

Assignment 4

Courrier Management System

Task1:

```
1  (* Assignment 4 *)
2  * CREATE DATABASE CourierManagementSystem;
3  * USE CourierManagementSystem;
4
5  * CREATE TABLE Users (
6      UserID INT PRIMARY KEY,
7      Name VARCHAR(255),
8      Email VARCHAR(255) UNIQUE,
9      Password VARCHAR(255),
10     ContactNumber VARCHAR(10),
11     Address TEXT
12 )
13
14 * CREATE TABLE Couriers (
15     CourierID INT PRIMARY KEY,
16     SenderName VARCHAR(255),
17     SenderAddress TEXT,
18     ReceiverName VARCHAR(255),
19     ReceiverAddress TEXT,
20     Weight DECIMAL(7, 3),
21     Status VARCHAR(50),
22     TrackingNumber VARCHAR(255) UNIQUE,
23     DeliveryDate DATE
24 )
25
26 * CREATE TABLE CourierServices (
27     ServiceID INT PRIMARY KEY,
28     ServiceName VARCHAR(100),
29     Cost DECIMAL(11, 3)
30 )
```

```
31 * CREATE TABLE Employees (
32     EmployeeID INT PRIMARY KEY,
33     Name VARCHAR(255),
34     Email VARCHAR(255) UNIQUE,
35     ContactNumber VARCHAR(10),
36     Role VARCHAR(50),
37     Salary DECIMAL(10, 3)
38 )
39
40 * CREATE TABLE Locations (
41     LocationID INT PRIMARY KEY,
42     LocationName VARCHAR(100),
43     Address TEXT
44 )
45
46 * CREATE TABLE Payments (
47     PaymentID INT PRIMARY KEY,
48     CourierID INT,
49     LocationID INT,
50     Amount DECIMAL(10, 3),
51     PaymentDate DATE,
52     FOREIGN KEY (CourierID) REFERENCES Couriers(CourierID),
53     FOREIGN KEY (LocationID) REFERENCES Locations(LocationID)
54 )
```

