

**Group name:** 6

## **Schema Definitions and Query Implementations**

### **Table Creation:**

Using SQL statement, we created **20** tables as follows:

- **Client** =

```
CREATE TABLE [dbo].[Client](
    [id] [int] NOT NULL,
    [EMAIL] [nvarchar](50) NULL,
    [PASSWORD] [nvarchar](50) NULL,
    [FRIST_NAME] [nvarchar](50) NULL,
    [LAST_NAME] [nvarchar](50) NULL,
    [CLINT_ADDRESS] [nvarchar](50) ,
    CONSTRAINT [PK_Client] PRIMARY KEY CLUSTERED
)
```

- **BILLING** =

```
CREATE TABLE [dbo].[BILLING](
    [BILLING_ID] [int] NOT NULL,
    [WIN_ID] [int] NULL,
    [order_Date] [date] NULL,
    [Quantity] [nvarchar](50) NULL,
    CONSTRAINT [PK_BILLING] PRIMARY KEY CLUSTERED
)
```

- **Bid** =

```
CREATE TABLE [dbo].[Bid](
    [Bid_id] [int] NOT NULL,
    [price] [numeric](18, 0) NULL,
    [Time_Stamp] [date] NOT NULL,
    [Auction_id] [int] NULL,
    [pur_id] [int] NULL,
    CONSTRAINT [PK_Bid] PRIMARY KEY CLUSTERED
)
```

- **AUCTIONS =**

```
CREATE TABLE [dbo].[AUCTIONS](  
    [AUCTION_ID] [int] NOT NULL,  
    [START_PRICE] [numeric](18, 0) NULL,  
    [Description] [nvarchar](200) NULL,  
    [Reserve] [nvarchar](50) NULL,  
    [increment] [int] NULL,  
    [Expiration] [nvarchar](50) NULL,  
    [client_id] [int] NULL,  
CONSTRAINT [PK_AUCTIONS] PRIMARY KEY CLUSTERED  
)
```

- **CLIENT\_PHONES =**

```
CREATE TABLE [dbo].[CLIENT_PHONES](  
    [ID] [int] NULL,  
    [PHONE_NUMBER] [int] NULL  
)
```

- **COMMENT =**

```
CREATE TABLE [dbo].[COMMENT](  
    [ID] [int] NULL,  
    [COMMENT_DESCRIPTION] [nvarchar](100) NULL,  
    [RATING] [nvarchar](50) NULL  
)
```

- **HardWare =**

```
CREATE TABLE [dbo].[HardWare](  
    [item_id] [int] NULL,  
    [Storage] [nvarchar](50) NULL,  
    [Speed] [nvarchar](50) NULL,  
    [Memory] [nvarchar](50) NULL  
)
```

- **Has =**

```
CREATE TABLE [dbo].[Has](  
    [item_id] [int] NULL,  
    [Bid_id] [int] NULL  
)
```

- **item =**

```
CREATE TABLE [dbo].[item](  
    [item_id] [int] NOT NULL,  
    [Bid_Price] [numeric](18, 0) NULL,  
    [Title] [varchar](50) NULL,  
    [Description] [nvarchar](50) NULL,  
    [Sup_Id] [int] NULL,  
    CONSTRAINT [PK_item] PRIMARY KEY CLUSTERED  
)
```

- **opreation\_system =**

```
CREATE TABLE [dbo].[opreation_system](  
    [item_id] [int] NULL,  
    [Author] [nvarchar](50) NULL,  
    [cost] [numeric](18, 0) NULL,  
    [version] [nvarchar](50) NULL,  
    [Manufacturer] [nvarchar](50) NULL,  
    [is_system_software] [bit] NULL,  
    [year_of_manufacturing] [date] NULL  
)
```

- **PROCEED =**

```
CREATE TABLE [dbo].[PROCEED](  
    [SUPPLIER_ID] [int] NULL,  
    [WINNER_ID] [int] NULL  
)
```

- **PURCHASER =**

```
CREATE TABLE [dbo].[PURCHASER](  
    [ID] [int] NOT NULL,  
    [SHIPPING_ADDRESS] [nvarchar](50) NOT NULL,  
    CONSTRAINT [PK_PURCHASER] PRIMARY KEY CLUSTERED  
)
```

- **SUPPLIER =**

```
CREATE TABLE [dbo].[SUPPLIER](
    [ID] [int] NOT NULL,
    [ACCOUNT_NUMBER] [int] NULL,
    [employee_id] [int] NULL,
    CONSTRAINT [PK_SUPPLIER] PRIMARY KEY CLUSTERED
)
```

- **WINNER =**

```
CREATE TABLE [dbo].[WINNER](
    [ID] [int] NOT NULL,
    CONSTRAINT [PK_WINNER] PRIMARY KEY CLUSTERED
)
```

- **DeleveryMan =**

```
CREATE TABLE [dbo].[DeleveryMan](
    [Phone_number] [int] NOT NULL,
    [WorkingHours] [int] NULL,
    [FRIST_NAME] [nvarchar](50) NULL,
    [LAST_NAME] [nvarchar](50) NULL,
    CONSTRAINT [PK_DeleveryMan] PRIMARY KEY CLUSTERED
)
```

- **Warehouse =**

```
CREATE TABLE [dbo].[Warehouse](
    [Warehouse_id] [int] NOT NULL,
    [location] [nvarchar](50) NULL,
    CONSTRAINT [PK_Warehouse] PRIMARY KEY CLUSTERED
)
```

- **Employee** =

```
CREATE TABLE [dbo].[Employee](
    [id] [int] NOT NULL,
    [salary] [int] NOT NULL,
    [FRIST_NAME] [nvarchar](50) NULL,
    [LAST_NAME] [nvarchar](50) NULL,
    CONSTRAINT [PK_Employee] PRIMARY KEY CLUSTERED
)
```

- **delivery** =

```
CREATE TABLE [dbo].[delivery](
    [Winner_id] [int] NOT NULL,
    [Phone_number] [int] NOT NULL
    CONSTRAINT [PK_delivery] PRIMARY KEY CLUSTERED
)
```

- **Pickup** =

```
CREATE TABLE [dbo].[Pickup](
    [Warehouse_id] [int] NOT NULL,
    [Phone_number] [int] NOT NULL
    CONSTRAINT [PK_Pickup] PRIMARY KEY CLUSTERED
)
```

- **Ship** =

```
CREATE TABLE [dbo].[Ship](
    [Warehouse_id] [int] NOT NULL,
    [Supplier_id] [int] NOT NULL
    CONSTRAINT [PK_Ship] PRIMARY KEY CLUSTERED
)
```

### **Database State:**

- **INSERTION OF TABLE Client:**

- INSERT INTO Client VALUES (1,'naehal@gmail.com','123','Naehal','Naseem','ABC Street 123')
- INSERT INTO Client VALUES (2,'renad@gmail.com','456','Naehal','Ahamed','ABC Street 456')
- INSERT INTO Client VALUES (3,'reem@gmail.com','111','shaden','mohammed','XYZ Street Saudi Arabia')
- INSERT INTO Client VALUES (4,'naehal@gmail.com','122','maha','Nasir','ABC Street 111')
- INSERT INTO Client VALUES (5,'naehal@gmail.com','133','reem','Hussain','ABC Street Saudi Arabia')
- INSERT INTO Client VALUES (6,'saleh@gmail.com','134','saleh','suliman','ABC Street Saudi Arabia')

- **INSERTION OF TABLE CLIENT\_PHONES:**

- INSERT INTO CLIENT\_PHONES VALUES (1,1233536)
- INSERT INTO CLIENT\_PHONES VALUES (2,2541563)
- INSERT INTO CLIENT\_PHONES VALUES (3,2223536)
- INSERT INTO CLIENT\_PHONES VALUES (4,5866975)
- INSERT INTO CLIENT\_PHONES VALUES (5,9985954)
- INSERT INTO CLIENT\_PHONES VALUES (6, 9985954)

- **INSERTION OF TABLE SUPPLIER:**

- INSERT INTO SUPPLIER VALUES (1,1111111)
- INSERT INTO SUPPLIER VALUES (2,2222222)
- INSERT INTO SUPPLIER VALUES (3,3333333)
- INSERT INTO SUPPLIER VALUES (4,4444444)
- INSERT INTO SUPPLIER VALUES (5,5555555)
- INSERT INTO SUPPLIER VALUES (6,6666666)

- **INSERTION OF TABLE PURCHASER:**

- INSERT INTO PURCHASER VALUES (1,'ABC Street 123')
- INSERT INTO PURCHASER VALUES (2,'ABC Street 456')
- INSERT INTO PURCHASER VALUES (3,'ABC Street 789')
- INSERT INTO PURCHASER VALUES (4,'ABC Street 111')
- INSERT INTO PURCHASER VALUES (5,'ABC Street 122')

- INSERT INTO PURCHASER VALUES (6,'ABC Street 231')

- **INSERTION OF TABLE COMMENT:**

- INSERT INTO COMMENT VALUES (1,'GOOD',4)
- 

- INSERT INTO COMMENT VALUES (2,'NICE',3)
- 

- INSERT INTO COMMENT VALUES (3,'GOOD',5)
- 

- INSERT INTO COMMENT VALUES (4,'BAD',2)
- 

- INSERT INTO COMMENT VALUES (5,'GOOD',4)
- 

- INSERT INTO COMMENT VALUES (6,'Like It',6)

- **INSERTION OF TABLE AUCTIONS:**

- INSERT INTO AUCTIONS VALUES(1,500,'AUCTION 1',"10,'2021-12-31',1)
- 

- INSERT INTO AUCTIONS VALUES(2,700,'AUCTION 2',"15,'2022-01-31',2)
- 

- INSERT INTO AUCTIONS VALUES(3,600,'AUCTION 3',"12,'2022-02-28',3)
- 

- INSERT INTO AUCTIONS VALUES(4,550,'AUCTION 4',"30,'2022-01-25',4)
- 

- INSERT INTO AUCTIONS VALUES(5,650,'AUCTION 5',"25,'2021-11-23',2)
- 

- INSERT INTO AUCTIONS VALUES(6,750,'AUCTION 6',"25,'2021-11-23',4)

- **INSERTION OF TABLE item:**

- INSERT INTO item VALUES (1,100,' COMPUTER ','1)
- 

- INSERT INTO item VALUES (2,50,'RAM','1)
- 

- INSERT INTO item VALUES (3,70,'HARD DRIVE','2)
- 

- INSERT INTO item VALUES (4,55,'LAPTOP','3)
- 

- INSERT INTO item VALUES (5,66,'COMPUTER','3)
- 

- INSERT INTO item VALUES (6,80,' LAPTOP ','2)

- **INSERTION OF TABLE HardWare:**

- INSERT INTO HardWare VALUES (1,2000,2,8)
- 

- INSERT INTO HardWare VALUES (2,4000,2,16)
- 

- INSERT INTO HardWare VALUES (3,1000,2,4)
- 

- INSERT INTO HardWare VALUES (4,6000,2,12)
- 

- INSERT INTO HardWare VALUES (5,2000,2,8)
- 

- INSERT INTO HardWare VALUES (6,7000,2,4)

- **INSERTION OF TABLE opreation\_system:**

```
INSERT INTO opreation_system VALUES (1,'Microsoft',500,1.1,'Microsoft',1,'2020-01-01')
```

