10 boxes x 20 bags	18.00
2	2	Chang	1	1	24 - 12 oz bottles	19.00
3	3	Aniseed Syr...	1	2	12 - 550 ml bottles	10.00
4	4	Chef Anton's...	2	2	48 - 6 oz jars	22.00
5	5	Chef Anton's...	2	2	36 boxes	21.35
6	6	Grandma's ...	3	2	12 - 8 oz jars	25.00
7	7	Uncle Bob's...	3	7	12 - 1 lb pkgs.	30.00
8	8	Northwoods...	3	2	12 - 12 oz jars	40.00
9	9	Mishi Kobe ...	4	6	18 - 500 g pkgs.	97.00

	OrderDetailID	OrderID	ProductID	Quantity
1	1	10248	1	12
2	2	10248	9	10
3	3	10248	7	5
4	4	10249	6	9
5	5	10249	5	40
6	6	10250	4	10
7	7	10250	5	35
8	8	10250	6	15
9	9	10251	2	6
10	10	10251	7	15

- <u>Exists:</u>
  - The EXISTS operator is used to test for the existence of any record in a subquery.
  - The EXISTS operator returns TRUE if the subquery returns one or more records.

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```
(10, 10251, 7, 15);

select * from Products;
select * from OrderDetails;

--Exists
SELECT ProductName FROM Products
WHERE exists (SELECT ProductID FROM OrderDetails WHERE Quantity=15);
```

142 %

Results | Messages

	ProductName
1	Chais
2	Chang
3	Aniseed Syrup
4	Chef Anton's Cajun Seasoning
5	Chef Anton's Gumbo Mix
6	Grandma's Boysenberry Spread
7	Uncle Bob's Organic Dried Pears
8	Northwoods Cranberry Sauce
9	Mishi Kobe Niku

- <u>Any:</u>
  - returns a boolean value as a result
  - returns TRUE if ANY of the subquery values meet the condition

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--Any
--1)
SELECT ProductName, productid
FROM Products WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE Quantity=15);

 --2)
SELECT ProductName, ProductID
FROM Products
WHERE ProductID=ANY(SELECT ProductID FROM OrderDetails WHERE Quantity>12);


--3)
SELECT ProductName
FROM Products
WHERE ProductID=ANY(SELECT ProductID FROM OrderDetails WHERE Quantity > 1000);
```

142 %

Results | Messages

	ProductName	productid
1	Grandma's Boysenberry Spread	6
2	Uncle Bob's Organic Dried Pears	7

	ProductName	ProductID
1	Chef Anton's Gumbo Mix	5
2	Grandma's Boysenberry Spread	6
3	Uncle Bob's Organic Dried Pears	7

	ProductName

- <u>All:</u>
  - Returns a boolean value as a result
  - Returns TRUE if ALL of the subquery values meet the condition
  - Is used with SELECT, WHERE and HAVING statements

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--All
SELECT ProductName
FROM Products
WHERE ProductID=ALL(SELECT ProductID FROM OrderDetails WHERE Quantity>0);
```

142 %

Results | Messages

ProductName

- <u>Group By and Having:</u>

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--group by
select  products.CategoryID,count(*) total_cat from products inner join
orderdetails on products.ProductID=orderdetails.ProductID
group by products.CategoryID

--group by having
select  products.CategoryID,count(*) total_cat from products inner join
orderdetails on products.ProductID=orderdetails.ProductID
group by products.CategoryID
having count(*)>1
```

142 %

Results | Messages

	CategoryID	total_cat
1	1	2
2	2	5
3	6	1
4	7	2

	CategoryID	total_cat
1	1	2
2	2	5
3	7	2

- <u>Aggregate functions:</u>
  - Sum: Returns the Addition of the values
  - Count: Returns the Total count used with the group by statement
  - Avg: Returns the Average value of the values
  - Max: Returns the Max values
  - Min: Return the Min values

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--  Aggregate function:-

--sum
select sum(quantity) total_ordered_products from OrderDetails

--count
select CategoryID,count(*) as total_product
from products group by CategoryID;