Data Insertion:

```
Assignment
SQL Address
My Scripts

100 * INSERT INTO Users (UserID, Name, Email, Password, ContactNumber, Address)
101 VALUES
102 (1, 'Raj Sharma', 'raj.sharma@gmail.com', 'password123', '9876543210', '123 Main Street, Delhi'),
103 (2, 'Priya Patel', 'priya.patel@gmail.com', 'priya456', '0987654321', '456 Oak Avenue, Mumbai'),
104 (3, 'Amit Kumar', 'amit.kumar@gmail.com', 'amit789', '2109876543', '789 Elm Street, Bangalore'),
105 (4, 'Neha Singh', 'neha.singh@gmail.com', 'neha101', '3456789012', '101 Pine Avenue, Hyderabad'),
106 (5, 'Vikram Reddy', 'vikram.reddy@gmail.com', 'vikram111', '4567890123', '202 Oak Street, Kolkata'),
107 (6, 'Smita Das', 'smita.das@gmail.com', 'smita222', '5678901234', '303 Maple Avenue, Chennai'),
108 (7, 'Sandeep Gupta', 'sandeep.gupta@gmail.com', 'sandeep333', '6789012345', '404 Elm Street, Pune'),
109 (8, 'Meera Iyer', 'meera.iyer@gmail.com', 'meera444', '7890123456', '505 Pine Avenue, Ahmedabad')
110
111 * INSERT INTO Couriers (CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliveryDate)
112 VALUES
113 (1, 'Raj Sharma', '789 Elm Street, Delhi', 'Neha Singh', '101 Elm Street, Mumbai', 5.5, 'In Transit', 'TRK001', '2023-04-01'),
114 (2, 'Priya Patel', '123 Oak Street, Kolkata', 'Smita Das', '456 Maple Street, Bangalore', 8.0, 'Delivered', 'TRK002', '2023-03-15'),
115 (3, 'Amit Kumar', '101 Elm Street, Hyderabad', 'Sandeep Gupta', '222 Oak Street, Chennai', 7.0, 'In Transit', 'TRK003', '2023-04-05'),
116 (4, 'Neha Singh', '404 Maple Avenue, Pune', 'Sandeep Gupta', '505 Elm Street, Ahmedabad', 4.0, 'Delivered', 'TRK004', '2023-04-01'),
117 (5, 'Vikram Reddy', '777 Elm Street, Delhi', 'Vikram Reddy', '888 Maple Avenue, Mumbai', 6.0, 'In Transit', 'TRK005', '2023-04-03'),
118 (6, 'Smita Das', '999 Elm Street, Kolkata', 'Meera Iyer', '111 Elm Street, Bangalore', 4.0, 'In Transit', 'TRK006', '2023-04-02'),
119 (7, 'Raj Sharma', '202 Oak Street, Hyderabad', 'Sandeep Gupta', '404 Maple Avenue, Pune', 5.0, 'Delivered', 'TRK007', '2023-03-28'),
120 (8, 'Sandeep Gupta', '303 Elm Street, Chennai', 'Vikram Reddy', '505 Elm Street, Ahmedabad', 3.0, 'In Transit', 'TRK008', '2023-04-11')
121
122 * INSERT INTO CourierServices (ServiceID, ServiceName, Cost)
123 VALUES
124 (1, 'Delivery Driver', 20.00),
125 (2, 'Express', 15.00),
126 (3, 'Next Day Delivery', 30.00),
127 (4, 'Two Day Delivery', 18.00),
128 (5, 'Storage', 12.00),
129 (6, 'Same Day Delivery', 25.00)
```

```
Assignment
SQL Address
My Scripts

130 * INSERT INTO Employees (EmployeeID, Name, Email, ContactNumber, Role, Salary)
131 VALUES
132 (1, 'Rohan Gupta', 'rohan.gupta@gmail.com', '111-222-3333', 'Delivery Driver', 50000.00),
133 (2, 'Smita Das', 'smita.das@gmail.com', '444-555-6666', 'Customer Service', 40000.00),
134 (3, 'Vikram Singh', 'vikram.singh@gmail.com', '777-888-9999', 'Warehouse Manager', 60000.00),
135 (4, 'Priya Reddy', 'priya.reddy@gmail.com', '000-111-2222', 'Dispatcher', 45000.00),
136 (5, 'Vikram Reddy', 'vikram.reddy@gmail.com', '111-222-3333', 'Delivery Driver', 55000.00),
137 (6, 'Smita Das', 'smita.das@gmail.com', '333-444-5555', 'Customer Service', 47000.00),
138 (7, 'Raj Sharma', 'raj.sharma@gmail.com', '555-666-7777', 'Warehouse Staff', 35000.00),
139 (8, 'Meera Iyer', 'meera.iyer@gmail.com', '777-888-9999', 'Dispatcher', 48000.00)
140
141 * INSERT INTO Locations (LocationID, LocationName, Address)
142 VALUES
143 (1, 'Main Office', '123 Main Street, Delhi'),
144 (2, 'Branch Office', '456 Oak Avenue, Mumbai'),
145 (3, 'Warehouse A', '777 Elm Street, Hyderabad'),
146 (4, 'Warehouse B', '888 Elm Street, Pune'),
147 (5, 'Branch Office 2', '999 Elm Street, Bangalore'),
148 (6, 'Branch Office 3', '101 Maple Avenue, Chennai'),
149 (7, 'Warehouse C', '111 Oak Street, Kolkata'),
150 (8, 'Branch Office 4', '222 Elm Street, Ahmedabad')
151
152 * INSERT INTO Payments (PaymentID, CourierID, LocationID, Amount, PaymentDate)
153 VALUES
154 (1, 1, 1, 10.00, '2023-04-01'),
155 (2, 2, 2, 15.00, '2023-04-01'),
156 (3, 3, 3, 12.00, '2023-04-01'),
157 (4, 4, 4, 18.00, '2023-04-01'),
158 (5, 5, 5, 10.00, '2023-04-01'),
159 (6, 6, 6, 12.00, '2023-04-01')
```