---

```
INSERT INTO opreation_system VALUES (2,'Microsoft',300,2.1,'Microsoft',1,'2019-01-01')
```

---

```
INSERT INTO opreation_system VALUES (3,'Microsoft',100,1.21,'Microsoft',1,'2018-01-01')
```

---

```
INSERT INTO opreation_system VALUES (4,'Microsoft',400,3.1,'Microsoft',1,'2017-01-01')
```

---

```
INSERT INTO opreation_system VALUES (5,'Microsoft',700,1.1,'Microsoft',1,'2021-01-01')
```

---

```
INSERT INTO opreation_system VALUES (6,'Microsoft',1000,2.1,'Microsoft',1,'2022-01-01')
```

- **INSERTION OF TABLE Bid:**

- INSERT INTO Bid VALUES (1,600,2021-12-14, 1,1)

---

- INSERT INTO Bid VALUES (2,650, 2021-12-14,1,1)

---

- INSERT INTO Bid VALUES (3,750, 2021-12-14,2,2)

---

- INSERT INTO Bid VALUES (4,630, 2021-12-14,3,3)

---

- INSERT INTO Bid VALUES (5,550, 2021-12-14,4,4)

---

- INSERT INTO Bid VALUES (6,5850, 2021-12-14,6,6)

- **INSERTION OF TABLE Has:**

- INSERT INTO Has VALUES (1,1)

---

- INSERT INTO Has VALUES (2,1)

---

- INSERT INTO Has VALUES (3,1)

---

- INSERT INTO Has VALUES (1,3)

---

- INSERT INTO Has VALUES (4,4)

---

- INSERT INTO Has VALUES (6,4)

- **INSERTION OF TABLE WINNER:**

- INSERT INTO WINNER VALUES(1)

---

- INSERT INTO WINNER VALUES(2)

---

- INSERT INTO WINNER VALUES(3)

---

- INSERT INTO WINNER VALUES(4)

---

- INSERT INTO WINNER VALUES(5)

---

- INSERT INTO WINNER VALUES(6)

- **INSERTION OF TABLE PROCEED:**

- INSERT INTO PROCEED VALUES(1,1)  
-----
- INSERT INTO PROCEED VALUES(2,2)  
-----
- INSERT INTO PROCEED VALUES(3,3)  
-----
- INSERT INTO PROCEED VALUES(4,4)  
-----
- INSERT INTO PROCEED VALUES(5,4)  
-----
- INSERT INTO PROCEED VALUES(6,5)

- **INSERTION OF TABLE BILLING:**

- INSERT INTO BILLING VALUES (1,1,'2022-01-25',10)  
-----
- INSERT INTO BILLING VALUES (2,2,'2022-02-28',15)  
-----
- INSERT INTO BILLING VALUES (3,3,'2022-03-01',20)  
-----
- INSERT INTO BILLING VALUES (4,4,'2022-01-25',15)  
-----
- INSERT INTO BILLING VALUES (5,5,'2022-04-22',13)  
-----
- INSERT INTO BILLING VALUES (6,6,'2022-05-23',11)

- **INSERTION OF TABLE DeleveryMan:**

- INSERT INTO DeleveryMan VALUES (1233536,10,'Renad','saud');  
-----
- INSERT INTO DeleveryMan VALUES (2223536,10,'Reem','Abdulmohsen');  
-----
- INSERT INTO DeleveryMan VALUES (2541563,8,'Neehal','saleh');  
-----
- INSERT INTO DeleveryMan VALUES (5667859,7,'saleh','suliman');  
-----
- INSERT INTO DeleveryMan VALUES (5866975,11,'Shaden','Saleh');  
-----
- INSERT INTO DeleveryMan VALUES (9985954,6,'shoog','Hussain');

- **INSERTION OF TABLE Warehouse :**

- INSERT INTO Warehouse VALUES (1,'ABC Street 123');
- 
- INSERT INTO Warehouse VALUES (2,'ABC Street 456');
- 
- INSERT INTO Warehouse VALUES (3,'ABC Street 789');
- 
- INSERT INTO Warehouse VALUES (4,'ABC Street 111');
- 
- INSERT INTO Warehouse VALUES (5,'ABC Street 122');
- 
- INSERT INTO Warehouse VALUES (6,'ABC Street 231');

- **INSERTION OF TABLE Employee :**

- INSERT INTO Employee VALUES (1,9000,'nora' , 'fahad');
- 
- INSERT INTO Employee VALUES (2,20000, 'sara' , 'saleh');
- 
- INSERT INTO Employee VALUES (3,9000, 'reem' , 'naser');
- 
- INSERT INTO Employee VALUES (4,1000,'maha' , 'fahad');
- 
- INSERT INTO Employee VALUES (5,15000,'renad','saad');
- 
- INSERT INTO Employee VALUES (6,20000,'neehal' , 'saleh');

- **INSERTION OF TABLE Pickup :**

- INSERT INTO Pickup VALUES (1,1233536)
- 
- INSERT INTO Pickup VALUES (2,2541563)
- 
- INSERT INTO Pickup VALUES (3,2223536)
- 
- INSERT INTO Pickup VALUES (4,5866975)
- 
- INSERT INTO Pickup VALUES (5,9985954)
- 
- INSERT INTO Pickup VALUES (6,5667859)

- **INSERTION OF TABLE Ship:**

- INSERT INTO Ship VALUES (1,1);  
-----

- INSERT INTO Ship VALUES (2,2);  
-----

- INSERT INTO Ship VALUES (3,3);  
-----

- INSERT INTO Ship VALUES (4,4);  
-----

- INSERT INTO Ship VALUES (5,5);  
-----

- INSERT INTO Ship VALUES (6,6);

- **INSERTION OF TABLE delivery:**

- INSERT INTO delivery VALUES (1,1233536);  
-----

- INSERT INTO delivery VALUES (2,2541563);  
-----

- INSERT INTO delivery VALUES (3,2223536);  
-----

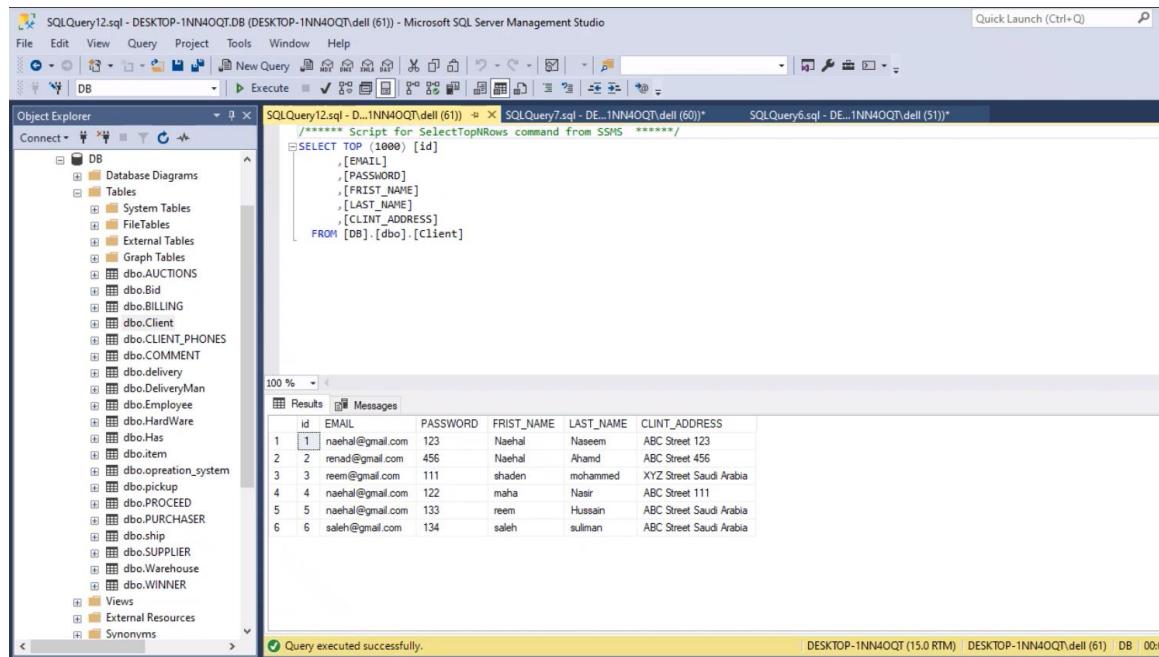
- INSERT INTO delivery VALUES (4,5866975);  
-----

- INSERT INTO delivery VALUES (5,9985954);  
-----

- INSERT INTO delivery VALUES (6,5667859);

**Screenshot:**

- Client:



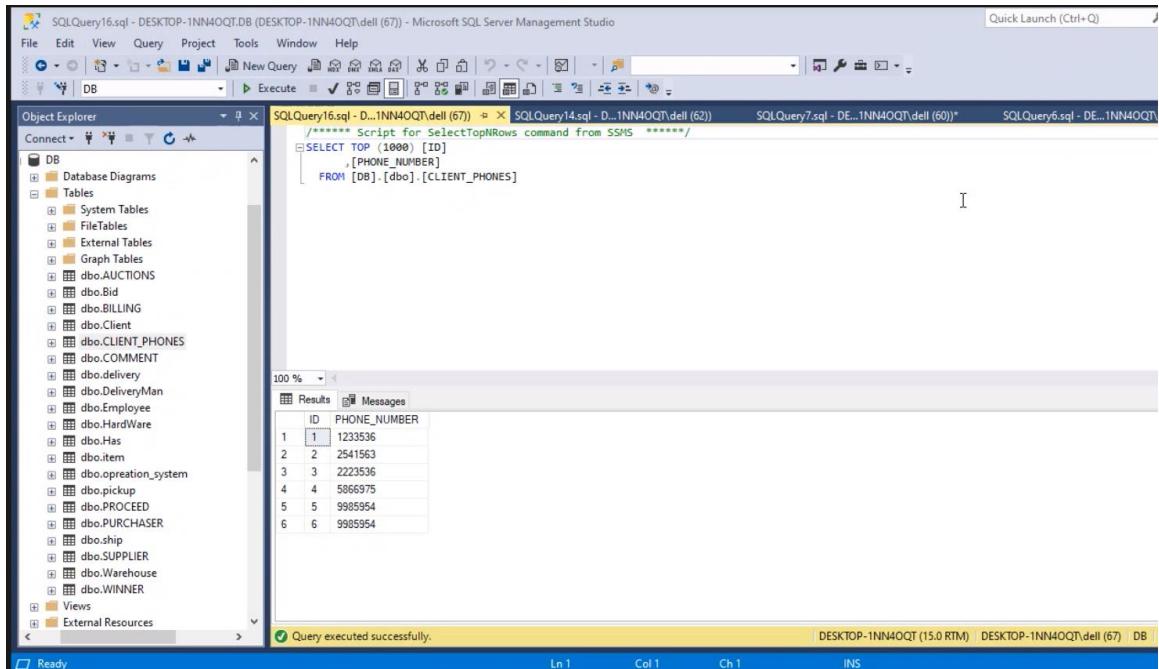
SQLQuery12.sql - DESKTOP-1NN4OQT.DB (DESKTOP-1NN4OQT\dell (61)) - Microsoft SQL Server Management Studio

```
/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [id]
    ,[EMAIL]
    ,[PASSWORD]
    ,[FRIST_NAME]
    ,[LAST_NAME]
    ,[CLINT_ADDRESS]
FROM [DB].[dbo].[Client]
```

	id	EMAIL	PASSWORD	FRIST_NAME	LAST_NAME	CLINT_ADDRESS
1	1	naeelah@gmail.com	123	Naeelah	Naseem	ABC Street 123
2	2	reanad@gmail.com	456	Naealah	Ahamd	ABC Street 456
3	3	reem@gmail.com	111	shaden	mohammed	XYZ Street Saudi Arabia
4	4	naeelah@gmail.com	122	maha	Nair	ABC Street 111
5	5	naealah@gmail.com	133	reem	Hussain	ABC Street Saudi Arabia
6	6	saleh@gmail.com	134	saleh	suliman	ABC Street Saudi Arabia

Query executed successfully.