--average
select avg(price) Avg_Price from products;

--Max and Min
select max(price) max_Price, min(price) min_price from products;
```

142 %

Results | Messages

	total_ordered_products
1	157

	CategoryID	total_product
1	1	2
2	2	5
3	6	1
4	7	1

	Avg_Price
1	31.372222

	max_Price	min_price
1	97.00	10.00

- <u>String functions:</u>
  - Ascii: Returns the Ascii value for the column or value which we pass.
  - Char: Char is just opposite of Ascii. It returns the char associated with the ascii number

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--string functions:-

--Ascii
select ProductName, ascii(ProductName) asciiValue from products;

--char
select char(65) as char;
```

142 %

Results | Messages

	ProductName	asciiValue
1	Chais	67
2	Chang	67
3	Aniseed Syrup	65
4	Chef Anton's Cajun Seasoning	67
5	Chef Anton's Gumbo Mix	67
6	Grandma's Boysenberry Spread	71
7	Uncle Bob's Organic Dried Pears	85
8	Northwoods Cranberry Sauce	78
9	Mishi Kobe Niku	77

	char
1	A

- Len: Returns the length of strings
- Lower: Convert the string into lower case
- Upper: Converts the string into upper case

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--len
select productname, len(productname) length from products;

--Lower and Upper

select Productname, Lower(ProductName) as Lowercase, Upper(ProductName) as  Uppercase
from products;
```

142 %

Results | Messages

	productname	length
1	Chais	5
2	Chang	5
3	Aniseed Syrup	13
4	Chef Anton's Cajun Seasoning	28
5	Chef Anton's Gumbo Mix	22
6	Grandma's Boysenberry Spread	28
7	Uncle Bob's Organic Dried Pears	31
8	Northwoods Cranberry Sauce	26
9	Mishi Kobe Niku	15

	Productname	Lowercase	Uppercase
1	Chais	chais	CHAIS
2	Chang	chang	CHANG
3	Aniseed Syrup	aniseed syrup	ANISEED SYRUP
4	Chef Anton's Cajun Seasoning	chef anton's cajun seasoning	CHEF ANTON'S CAJUN SEASONING
5	Chef Anton's Gumbo Mix	chef anton's gumbo mix	CHEF ANTON'S GUMBO MIX
6	Grandma's Boysenberry Spread	grandma's boysenberry spread	GRANDMA'S BOYSENBERRY SPREAD
7	Uncle Bob's Organic Dried Pe...	uncle bob's organic dried pe...	UNCLE BOB'S ORGANIC DRIED PEA...
8	Northwoods Cranberry Sauce	northwoods cranberry sauce	NORTHWOODS CRANBERRY SAUCE
9	Mishi Kobe Niku	mishi kobe niku	MISHI KOBE NIKU

- Reverse: Reverse the strings chars
- Str: Convert the integer values into strings

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--len
select productname, len(productname) length from products;

--Lower and Upper

select Productname, Lower(ProductName) as Lowercase, Upper(ProductName) as  Uppercase
from products;
```

142 %

Results | Messages

	productname	length
1	Chais	5
2	Chang	5
3	Aniseed Syrup	13
4	Chef Anton's Cajun Seasoning	28
5	Chef Anton's Gumbo Mix	22
6	Grandma's Boysenberry Spread	28
7	Uncle Bob's Organic Dried Pears	31
8	Northwoods Cranberry Sauce	26
9	Mishi Kobe Niku	15

	Productname	Lowercase	Uppercase
1	Chais	chais	CHAIS
2	Chang	chang	CHANG
3	Aniseed Syrup	aniseed syrup	ANISEED SYRUP
4	Chef Anton's Cajun Seasoning	chef anton's cajun seasoning	CHEF ANTON'S CAJUN SEASONING
5	Chef Anton's Gumbo Mix	chef anton's gumbo mix	CHEF ANTON'S GUMBO MIX
6	Grandma's Boysenberry Spread	grandma's boysenberry spread	GRANDMA'S BOYSENBERRY SPREAD
7	Uncle Bob's Organic Dried Pe...	uncle bob's organic dried pe...	UNCLE BOB'S ORGANIC DRIED PEA...
8	Northwoods Cranberry Sauce	northwoods cranberry sauce	NORTHWOODS CRANBERRY SAUCE
9	Mishi Kobe Niku	mishi kobe niku	MISHI KOBE NIKU

- <u>Date Functions:</u>
  - Getdate: Returns the current date
  - Dateadd: Add months to existed date
  - Datediff: It will return the difference of date, months, years.