Output			
Action Output			
#	Time	Action	Message
1	14:10:32	CREATE DATABASE CourierManagementSystem	Row(s) affected
2	14:10:34	USE CourierManagementSystem	Row(s) affected
3	14:10:30	CREATE TABLE Users (UserID INT PRIMARY KEY, Name VARCHAR(255), Email VARCHAR(255) UNI...	Row(s) affected
4	14:10:41	CREATE TABLE Couriers (CourierID INT PRIMARY KEY, SenderName VARCHAR(255), SenderAddress ...	Row(s) affected
5	14:10:44	CREATE TABLE CourierServices (ServiceID INT PRIMARY KEY, ServiceName VARCHAR(100), Cost D...	Row(s) affected
6	14:10:52	CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, Name VARCHAR(255), Email VARCHAR...	Row(s) affected
7	14:10:54	CREATE TABLE Locations (LocationID INT PRIMARY KEY, LocationName VARCHAR(100), Address T...	Row(s) affected
8	14:11:00	CREATE TABLE Payments (PaymentID INT PRIMARY KEY, CourierID INT, LocationID INT, Amount ...	Row(s) affected
9	14:11:08	INSERT INTO Users (UserID, Name, Email, Password, ContactNumber, Address) VALUES (1, 'Raj Sharma', 'raj.sh...	Row(s) affected Records: 8 Duplicates: 0 Warnings: 0
10	14:11:10	INSERT INTO Couriers (CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Stat...	Row(s) affected Records: 8 Duplicates: 0 Warnings: 0
11	14:11:17	INSERT INTO CourierServices (ServiceID, ServiceName, Cost) VALUES (1, 'Delivery Driver', 20.00), (2, 'Express...	Row(s) affected Records: 8 Duplicates: 0 Warnings: 0
12	14:11:20	INSERT INTO Employees (EmployeeID, Name, Email, ContactNumber, Role, Salary) VALUES (1, 'Rohan Gupta', ...	Row(s) affected Records: 8 Duplicates: 0 Warnings: 0
13	14:11:23	INSERT INTO Locations (LocationID, LocationName, Address) VALUES (1, 'Main Office', '123 Main Street, Delhi...	Row(s) affected Records: 8 Duplicates: 0 Warnings: 0
14	14:11:28	INSERT INTO Payments (PaymentID, CourierID, LocationID, Amount, PaymentDate) VALUES (1, 1, 1, 20.00, '20...	Row(s) affected Records: 8 Duplicates: 0 Warnings: 0

Task 2:

1.

The screenshot shows the SQL Studio interface. The query editor contains the following SQL code:

```
126 /*Task 2*/
127
128 /*1*/
129 * SELECT * FROM Users;
130
```

The result grid displays 8 rows of user data:

UserID	Name	Email	Password	ContactNumber	Address
1	Raj Sharma	raj.sharma@email.com	password123	9876543210	123 Main Street, Delhi
2	Priya Patel	priya.patel@email.com	pass456	7890123456	456 Oak Avenue, Mumbai
3	Amit Verma	amit.verma@email.com	amit123	5678901234	789 Elm Street, Bangalore
4	Neha Reddy	neha.reddy@email.com	neha456	7654321098	101 Pine Avenue, Hyderabad
5	Rahul Kapoor	rahul.kapoor@email.com	rahul567	6543210987	202 Oak Street, Kolkata
6	Anjali Desai	anjali.desai@email.com	anjali456	5432109876	303 Maple Avenue, Chennai
7	Sanjay Mehra	sanjay.mehra@email.com	sanjay789	4321098765	404 Elm Street, Pune
8	Neha Iyer	neha.iyer@email.com	neha987	3210987654	505 Pine Avenue, Ahmedabad

The output pane shows the execution details:

Time	Action	Message	Duration / Fetch
1 14:25:17	SELECT * FROM Users LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

2.

The screenshot shows the SQL Studio interface. The query editor contains the following SQL code:

```
132 /*2*/
133 * SELECT * FROM Couriers
134 WHERE SenderName = (SELECT Name FROM Users WHERE UserID = 2) OR
135 ReceiverName = (SELECT Name FROM Users WHERE UserID = 2);
136
```

The result grid displays 1 row of courier data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
2	Priya Patel	123 Oak Street, Kolkata	Amit Gupta	456 Maple Street, Bangalore	5.00	Delivered	TR789012	2023-07-28

The output pane shows the execution details:

Time	Action	Message	Duration / Fetch
1 14:27:20	SELECT * FROM Couriers WHERE SenderName = (SELECT Name FROM Users WHERE UserID = 2) OR Rec...	1 row(s) returned	0.000 sec / 0.000 sec

3.

The screenshot shows a SQL IDE interface with a query editor at the top containing the following SQL code:

```

136
137
138 /*3*/
139 * SELECT * FROM Couriers;
140

```

Below the query editor, the 'Result Grid' displays the results of the query. The table has 10 columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, and DeliveryDate. There are 10 rows of data.