- CLIENT\_PHONES:



SQLQuery16.sql - DESKTOP-1NN4OQT.DB (DESKTOP-1NN4OQT\dell (67)) - Microsoft SQL Server Management Studio

```
/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [ID]
    ,[PHONE_NUMBER]
FROM [DB].[dbo].[CLIENT_PHONES]
```

ID	PHONE_NUMBER
1	123353
2	2541563
3	2223536
4	5866975
5	9985954
6	9985954

Query executed successfully.

- SUPPLIER:**

```
SQLQuery43.sql - DESKTOP-1NN4OQT.DB (DESKTOP-1NN4OQT.dell (64)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer Results Messages
SQLQuery43.sql - D...1NN4OQT.dell (64) SQLQuery42.sql - D...1NN4OQT.dell (62) SQLQuery41.sql - D...1NN4OQT.dell (59) SQLQuery40.sql - D...1NN4OQT.dell (57)
SELECT TOP (1000) [ID]
, [ACCOUNT_NUMBER]
FROM [DB].[dbo].[SUPPLIER]
```

ID	ACCOUNT_NUMBER
1	1111111
2	2222222
3	3333333
4	4444444
5	5555555
6	6666666

Query executed successfully.

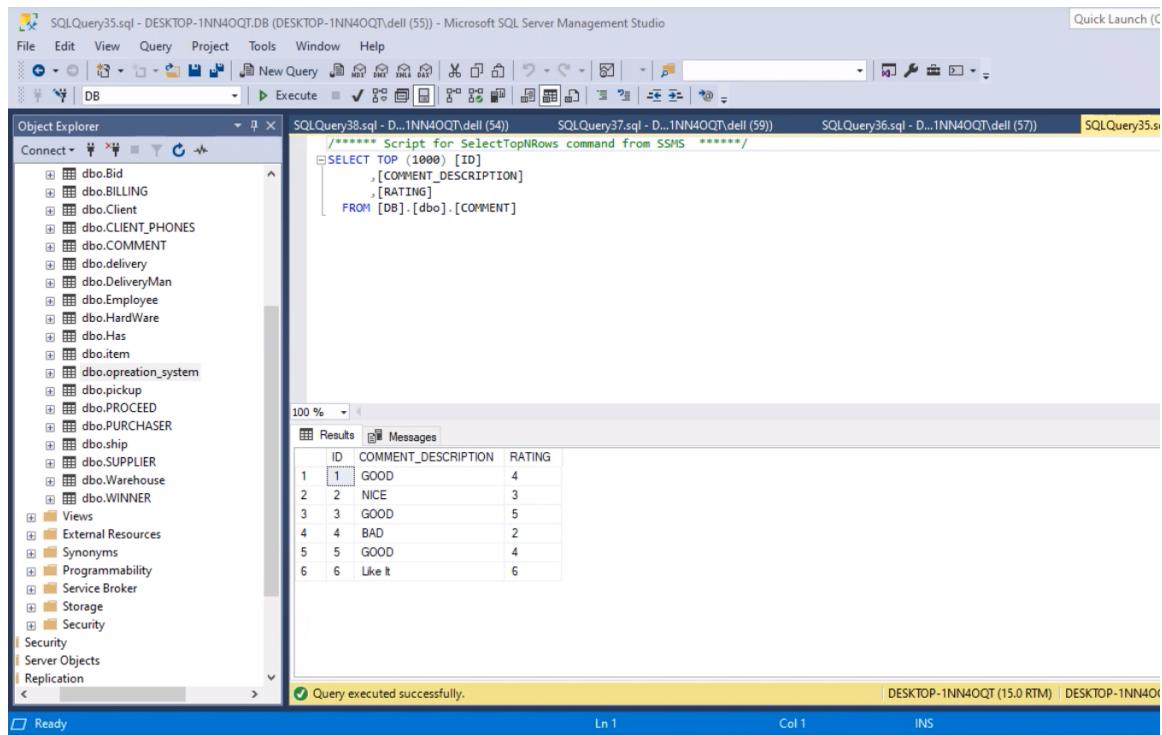
- PURCHASER:**

```
SQLQuery42.sql - DESKTOP-1NN4OQT.DB (DESKTOP-1NN4OQT.dell (62)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer Results Messages
SQLQuery42.sql - D...1NN4OQT.dell (62) SQLQuery41.sql - D...1NN4OQT.dell (59) SQLQuery40.sql - D...1NN4OQT.dell (57) SQLQuery7.sql - DE...1NN4OQT.dell (60)
SELECT TOP (1000) [ID]
,[SHIPPING_ADDRESS]
FROM [DB].[dbo].[PURCHASER]
```

ID	SHIPPING_ADDRESS
1	ABC Street 123
2	ABC Street 456
3	ABC Street 789
4	ABC Street 111
5	ABC Street 122
6	ABC Street 231

Query executed successfully.

- INSERTION OF TABLE COMMENT:**



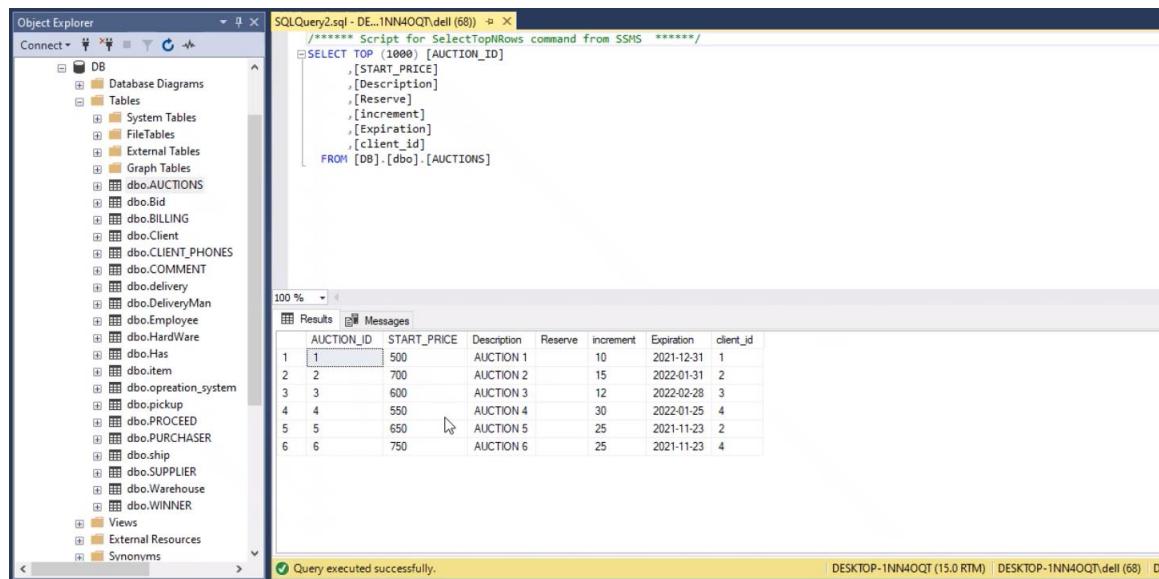
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like tables (e.g., Bid, BILLING, Client, etc.) and views. The central pane displays a query results grid titled 'Results'.

```
SELECT TOP (1000) [ID]
      ,[COMMENT_DESCRIPTION]
      ,[RATING]
  FROM [DB].[dbo].[COMMENT]
```

ID	COMMENT_DESCRIPTION	RATING
1	GOOD	4
2	NICE	3
3	GOOD	5
4	BAD	2
5	GOOD	4
6	Like it	6

At the bottom, a message bar indicates "Query executed successfully."

- INSERTION OF TABLE AUCTIONS:**



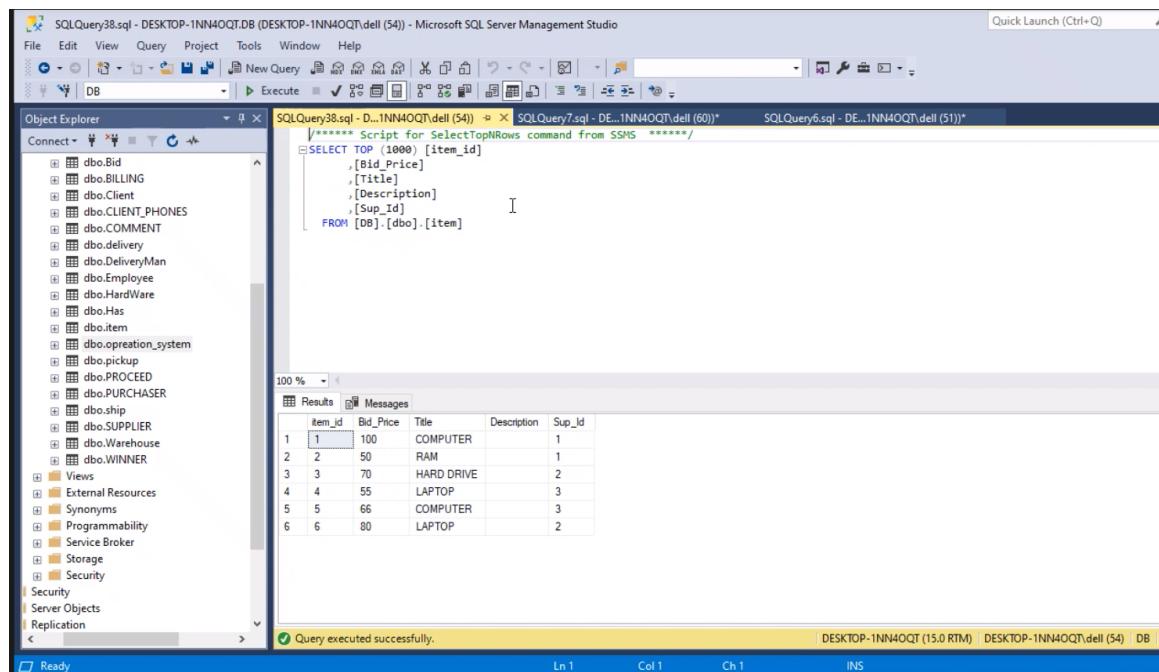
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like tables (e.g., AUCTIONS, Bid, BILLING, Client, etc.) and views. The central pane displays a query results grid titled 'Results'.

```
SELECT TOP (1000) [AUCTION_ID]
      ,[START_PRICE]
      ,[Description]
      ,[Reserve]
      ,[increment]
      ,[Expiration]
      ,[client_id]
  FROM [DB].[dbo].[AUCTIONS]
```

AUCTION_ID	START_PRICE	Description	Reserve	increment	Expiration	client_id
1	500	AUCTION 1	10	2021-12-31	1	
2	700	AUCTION 2	15	2022-01-31	2	
3	600	AUCTION 3	12	2022-02-28	3	
4	550	AUCTION 4	30	2022-01-25	4	
5	650	AUCTION 5	25	2021-11-23	2	
6	750	AUCTION 6	25	2021-11-23	4	

At the bottom, a message bar indicates "Query executed successfully."