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
-- Date Functions:
--Getdate
select getdate() as curr_date;

--dateadd
select dateadd(mm,2,getdate()) as After_2_Months;

--datediff
select datediff(year,'2002-03-10',getdate()) Age;
```

142 %

Results | Messages

	curr_date
1	2024-01-23 08:09:50.387

	After_2_Months
1	2024-03-23 08:09:50.387

	Age
1	22

- Datepart: Return the Specific part of the date
- Month: Return the month from date
- Year: Return the year from date
- Day: Return the day from date

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--datepart
select datepart(mm,'2002-03-10') BirthMonth;
select datepart(year,'2002-03-10') BirthYear;

-- day, month, year
select day ('2002-03-10') BirthDay;
select month ('2002-03-10') BirthMonth;
select year ('2002-03-10') BirthYear;
```

142 %

Results | Messages

	BirthMonth
1	3

	BirthYear
1	2002

	BirthDay
1	10

	BirthMonth
1	3

	BirthYear
1	2002

- <u>Mathematical Functions:</u>
  - ABS: Returns the absolute values
  - Sin: Returns the value of angle in radian
  - Ceiling: Returns the greatest to the specified value

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--Mathematical Functions
--ABS
select abs(-101) as Absolute_value;

--sin
select sin(90) angle_in_radian;

--ceiling
select ceiling(price) ceil_price from products;
```

142 %

Results | Messages

	Absolute_value
1	101

	angle_in_radian
1	0.893996663600558

	ceil_price
1	18
2	19
3	10
4	22
5	22
6	25
7	30
8	40
9	97

- Exp: Returns the Exponential values
- Log: Returns the logarithmic values

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--exp
select exp(Price) Exponential_values from products;

--log
select log(Price) log_values from products;
```

142 %

Results | Messages

	Exponential_values
1	65659969.1373305
2	178482300.963187
3	22026.4657948067
4	3584912846.13159
5	1871488611.37931
6	72004899337.3859
7	10686474581524.5
8	2.3538526683702E+17
9	1.33833471920427E+42

	log_values
1	2.89037175789616
2	2.94443897916644
3	2.30258509299405
4	3.09104245335832
5	3.06105173967463
6	3.2188758248682
7	3.40119738166216
8	3.68887945411394
9	4.57471097850338

- Floor: Returns the smallest to the specified value
- Round: Round of the values

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--floor
select floor(price) floor_price from products;

--round
select round(price,1) round_price from products;
```

142 %

Results | Messages

	floor_price
1	18
2	19
3	10
4	22
5	21
6	25
7	30
8	40
9	97

	round_price
1	18.00
2	19.00
3	10.00
4	22.00
5	21.40
6	25.00
7	30.00
8	40.00
9	97.00

- <u>Nested Subquery:</u>
  - Query inside Query means subquery
  - A subquery can be nested inside other subqueries which is called nested subquery.

SQLassg4 sql.sql -...e (PRATIK\dell (84))*

```sql
--nested subquery

select productname, productID from products
where productID in (select productID from orderdetails where price
in(select price from orderdetails where OrderID>5));
```

142 %

Results | Messages

	productname	productID
1	Chais	1
2	Chang	2
3	Chef Anton's Cajun Seasoning	4
4	Chef Anton's Gumbo Mix	5
5	Grandma's Boysenberry Spread	6
6	Uncle Bob's Organic Dried Pears	7
7	Mishi Kobe Niku	9