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Raj Sharma	785 Pine Street, Delhi	Suresh Kumar	367 Elm Street, Mumbai	8.50	In Transit	TH123456	2023-04-01
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered	TH789012	2023-03-28
3	Jack Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TH654321	2023-04-05
4	Anu Verma	505 Maple Avenue, Pune	Shweta Reddy	600 Elm Street, Ahmedabad	4.50	Delivered	TH987654	2023-04-02
5	Kavita Patel	777 Oak Street, Delhi	Vedat Iyer	888 Maple Avenue, Mumbai	9.20	In Transit	TH109902	2023-04-08
6	Rohit Deshmukh	999 Pine Street, Kolkata	Smrithi Kapoor	111 Elm Street, Bangalore	6.80	Scheduled	TH345678	2023-04-10
7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehta	404 Maple Avenue, Pune	5.30	Delivered	TH012345	2023-03-30
8	Sangeeta Iyer	202 Elm Street, Chennai	Vivek Desai	303 Pine Street, Ahmedabad	8.00	In Transit	TH678901	2023-04-12
9								
10								

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Feat
1	14:27:50	SELECT * FROM Couriers LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

4.

The screenshot shows the same SQL IDE interface, but with a modified query in the query editor:

```

140
141 /*4*/
142 * SELECT * FROM Couriers WHERE CourierID = 3;
143
144

```

The 'Result Grid' now displays only one row of data, corresponding to CourierID 3.

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
3	Jack Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TH654321	2023-04-05

The 'Output' pane shows the execution log with two entries:

#	Time	Action	Message	Duration / Feat
1	14:27:50	SELECT * FROM Couriers LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
2	14:28:14	SELECT * FROM Couriers WHERE CourierID = 3 LIMIT 0, 1000	1 row(s) returned	0.019 sec / 0.000 sec

5.

143
144
145 /*5*/
146 * SELECT * FROM Couriers WHERE CourierID = 1;
147

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit	TN123456	2023-04-01

Couriers 5 x

Output

Time	Action	Message	Duration / Refresh
1 14:27:50	SELECT * FROM Couriers LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
2 14:28:14	SELECT * FROM Couriers WHERE CourierID = 1 LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
3 14:28:28	SELECT * FROM Couriers WHERE CourierID = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

6.

147
148 /*6*/
149 * SELECT * FROM Couriers WHERE Status != 'Delivered';
150
151

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit	TN123456	2023-04-01
2	Yash Singh	111 Pine Street, Hyderabad	Ashu Gupta	222 Oak Street, Chennai	7.80	In Transit	TN654321	2023-04-05
3	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.20	In Transit	TN789012	2023-04-08
4	Rohit Deshmukh	999 Pine Street, Kolkata	Griha Kapoor	111 Elm Street, Bangalore	6.80	Scheduled	TN345678	2023-04-10
5	Sangeeta Iyer	202 Elm Street, Chennai	Vivek Desai	303 Pine Street, Ahmedabad	8.00	In Transit	TN678901	2023-04-12

Couriers 6 x

Output

Time	Action	Message	Duration / Refresh
1 14:29:03	SELECT * FROM Couriers WHERE Status != 'Delivered' LIMIT 0, 1000	5 row(s) returned	0.062 sec / 0.000 sec

7.

The screenshot shows a database IDE with a SQL query editor at the top and a results pane at the bottom. The query is:

```
150
151
152 /*7*/
153 * SELECT * FROM Couriers WHERE DeliveryDate = CURDATE();
154
```

The results pane shows the execution output:

Time	Action	Message	Duration / Refresh
14:29:33	SELECT * FROM Couriers WHERE Status != 'Delivered' LIMIT 0, 1000	5 row(s) returned	0.062 sec / 0.000 sec
14:29:32	SELECT * FROM Couriers WHERE DeliveryDate = CURDATE() LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

8.

The screenshot shows a database IDE with a SQL query editor at the top and a results pane at the bottom. The query is:

```
154
155 /*8*/
156 * SELECT * FROM Couriers WHERE Status = 'In Transit';
157
158
```

The results pane shows the execution output:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Raj Sharma	399 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit	TH123456	2023-04-01
3	Yash Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TH654321	2023-04-05
5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.20	In Transit	TH987654	2023-04-09
8	Sangeeta Iyer	202 Elm Street, Chennai	Vivek Desai	303 Pine Street, Ahmedabad	6.00	In Transit	TH456789	2023-04-12

The results pane also shows the execution output:

Time	Action	Message	Duration / Refresh
14:29:56	SELECT * FROM Couriers WHERE Status = 'In Transit' LIMIT 0, 1000	4 row(s) returned	0.076 sec / 0.000 sec

9.

The screenshot shows a SQL query editor with the following query:

```
159 /*9*/
160 * SELECT CourierID, COUNT(*) AS TotalPackages
161 FROM Couriers
162 GROUP BY CourierID;
163
```

The query is executed, and the results are displayed in a table:

CourierID	TotalPackages
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1

The bottom of the screenshot shows the 'Output' pane with the following message:

```
1 14:30:41 SELECT CourierID, COUNT(*) AS TotalPackages FROM Couriers GROUP BY CourierID LIMIT 0, 1000
8 row(s) returned
0.000 sec / 0.000 sec
```

10.