- **INSERTION OF TABLE item:**



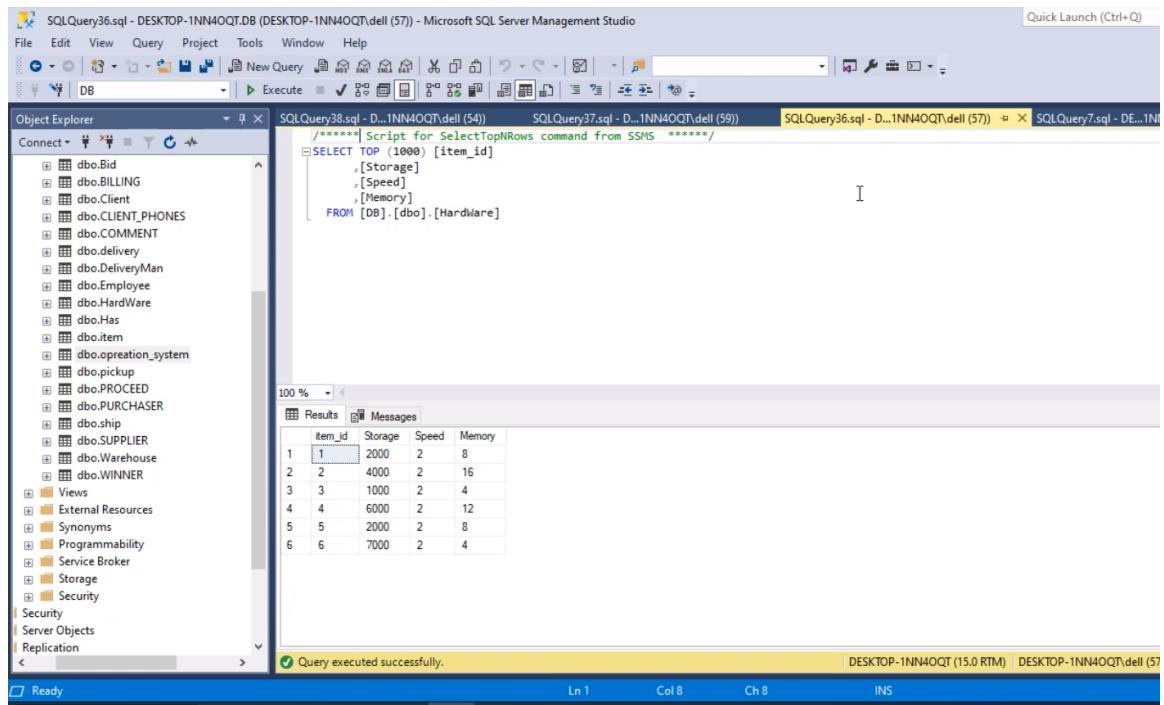
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Bid, BILLING, Client, etc. In the center, a query window displays a SELECT statement and its results. The results show six items with their respective IDs, bid prices, titles, descriptions, and supplier IDs.

```

SELECT TOP (1000) [item_id]
      ,[Bid_Price]
      ,[title]
      ,[Description]
      ,[Sup_Id]
   FROM [DB].[dbo].[item]
  
```

item_id	Bid_Price	Title	Description	Sup_Id
1	100	COMPUTER		1
2	50	RAM		1
3	70	HARD DRIVE		2
4	55	LAPTOP		3
5	66	COMPUTER		3
6	80	LAPTOP		2

- **INSERTION OF TABLE HardWare:**



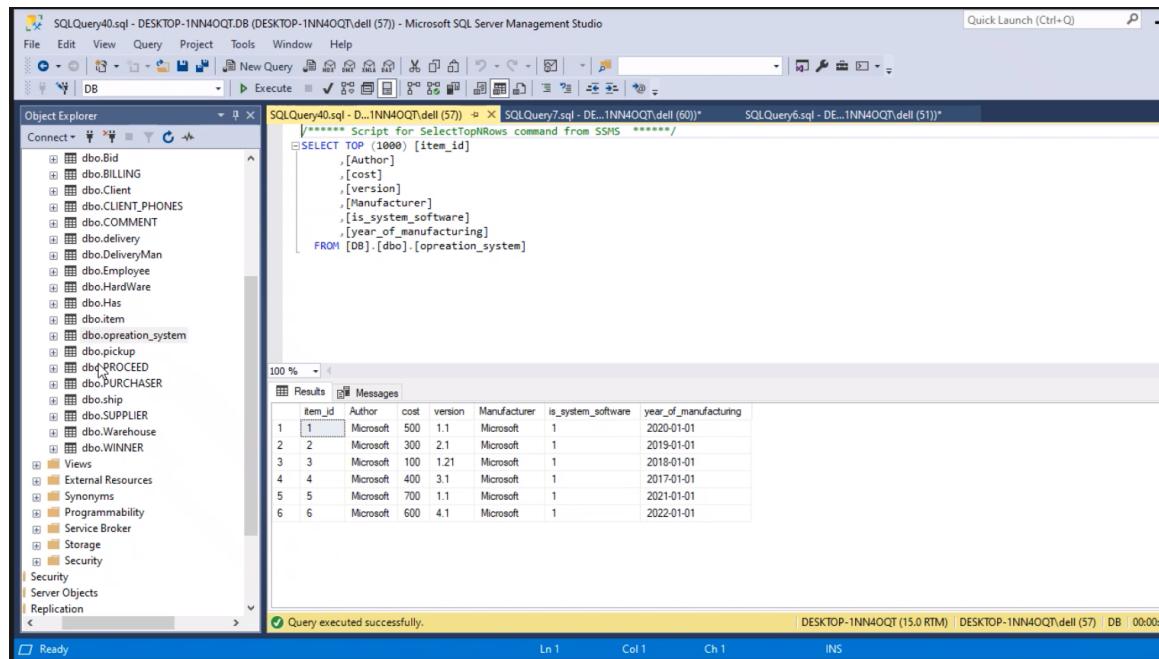
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects. In the center, a query window displays a SELECT statement and its results. The results show six hardware components with their storage, speed, and memory values.

```

SELECT TOP (1000) [item_id]
      ,[Storage]
      ,[Speed]
      ,[Memory]
   FROM [DB].[dbo].[Hardware]
  
```

item_id	Storage	Speed	Memory
1	2000	2	8
2	4000	2	16
3	1000	2	4
4	6000	2	12
5	2000	2	8
6	7000	2	4

- **INSERTION OF TABLE operation\_system:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Bid, BILLING, Client, etc. The central pane displays a T-SQL script:

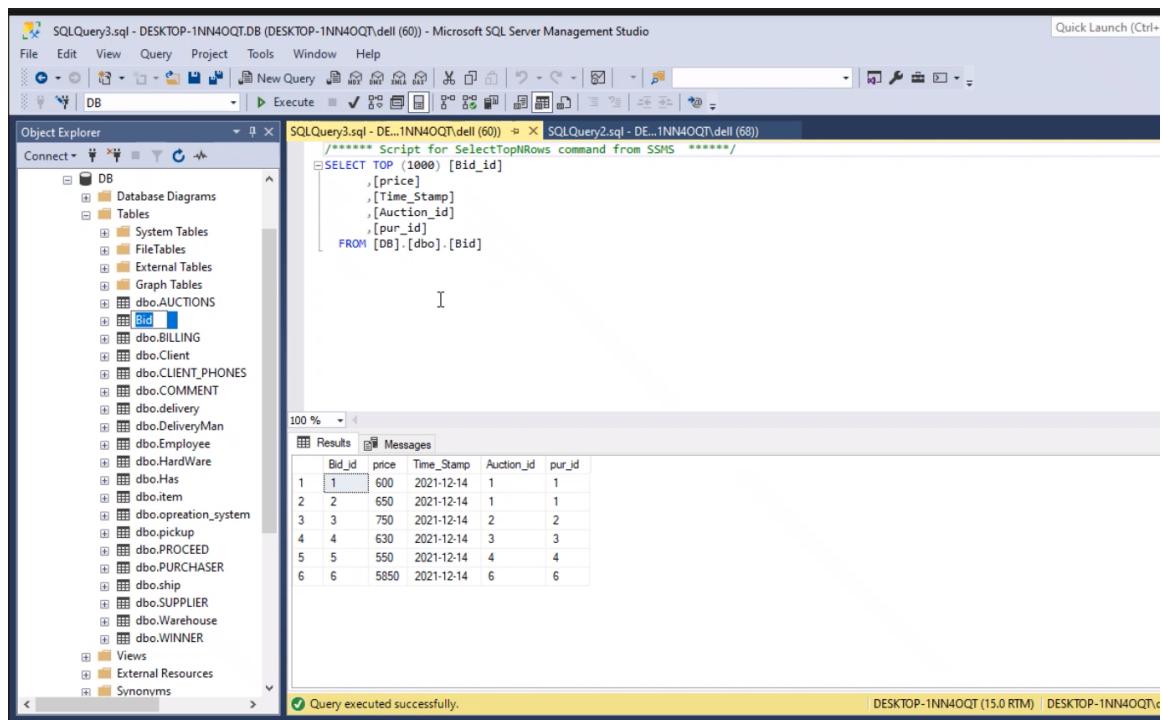
```
SELECT TOP (1000) [item_id]
    ,[Author]
    ,[cost]
    ,[version]
    ,[Manufacturer]
    ,[is_system_software]
    ,[year_of_manufacturing]
FROM [DB].[dbo].[operation_system]
```

Below the script, the Results pane shows a table with six rows of data:

item_id	Author	cost	version	Manufacturer	is_system_software	year_of_manufacturing
1	Microsoft	500	1.1	Microsoft	1	2020-01-01
2	Microsoft	300	2.1	Microsoft	1	2019-01-01
3	Microsoft	100	1.21	Microsoft	1	2018-01-01
4	Microsoft	400	3.1	Microsoft	1	2017-01-01
5	Microsoft	700	1.1	Microsoft	1	2021-01-01
6	Microsoft	600	4.1	Microsoft	1	2022-01-01

At the bottom, a message indicates "Query executed successfully."

- **INSERTION OF TABLE Bid:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like DB, Database Diagrams, Tables, etc. The central pane displays a T-SQL script:

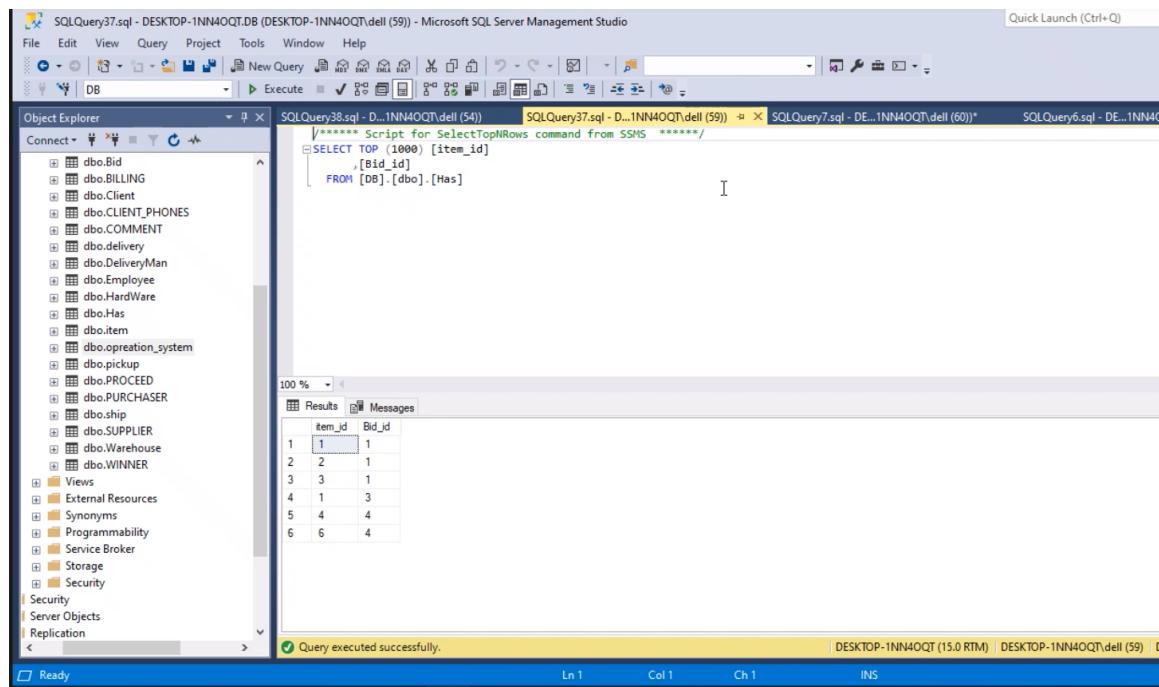
```
SELECT TOP (1000) [Bid_id]
    ,[price]
    ,[Time_Stamp]
    ,[Auction_id]
    ,[pur_id]
FROM [DB].[dbo].[Bid]
```

Below the script, the Results pane shows a table with six rows of data:

Bid_id	price	Time_Stamp	Auction_id	pur_id
1	600	2021-12-14	1	1
2	650	2021-12-14	1	1
3	750	2021-12-14	2	2
4	630	2021-12-14	3	3
5	550	2021-12-14	4	4
6	5850	2021-12-14	6	6

At the bottom, a message indicates "Query executed successfully."

- **INSERTION OF TABLE Has:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Bid, BILLING, Client, etc. The central pane displays a query window with the following SQL script:

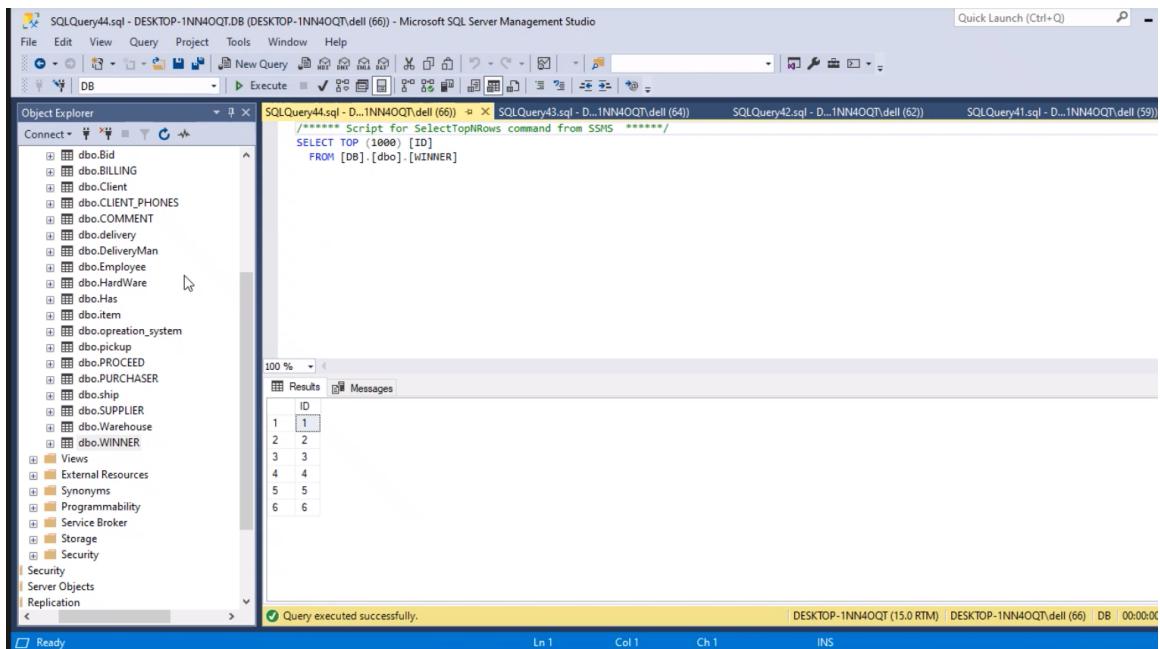
```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [item_id]
      ,[Bid_id]
  FROM [DB].[dbo].[Has]
```

The results pane shows a table with two columns: item\_id and Bid\_id. The data is as follows:

item_id	Bid_id
1	1
2	1
3	1
4	3
5	4
6	4

A status bar at the bottom indicates "Query executed successfully." and shows the session details: DESKTOP-1NN4OQT (15.0 RTM) | DESKTOP-1NN4OQT.dell (59).

- **INSERTION OF TABLE WINNER:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects. The central pane displays a query window with the following SQL script:

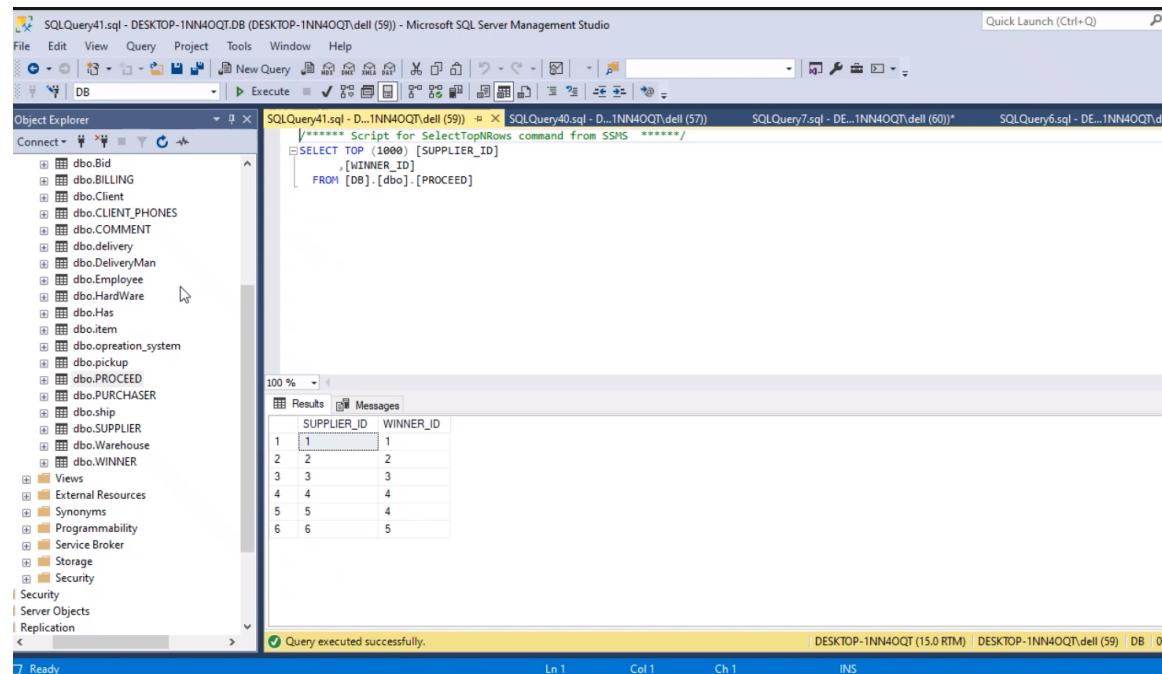
```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [ID]
  FROM [DB].[dbo].[WINNER]
```

The results pane shows a table with one column: ID. The data is as follows:

ID
1
2
3
4
5
6

A status bar at the bottom indicates "Query executed successfully." and shows the session details: DESKTOP-1NN4OQT (15.0 RTM) | DESKTOP-1NN4OQT.dell (66) | DB 00:00:00.

- INSERTION OF TABLE PROCEED:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like tables, views, and stored procedures. The central pane contains a query window with the following SQL script:

```

/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [SUPPLIER_ID]
      ,[WINNER_ID]
   FROM [DB].[dbo].[PROCEED]

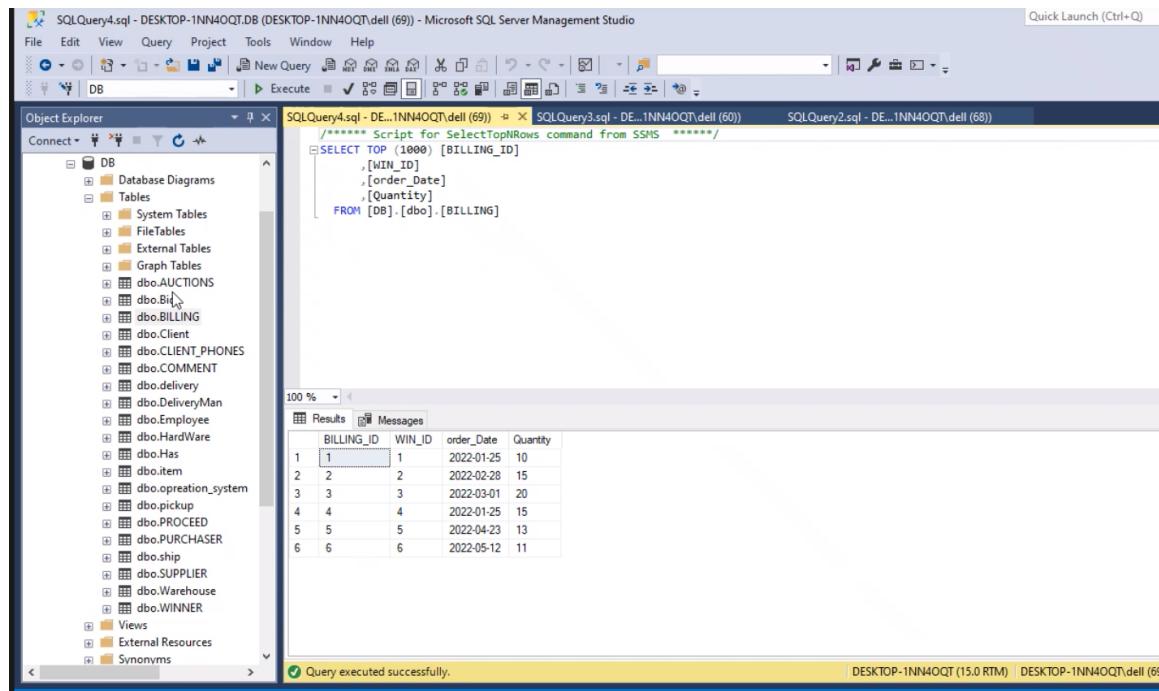
```

The results pane displays a table with two columns: SUPPLIER\_ID and WINNER\_ID. The data is as follows:

SUPPLIER_ID	WINNER_ID
1	1
2	2
3	3
4	4
5	4
6	5

A status bar at the bottom indicates "Query executed successfully."