The screenshot shows a SQL query editor with the following query:

```
164 /*10*/
165 * SELECT CourierID, AVG(DATEDIFF('2023-05-01', DeliveryDate)) AS AverageDeliveryTime
166 FROM Couriers
167 GROUP BY CourierID;
168
```

The query is executed, and the results are displayed in a table:

CourierID	AverageDeliveryTime
1	30.0000
2	34.0000
3	26.0000
4	29.0000
5	23.0000
6	21.0000
7	32.0000
8	19.0000

The bottom of the screenshot shows the 'Output' pane with the following message:

```
1 14:32:21 SELECT CourierID, AVG(DATEDIFF('2023-05-01', DeliveryDate)) AS AverageDeliveryTime FROM Couriers GRO...
8 row(s) returned
0.000 sec / 0.000 sec
```


11.

168
169
170 /*11*/
171 * SELECT * FROM Couriers WHERE Weight BETWEEN 5.0 AND 10.0;
172

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TradingNumber	DeliveryDate
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit	78123456	2023-04-01
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered	76789012	2023-03-28
3	Yash Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	78012345	2023-04-05
4	Kavita Patel	777 Oak Street, Delhi	Vinod Tyer	888 Maple Avenue, Mumbai	9.20	In Transit	76789002	2023-04-08
5	Rohit Deshmukh	999 Pine Street, Kolkata	Smriti Kapoor	111 Elm Street, Bangalore	6.80	Scheduled	78123457	2023-04-10
7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Avenue, Pune	5.50	Delivered	78012340	2023-03-30
8	Sangam Tyer	202 Elm Street, Chennai	Vinod Desai	303 Pine Street, Ahmedabad	8.00	In Transit	78078901	2023-04-12

Couriers 11 x

Output

Time	Action	Message	Duration / Fetch
14:32:21	SELECT CourierID, AVG(DATEDIFF(2023-05-01, DeliveryDate)) AS AverageDeliveryTime FROM Couriers GRO...	8 row(s) returned	0.000 sec / 0.000 sec
14:33:08	SELECT * FROM Couriers WHERE Weight BETWEEN 5.0 AND 10.0 LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

12.

172
173 /*12*/
174 * SELECT * FROM Employees WHERE Name LIKE "%John%";
175
176

EmployeeID	Name	Email	ContactNumber	Role	Salary
1	John Doe	john.doe@company.com	9876543210	Software Engineer	120000
2	John Smith	john.smith@company.com	8765432109	Product Manager	150000
3	John Johnson	john.johnson@company.com	7654321098	Marketing Specialist	90000

Employees 12 x

Output

Time	Action	Message	Duration / Fetch
14:32:21	SELECT CourierID, AVG(DATEDIFF(2023-05-01, DeliveryDate)) AS AverageDeliveryTime FROM Couriers GRO...	8 row(s) returned	0.000 sec / 0.000 sec
14:33:08	SELECT * FROM Couriers WHERE Weight BETWEEN 5.0 AND 10.0 LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
14:33:33	SELECT * FROM Employees WHERE Name LIKE "%John%" LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

13.

The screenshot displays a database management interface. At the top, a SQL query is entered in a text area:

```
177 /*13*/  
178 * SELECT c.*, p.Amount  
179 FROM Couriers c  
180 JOIN Payments p ON c.CourierID = p.CourierID  
181 WHERE p.Amount > 50.00;
```

Below the query editor, a "Result Grid" is shown with a table structure. The table has the following columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliveryDate, and Amount. The grid is currently empty.

At the bottom, an "Output" pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	14:34:18	SELECT c.*, p.Amount FROM Couriers c JOIN Payments p ON c.CourierID = p.CourierID WHERE p.Amount > 50.00	6 row(s) returned	0.000 sec / 0.000 sec

Task 3:

14.

```
183 /* task 3*/
184
185 /*14*/ SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersHandled
186 FROM Employees e
187 LEFT JOIN Couriers c ON e.EmployeeID = c.CourierID
188 GROUP BY e.EmployeeID, e.Name;
189
```

EmployeeID	Name	TotalCouriersHandled
1	Rohan Gupta	1
2	Chitra Tyer	1
3	Vikram Singh	1
4	Pragya Desai	1
5	Kunal Kapoor	1
6	Aarti Singh	1
7	Rajat Mehra	1
8	Ananya Tyer	1