- INSERTION OF TABLE BILLING:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects. The central pane contains a query window with the following SQL script:

```

/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [BILLING_ID]
      ,[WIN_ID]
      ,[order_Date]
      ,[Quantity]
   FROM [DB].[dbo].[BILLING]

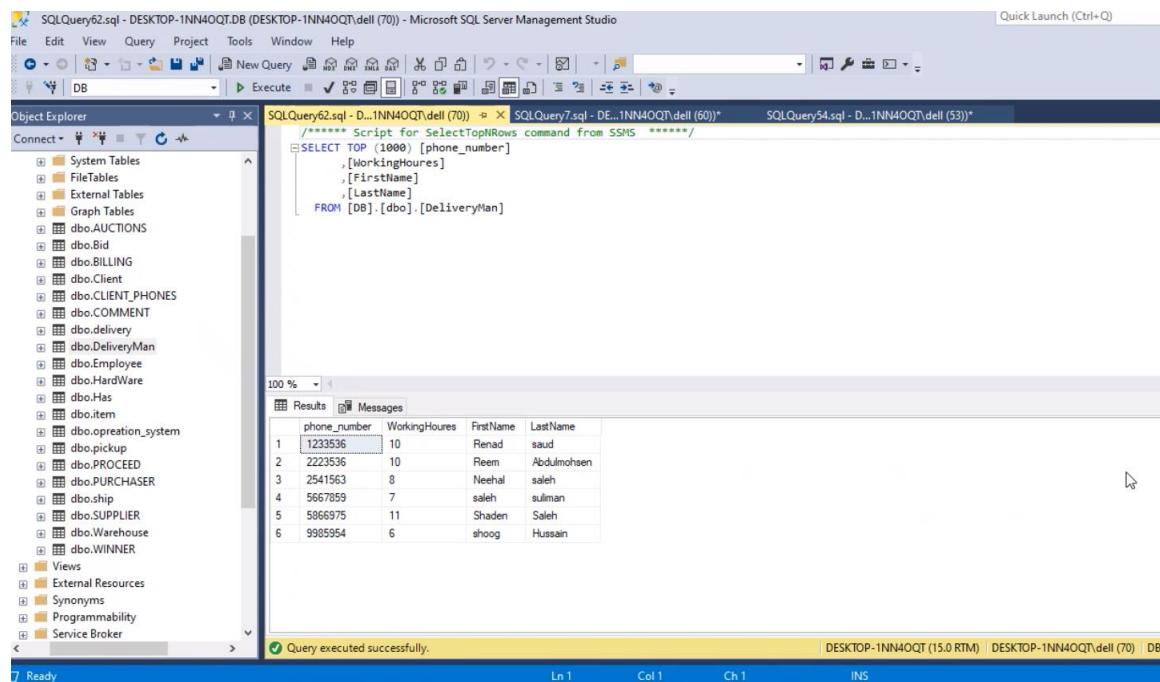
```

The results pane displays a table with four columns: BILLING\_ID, WIN\_ID, order\_Date, and Quantity. The data is as follows:

BILLING_ID	WIN_ID	order_Date	Quantity
1	1	2022-01-25	10
2	2	2022-02-28	15
3	3	2022-03-01	20
4	4	2022-01-25	15
5	5	2022-04-23	13
6	6	2022-05-12	11

A status bar at the bottom indicates "Query executed successfully."

- INSERTION OF TABLE DeleveryMan:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, External Tables, and Tables. The central pane displays a T-SQL script:

```

/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [phone_number]
,[WorkingHours]
,[FirstName]
,[LastName]
FROM [DB].[dbo].[DeliveryMan]

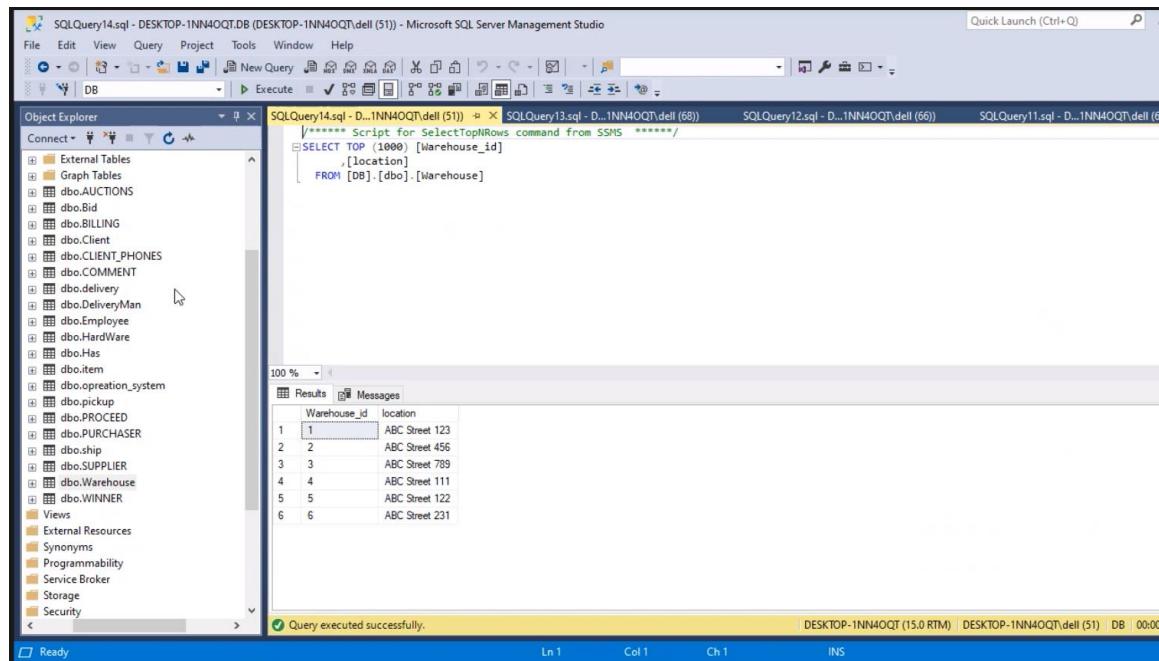
```

The Results pane shows the output of the query, which is a table with columns: phone\_number, WorkingHours, FirstName, and LastName. The data is as follows:

phone_number	WorkingHours	FirstName	LastName
123536	10	Renad	saud
2223536	10	Reem	Abdulmohsen
2541563	8	Neehal	saleh
5657859	7	saleh	sulman
5866975	11	Shaden	Saleh
9985954	6	shoog	Hussain

At the bottom, a message indicates "Query executed successfully."

- INSERTION OF TABLE Warehouse:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects. The central pane displays a T-SQL script:

```

/*
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [Warehouse_id]
,[location]
FROM [DB].[dbo].[Warehouse]

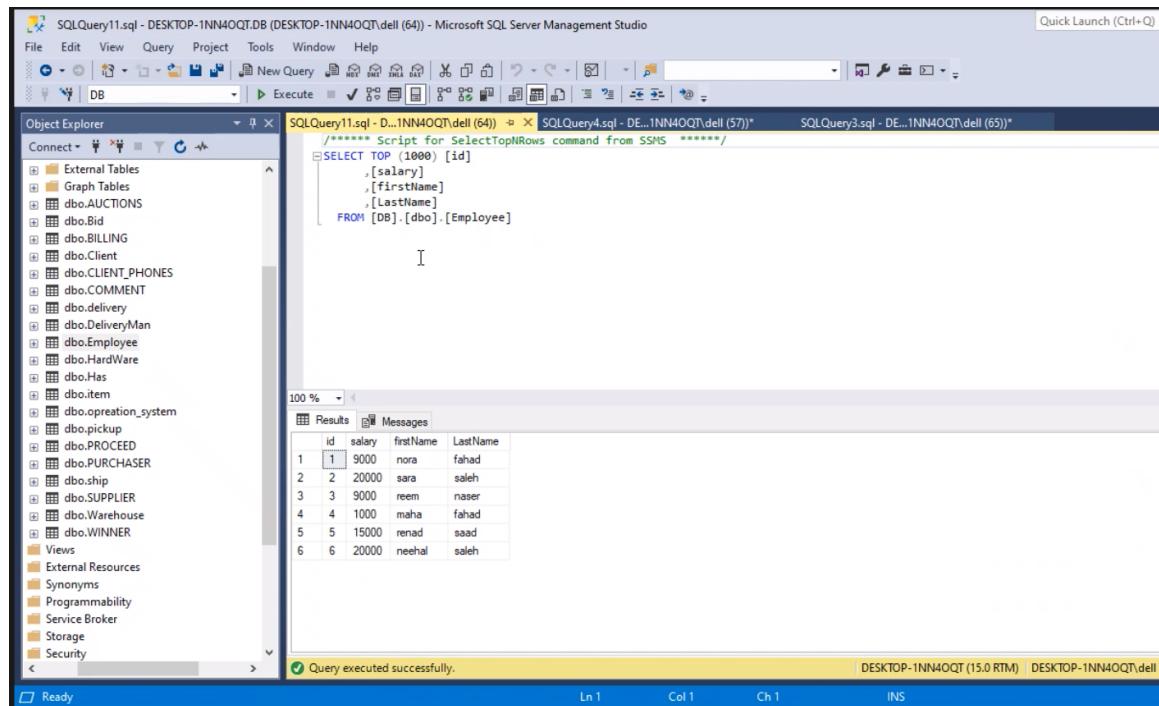
```

The Results pane shows the output of the query, which is a table with columns: Warehouse\_id and location. The data is as follows:

Warehouse_id	location
1	ABC Street 123
2	ABC Street 456
3	ABC Street 789
4	ABC Street 111
5	ABC Street 122
6	ABC Street 231

At the bottom, a message indicates "Query executed successfully."

- INSERTION OF TABLE Employee:**

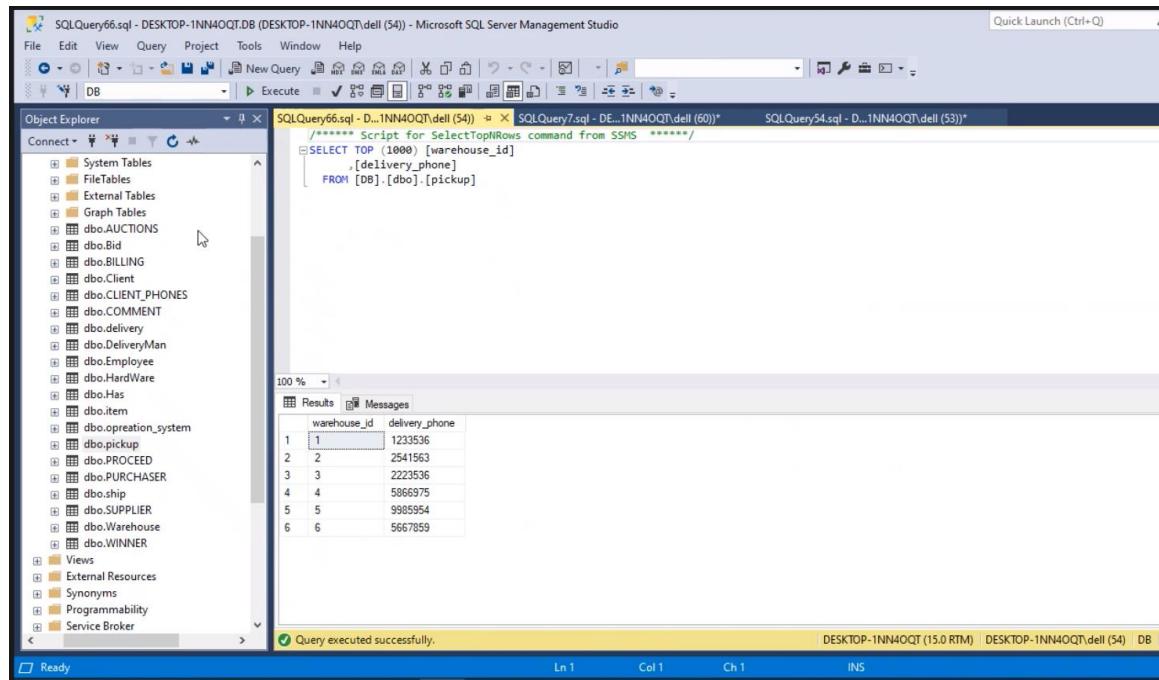


The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like External Tables, Graph Tables, and tables such as dbo.AUCTIONS, dbo.Bid, dbo.BILLING, dbo.Client, dbo.CLIENT\_PHONES, dbo.COMMENT, dbo.delivery, dbo.DeliveryMan, dbo.Employee, dbo.HardWare, dbo.Has, dbo.item, dbo.operation\_system, dbo.pickup, dbo.PROCEED, dbo.PURCHASER, dbo.ship, dbo.SUPPLIER, dbo.Warehouse, and dbo.WINNER. The central pane displays a T-SQL script for selecting top 1000 rows from the Employee table, which includes columns id, salary, firstName, and LastName. The results pane shows the following data:

	id	salary	firstName	LastName
1	1	9000	nora	fahad
2	2	20000	sara	saleh
3	3	9000	reem	naser
4	4	1000	maha	fahad
5	5	15000	renad	saad
6	6	20000	neehal	saleh

At the bottom, a message indicates "Query executed successfully."

- INSERTION OF TABLE Pickup:**

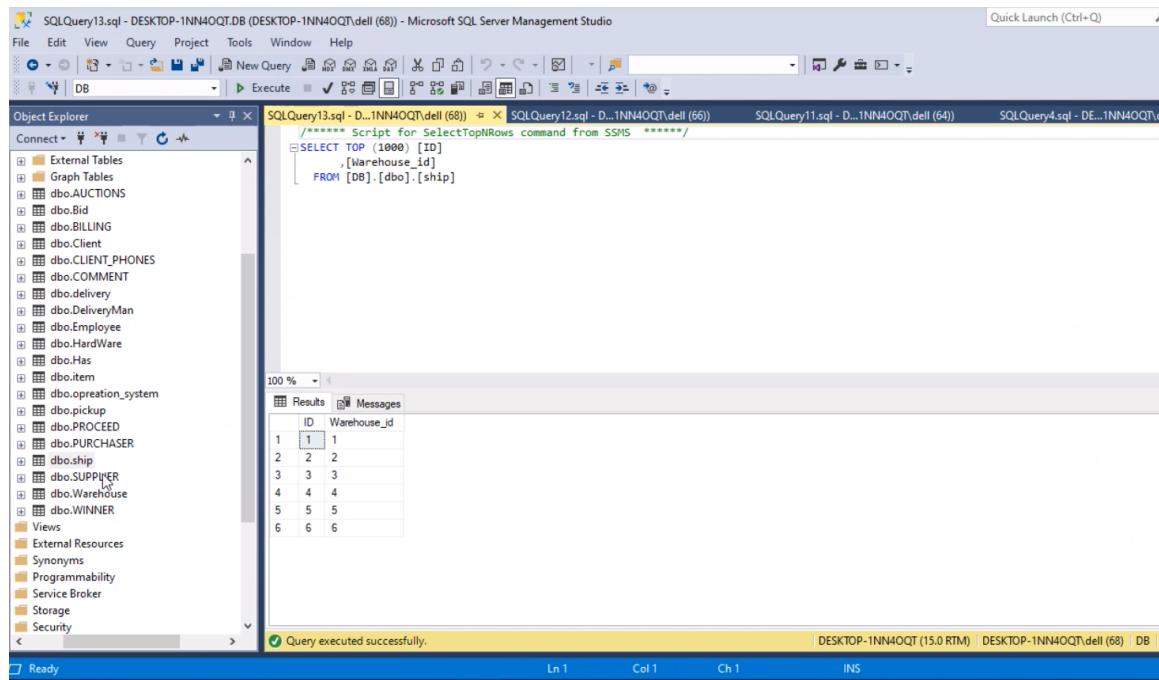


The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects. The central pane displays a T-SQL script for selecting top 1000 rows from the pickup table, which includes columns warehouse\_id and delivery\_phone. The results pane shows the following data:

	warehouse_id	delivery_phone
1	1	1233536
2	2	2541563
3	3	2223536
4	4	5866975
5	5	9985954
6	6	5667859

At the bottom, a message indicates "Query executed successfully."

- INSERTION OF TABLE Ship:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like External Tables, Graph Tables, and tables such as AUCTIONS, BILLING, Client, CLIENT\_PHONES, COMMENT, delivery, DeliveryMan, Employee, HardWare, Has, item, operation\_system, pickup, PROCCEED, PURCHASER, ship, SUPPLIER, Warehouse, and WINNER. The Results tab in the center displays the output of a query:

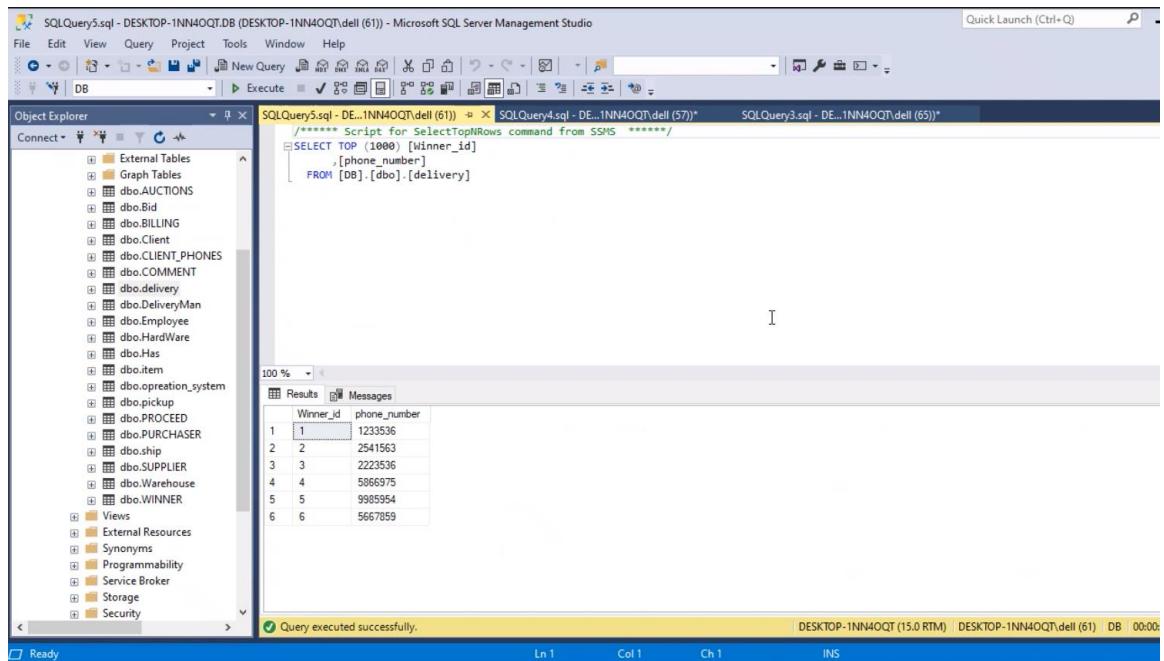
```
SELECT TOP (1000) [ID]
      ,[Warehouse_id]
  FROM [DB].[dbo].[Ship]
```

The results table shows the following data:

ID	Warehouse_id
1	1
2	2
3	3
4	4
5	5
6	6

A status bar at the bottom indicates "Query executed successfully." and shows the session details: DESKTOP-1NN4OQT (15.0 RTM) | DESKTOP-1NN4OQT\ dell (68) | DB | Ready.

- INSERTION OF TABLE delivery:**



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects. The Results tab in the center displays the output of a query:

```
SELECT TOP (1000) [Winner_id]
      ,[phone_number]
  FROM [DB].[dbo].[delivery]
```

The results table shows the following data:

Winner_id	phone_number
1	1233536
2	2541563
3	2223536
4	5866975
5	9985954
6	5667859

A status bar at the bottom indicates "Query executed successfully." and shows the session details: DESKTOP-1NN4OQT (15.0 RTM) | DESKTOP-1NN4OQT\ dell (61) | DB | 00:00 | Ready.