Result 14

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14.36.53	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersHandled FROM Employees e	8 rows returned	0.031 sec / 0.000 sec

15.

```
189
190 /*15*/ SELECT l.LocationName, SUM(p.Amount) AS TotalRevenue
191 FROM Locations l
192 LEFT JOIN Payments p ON l.LocationID = p.LocationID
193 GROUP BY l.LocationName;
194
195
```

LocationName	TotalRevenue
Main Office	30.00
Branch Office	30.00
Warehouse A	15.00
Warehouse B	25.00
Branch Office 2	18.00
Branch Office 3	22.00
Warehouse C	30.00
Branch Office 4	20.00

Result 15

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14.39.23	SELECT (LocationName, SUM(p.Amount) AS TotalRevenue FROM Locations l LEFT JOIN Payments p...	8 rows returned	0.060 sec / 0.000 sec

16.

The screenshot shows a SQL query editor with the following code:

```
194  
195  
196 * /*16*/ SELECT l.LocationName, COUNT(c.CourierID) AS TotalCouriersDelivered  
197 FROM Locations l  
198 LEFT JOIN Couriers c ON l.LocationID = c.CourierID  
199 GROUP BY l.LocationName;  
200
```

The query results are displayed in a table with two columns: LocationName and TotalCouriersDelivered.

LocationName	TotalCouriersDelivered
Main Office	1
Branch Office	1
Warehouse A	1
Warehouse B	1
Branch Office 2	1
Branch Office 3	1
Warehouse C	1
Branch Office 4	1

The bottom panel shows the execution log with two entries:

Time	Action	Message	Duration / Fetch
14:38:23	SELECT l.LocationName, SUM(p.Amount) AS TotalRevenue FROM Locations l LEFT JOIN Payments p ON l.LocationID = p.LocationID	8 row(s) returned	0.000 sec / 0.000 sec
14:38:53	SELECT l.LocationName, COUNT(c.CourierID) AS TotalCouriersDelivered FROM Locations l LEFT JOIN Couriers c ON l.LocationID = c.CourierID	8 row(s) returned	0.000 sec / 0.000 sec

17.

The screenshot shows a SQL query editor with the following code:

```
200  
201 * /*17*/ SELECT CourierID, AVG(DATEDIFF('2023-05-01', DeliveryDate)) AS AvgDeliveryTime  
202 FROM Couriers  
203 GROUP BY CourierID  
204 ORDER BY AvgDeliveryTime DESC  
205 LIMIT 1;  
206
```

The query results are displayed in a table with two columns: CourierID and AvgDeliveryTime.

CourierID	AvgDeliveryTime
2	34.0000

The bottom panel shows the execution log with one entry:

Time	Action	Message	Duration / Fetch
14:40:45	SELECT CourierID, AVG(DATEDIFF('2023-05-01', DeliveryDate)) AS AvgDeliveryTime FROM Couriers	1 row(s) returned	0.000 sec / 0.000 sec

18.

The screenshot shows the SQL Server Enterprise Manager interface. The query editor contains the following SQL code:

```

206
207 * /*18*/ SELECT l.LocationName, SUM(p.Amount) AS TotalPayments
208       FROM Locations l
209       LEFT JOIN Payments p ON l.LocationID = p.LocationID
210       GROUP BY l.LocationName
211       HAVING SUM(p.Amount) < 5000.00;
212

```

The Results pane displays the following data:

LocationName	TotalPayments
Main Office	20.00
Branch Office	20.00
Warehouse A	15.00
Warehouse B	25.00
Branch Office 2	18.00
Branch Office 3	22.00
Warehouse C	30.00
Branch Office 4	28.00

The Output pane shows the execution of the query:

#	Time	Action	Message	Duration / Fetch
1	14:40:45	SELECT CountID, AVG(DATEDIFF(2023-05-01, DeliveryDate)) AS AvgDeliveryTime FROM Counters	1 row(s) returned	0:000 sec / 0:000 sec
2	14:41:10	SELECT l.LocationName, SUM(p.Amount) AS TotalPayments FROM Locations l LEFT JOIN Payments p ON l.LocationID = p.LocationID GROUP BY l.LocationName HAVING SUM(p.Amount) < 5000.00	8 row(s) returned	0:000 sec / 0:000 sec

19.