## Query Implementation:

### -View:

- Supplier\_items:** this view shows the names of supplier, and the number of items they have placed.

```
CREATE VIEW View_Supplier_Items
AS
SELECT (c.FIRST_NAME + ' ' + c.LAST_NAME) As 'Supplier Name', COUNT(i.item_id) 'No of Items'
FROM SUPPLIER s
INNER JOIN item i on i.Sup_Id = s.ID
INNER JOIN Client c on c.id = s.ID
GROUP BY c.FIRST_NAME + ' ' + c.LAST_NAME
```

```
CREATE VIEW View_Supplier_Items
AS
SELECT (c.FIRST_NAME + ' ' + c.LAST_NAME) As 'Supplier Name', COUNT(i.item_id) 'No of Items'
FROM SUPPLIER s
INNER JOIN item i on i.Sup_Id = s.ID
INNER JOIN Client c on c.id = s.ID
GROUP BY c.FIRST_NAME + ' ' + c.LAST_NAME
```

100 % Messages  
Commands completed successfully.  
Completion time: 2021-12-16T05:42:37.8288385+03:00

- Winner\_Details:** This view returns all of the winners' information as well as the item they won.

```
CREATE VIEW View_Winner_Details
AS
SELECT C.* , T.* FROM PURCHASER P
INNER JOIN Client C ON C.id = P.ID
INNER JOIN WINNER W ON W.ID = C.id
INNER JOIN Bid B ON B.pur_id = P.ID
INNER JOIN Has H ON H.Bid_id = B.Bid_id
INNER JOIN item T ON T.item_id = H.item_id
```

```
CREATE VIEW View_Winner_Details
AS
SELECT C.* , T.* FROM PURCHASER P
INNER JOIN Client C ON C.id = P.ID
INNER JOIN WINNER W ON W.ID = C.id
INNER JOIN Bid B ON B.pur_id = P.ID
INNER JOIN Has H ON H.Bid_id = B.Bid_id
INNER JOIN item T ON T.item_id = H.item_id
```

100 % Messages  
Msg 2714, Level 16, State 3, Procedure View\_Winner\_Details, Line 1 [Batch Start Line 0]  
There is already an object named 'View\_Winner\_Details' in the database.  
Completion time: 2021-12-16T05:43:43.5484387+03:00

**-Query:****1. Retrieve the number of bids and the highest bid made for every item.**

```
SELECT T.item_id, T.Title , COUNT(B.Bid_id) 'No of Bids' , Max(B.price) 'Highest Bid Price'
FROM Bid B
INNER JOIN Has H ON H.Bid_id = B.Bid_id
INNER JOIN item T ON T.item_id = H.item_id
GROUP BY T.item_id,T.Title
```

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database structure for 'DESKTOP-5F0D00B (SQL Server 15.0...)'. In the center is the SQL Query window titled 'SQLQuery17.sql - ...SF0D00B\USER (61)\*'. The query is displayed in the text area:

```
SELECT T.item_id, T.Title , COUNT(B.Bid_id) 'No of Bids' , Max(B.price) 'Highest Bid Price'
FROM Bid B
INNER JOIN Has H ON H.Bid_id = B.Bid_id
INNER JOIN item T ON T.item_id = H.item_id
GROUP BY T.item_id,T.Title
```

Below the query is the results grid, which displays the following data:

item_id	Title	No of Bids	Highest Bid Price
1	COMPUTER	2	750
2	RAM	1	600
3	HARD DRIVE	1	600
4	LAPTOP	1	630
5	LAPTOP	1	630

A status bar at the bottom indicates 'Query executed successfully.' and shows the session details: DESKTOP-5F0D00B (15.0 RTM) | DESKTOP-5.

**2. Retrieve the name of every supplier, and the number of items placed by him/her.**

```
SELECT * FROM View_Supplier_Items
```

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database structure for 'DESKTOP-5F0D00B (SQL Server 15.0...)'. In the center is the SQL Query window titled 'SQLQuery17.sql - ...SF0D00B\USER (61)\*'. The query is displayed in the text area:

```
SELECT * FROM View_Supplier_Items
```

Below the query is the results grid, which displays the following data:

Supplier Name	No of Items
Naehal Ahamed	2
Naehal Naseem	2
shaden mohammed	2

A status bar at the bottom indicates 'Query executed successfully.' and shows the session details: DESKTOP-5F0D00B (15.0 RTM) | DESK.

### 3. Retrieve the names, the bids of the highest bidding purchasers.

```
SELECT (c.FIRST_NAME + '' + c.LAST_NAME) As 'Supplier Name' , MAX(B.price) 'Highest Bid' FROM Bid B
INNER JOIN SUPPLIER S ON S.ID = B.Bid_id
INNER JOIN Client C ON C.id = S.ID
GROUP BY (c.FIRST_NAME + '' + c.LAST_NAME)
```

The screenshot shows the Object Explorer on the left with the database 'DESKTOP-5F0D00B SQL Server 15.0' selected. The 'Tables' node under 'DB' is expanded, showing various tables like AUCTIONS, BILLING, CLIENT\_PHONES, etc. The 'Results' tab in the center displays the output of the query:

Supplier Name	Highest Bid
maha Nasir	630
Naeah Ahamd	650
Naeah Naseem	600
reem Hussain	550
saleh sulman	5850
shaden mohammed	750

At the bottom, a message says 'Query executed successfully.'

### 4. List all software that have memory between 8GB and 16GB.

```
SELECT * FROM item I
INNER JOIN HardWare H ON I.item_id = H.item_id
WHERE H.Memory BETWEEN 8 AND 16
```

The screenshot shows the Object Explorer on the left with the database 'DESKTOP-5F0D00B SQL Server 15.0' selected. The 'Tables' node under 'DB' is expanded, showing various tables like AUCTIONS, BILLING, CLIENT\_PHONES, etc. The 'Results' tab in the center displays the output of the query:

item_id	Bid_Price	Title	Description	Sup_Id	item_id	Storage	Speed	Memory
1	100	COMPUTER		1	1	2000	2	8
2	50	RAM		1	2	4000	2	16
3	45	LAPTOP		3	4	6000	2	12
4	66	COMPUTER		3	5	2000	2	8

At the bottom, a message says 'Query executed successfully.'

**5. List all software that have are manufactured in 2020 and still in auction.**

```
SELECT I.item_id , I.Title,OS.Author,OS.cost,OS.Manufacturer,OS.year_of_manufacturing
FROM item I
INNER JOIN operation_system OS ON I.item_id = OS.item_id
WHERE YEAR( OS.year_of_manufacturing ) = 2020
```

The screenshot shows the Object Explorer on the left with the database structure. The central pane displays the query:

```
SELECT I.item_id , I.Title,OS.Author,OS.cost,OS.Manufacturer,OS.year_of_manufacturing
FROM item I
INNER JOIN operation_system OS ON I.item_id = OS.item_id
WHERE YEAR( OS.year_of_manufacturing ) = 2020
```

The results pane shows a single row of data:

item_id	Title	Author	cost	Manufacturer	year_of_manufacturing
1	COMPUTER	Microsoft	500	Microsoft	2020-01-01

A status bar at the bottom indicates "Query executed successfully."

**6. For each item, retrieve the number of bids, and the average price.**

```
SELECT I.Title,COUNT(B.Bid_id) 'No of Bids',AVG(B.price) 'Average Price'
FROM item I
INNER JOIN Has H ON H.item_id = I.item_id
INNER JOIN Bid B ON B.Bid_id = H.Bid_id
GROUP BY I.Title
```

The screenshot shows the Object Explorer on the left with the database structure. The central pane displays the query:

```
SELECT I.Title,COUNT(B.Bid_id) 'No of Bids',AVG(B.price) 'Average Price'
FROM item I
INNER JOIN Has H ON H.item_id = I.item_id
INNER JOIN Bid B ON B.Bid_id = H.Bid_id
GROUP BY I.Title
```

The results pane shows the following data:

Title	No of Bids	Average Price
COMPUTER	2	675.000000
LAPTOP	1	630.000000
HARD DRIVE	1	600.000000
LAPTOP	1	630.000000
RAM	1	600.000000

A status bar at the bottom indicates "Query executed successfully."

## 7. Find all items that has no bids yet.

```
SELECT I.item_id , I.Title
FROM item I
LEFT JOIN Has H ON H.item_id = I.item_id
LEFT JOIN Bid B ON B.Bid_id = H.Bid_id
WHERE B.price IS NULL
```

The screenshot shows the Object Explorer on the left with the database structure. The SQL Query window contains the following code:

```
SELECT I.item_id , I.Title
FROM item I
LEFT JOIN Has H ON H.item_id = I.item_id
LEFT JOIN Bid B ON B.Bid_id = H.Bid_id
WHERE B.price IS NULL
```

The Results pane shows a single row of data:

item_id	Title
1	COMPUTER

The status bar at the bottom indicates "Query executed successfully."

## 8. For each auction on which more than two bids opened, retrieve the auction description, project expiration, and the number of bids that open in that auction.

```
SELECT A.Description,A.Expiration,COUNT(B.Bid_id) 'Count of Bids'
FROM AUCTIONS A
INNER JOIN Bid B ON B.Auction_id = A.AUCTION_ID
GROUP BY A.Description,A.Expiration
HAVING COUNT(B.Bid_id) >2
```

The screenshot shows the Object Explorer on the left with the database structure. The SQL Query window contains the following code:

```
SELECT A.Description,A.Expiration,COUNT(B.Bid_id) 'Count of Bids'
FROM AUCTIONS A
INNER JOIN Bid B ON B.Auction_id = A.AUCTION_ID
GROUP BY A.Description,A.Expiration
HAVING COUNT(B.Bid_id) >2
```

The Results pane shows a table with three columns: Description, Expiration, and Count of Bids. The count of bids is currently blank.

Description	Expiration	Count of Bids

The status bar at the bottom indicates "Query executed successfully."

**9. List all purchasers or suppliers who recorded a comment with 3 and above as transaction rating.**

```
SELECT (CL.FIRST_NAME + '' + CL.LAST_NAME) As 'Supplier Name' , C.RATING FROM PURCHASER P
INNER JOIN COMMENT C ON C.ID = P.ID
INNER JOIN Client CL ON CL.ID = P.ID
WHERE C.RATING >=3
```

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database structure including tables like PURCHASER, COMMENT, and Client. The main area is a query window titled 'SQLQuery17.sql - DESKTOP-5F0D00B\USER (61)\*'. The query is:

```
=SELECT (CL.FIRST_NAME + '' + CL.LAST_NAME) As 'Supplier Name' , C.RATING FROM PURCHASER P
INNER JOIN COMMENT C ON C.ID = P.ID
INNER JOIN Client CL ON CL.ID = P.ID
WHERE C.RATING >=3
```

The results pane shows a table with two columns: 'Supplier Name' and 'RATING'. The data is:

Supplier Name	RATING
Naethal Naseem	4
Naethal Ahemd	3
shaden mohammed	5
reem Hussan	4
saleh sulman	6

At the bottom of the interface, it says 'Query executed successfully.'

**10. List all purchasers with a shipping address in Saudi Arabia**

```
SELECT * FROM BILLING B
INNER JOIN WINNER W ON W.ID = B.WIN_ID
INNER JOIN Client C ON C.id = W.ID
WHERE C.CLINT_ADDRESS LIKE '% ' + 'Saudi Arabia'
```

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database structure including tables like BILLING, WINNER, and Client. The main area is a query window titled 'SQLQuery17.sql - DESKTOP-5F0D00B\USER (61)\*'. The query is:

```
=SELECT * FROM BILLING B
INNER JOIN WINNER W ON W.ID = B.WIN_ID
INNER JOIN Client C ON C.id = W.ID
WHERE C.CLINT_ADDRESS LIKE '% ' + 'Saudi Arabia'
```

The results pane shows a table with ten columns: BILLING\_ID, WIN\_ID, order\_Date, Quantity, ID, EMAIL, PASSWORD, FIRST\_NAME, LAST\_NAME, and CLINT\_ADDRESS. The data is:

BILLING_ID	WIN_ID	order_Date	Quantity	ID	EMAIL	PASSWORD	FIRST_NAME	LAST_NAME	CLINT_ADDRESS
1	3	2022-03-01	20	3	reem@gmail.com	111	shaden	mohammed	XYZ Street Saudi Arabia
2	5	2022-04-22	13	5	naethal@gmail.com	133	reem	Hussan	ABC Street Saudi Arabia
3	6	2022-05-23	11	6	saleh@gmail.com	134	saleh	sulman	ABC Street Saudi Arabia

At the bottom of the interface, it says 'Query executed successfully.'