The screenshot shows the SQL Server Enterprise Manager interface. The query editor contains the following SQL code:

```

212
213 * /*19*/ SELECT l.LocationName, SUM(p.Amount) AS TotalPayments
214       FROM Locations l
215       LEFT JOIN Payments p ON l.LocationID = p.LocationID
216       GROUP BY l.LocationName;
217
218

```

The Results pane displays the following data:

LocationName	TotalPayments
Main Office	20.00
Branch Office	20.00
Warehouse A	15.00
Warehouse B	25.00
Branch Office 2	18.00
Branch Office 3	22.00
Warehouse C	30.00
Branch Office 4	28.00

The Output pane shows the execution of the query:

#	Time	Action	Message	Duration / Fetch
1	14:41:40	SELECT l.LocationName, SUM(p.Amount) AS TotalPayments FROM Locations l LEFT JOIN Payments p ON l.LocationID = p.LocationID GROUP BY l.LocationName	8 row(s) returned	0:000 sec / 0:000 sec

20.

The screenshot shows the SQL Studio interface. The query editor contains the following SQL code:

```
219 * /*20*/ SELECT c.CourierID, c.ReceiverName, SUM(p.Amount) AS TotalPayments
220 FROM Couriers c
221 JOIN Payments p ON c.CourierID = p.CourierID
222 WHERE p.LocationID = 5
223 GROUP BY c.CourierID, c.ReceiverName
224 HAVING SUM(p.Amount) > 1000.00;
```

The Results pane shows a table with the following columns: CourierID, ReceiverName, TotalPayments.

The Output pane shows the following message:

```
1 14:43:43 SELECT c.CourierID, c.ReceiverName, SUM(p.Amount) AS TotalPayments FROM Couriers c JOIN Pay... 0 row(s) returned
```

21.

The screenshot shows the SQL Studio interface. The query editor contains the following SQL code:

```
226 * /*21*/ SELECT c.CourierID, c.ReceiverName, SUM(p.Amount) AS TotalPayments
227 FROM Couriers c
228 JOIN Payments p ON c.CourierID = p.CourierID
229 WHERE p.PaymentDate > '2023-05-01'
230 GROUP BY c.CourierID, c.ReceiverName
231 HAVING SUM(p.Amount) > 1000.00;
232
```

The Results pane shows a table with the following columns: CourierID, ReceiverName, TotalPayments.

The Output pane shows the following message:

```
1 14:45:20 SELECT c.CourierID, c.ReceiverName, SUM(p.Amount) AS TotalPayments FROM Couriers c JOIN Pay... 0 row(s) returned
```

22.

The screenshot displays the SQL Server Enterprise Manager interface. The main window shows a query editor with the following SQL code:

```
232  
233 /*22*/ SELECT l.LocationName, SUM(p.Amount) AS TotalAmountReceived  
234 FROM Locations l  
235 LEFT JOIN Payments p ON l.LocationID = p.LocationID  
236 WHERE p.PaymentDate > '2023-04-01'  
237 GROUP BY l.LocationName  
238 HAVING SUM(p.Amount) > 5000.00;
```

Below the query editor, the 'Result Grid' is visible, showing a table with two columns: 'LocationName' and 'TotalAmountReceived'. The table is currently empty.

At the bottom, the 'Output' pane shows the execution log:

Time	Action	Message	Duration / Fetch
14:46:04	SELECT l.LocationName, SUM(p.Amount) AS TotalAmountReceived FROM Locations l LEFT JOIN Pa...	5 row(s) returned	0.000 sec / 0.000 sec

TASK4:

23.

The screenshot shows a database query editor with a SQL query for Task 4. The query is as follows:

```
/* ----- TASK 4 ----- */
/*23*/ SELECT p.*, c.*
FROM Payments p
INNER JOIN Couriers c ON p.CourierID = c.CourierID;
```

The results are displayed in a table with the following columns: PaymentID, CourierID, LocationID, Amount, PaymentDate, CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, and Status. The table contains 8 rows of data.

PaymentID	CourierID	LocationID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status
1	1	1	20.00	2023-03-25	1	Raj Sharma	700 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit
2	2	2	30.00	2023-04-02	2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	400 Maple Street, Bangalore	6.00	Delivered
3	3	3	15.00	2023-04-01	3	Jyoti Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit
4	4	4	25.00	2023-04-02	4	Anur Verma	505 Maple Avenue, Pune	Shweta Reddy	600 Elm Street, Ahmedabad	4.50	Delivered
5	5	5	18.00	2023-04-03	5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.20	In Transit
6	6	6	22.00	2023-04-04	6	Rohit Deshmukh	999 Pine Street, Kolkata	Drishya Kapoor	111 Elm Street, Bangalore	6.80	Scheduled
7	7	7	30.00	2023-04-05	7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Avenue, Pune	5.30	Delivered
8	8	8	20.00	2023-04-06	8	Sangeeta Iyer	202 Elm Street, Chennai	Vivik Desai	303 Pine Street, Ahmedabad	9.00	In Transit

The bottom of the screenshot shows the 'Output' section with the following details:

- Time: 14:52:56
- Action: SELECT p.*, c.* FROM Payments p INNER JOIN Couriers c ON p.CourierID = c.CourierID LIMIT 0, 1000
- Message: 8 rows returned
- Duration / Fetch: 0.000 sec / 0.000 sec

24.

The screenshot shows a database query editor with a SQL query for Task 4. The query is as follows:

```
/*24*/ SELECT p.*, l.*
FROM Payments p
INNER JOIN Locations l ON p.LocationID = l.LocationID;
```

The results are displayed in a table with the following columns: PaymentID, CourierID, LocationID, Amount, PaymentDate, LocationID, LocationName, and Address. The table contains 8 rows of data.

PaymentID	CourierID	LocationID	Amount	PaymentDate	LocationID	LocationName	Address
1	1	1	20.00	2023-03-25	1	Main Office	123 Main Street, Delhi
2	2	2	30.00	2023-04-02	2	Branch Office	456 Oak Avenue, Mumbai
3	3	3	15.00	2023-04-01	3	Warehouse A	777 Pine Street, Hyderabad
4	4	4	25.00	2023-04-02	4	Warehouse B	888 Oak Street, Pune
5	5	5	18.00	2023-04-03	5	Branch Office 2	999 Elm Street, Bangalore
6	6	6	22.00	2023-04-04	6	Branch Office 3	101 Maple Avenue, Chennai
7	7	7	30.00	2023-04-05	7	Warehouse C	222 Oak Street, Kolkata
8	8	8	20.00	2023-04-06	8	Branch Office 4	333 Elm Street, Ahmedabad

The bottom of the screenshot shows the 'Output' section with the following details:

- Time: 14:52:42
- Action: SELECT p.*, l.* FROM Payments p INNER JOIN Locations l ON p.LocationID = l.LocationID LIMIT 0, 1
- Message: 8 rows returned
- Duration / Fetch: 0.000 sec / 0.000 sec

25.

SQL Editor

```

249
250
251 * /*25*/ SELECT p.*, c.*, l.*
252       FROM Payments p
253       INNER JOIN Couriers c ON p.CourierID = c.CourierID
254       INNER JOIN Locations l ON p.LocationID = l.LocationID;
255

```

Result Grid

PaymentID	CourierID	LocationID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status
1	1	1	20.00	2023-03-25	1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit
2	2	2	30.00	2023-04-02	2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered
3	3	3	15.00	2023-04-01	3	Jyoti Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit
4	4	4	25.00	2023-04-02	4	Arun Verma	505 Maple Avenue, Pune	Shweta Reddy	606 Elm Street, Ahmedabad	4.50	Delivered
5	5	5	18.00	2023-04-03	5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.20	In Transit
6	6	6	22.00	2023-04-04	6	Rohit Deshmukh	999 Pine Street, Kolkata	Deepika Kapoor	111 Elm Street, Bangalore	6.80	Scheduled
7	7	7	30.00	2023-04-05	7	Raj Sharma	303 Oak Street, Hyderabad	Suran Mehta	404 Maple Avenue, Pune	5.50	Delivered
8	8	8	20.00	2023-04-06	8	Sangeeta Iyer	202 Elm Street, Chennai	Vinod Desai	303 Pine Street, Ahmedabad	8.00	In Transit

Result 11

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:52:42	SELECT p.*, l.* FROM Payments p INNER JOIN Locations l ON p.LocationID = l.LocationID LIMIT 0, 1...	8 row(s) returned	0.000 sec / 0.000 sec
2	14:53:10	SELECT p.*, c.*, l.* FROM Payments p INNER JOIN Couriers c ON p.CourierID = c.CourierID INNE...	8 row(s) returned	0.000 sec / 0.000 sec

26.

SQL Editor

```

255
256
257 * /*26*/ SELECT p.*, c.*
258       FROM Payments p
259       LEFT JOIN Couriers c ON p.CourierID = c.CourierID;
260
261

```

Result Grid

PaymentID	CourierID	LocationID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status
1	1	1	20.00	2023-03-25	1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit
2	2	2	30.00	2023-04-02	2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered
3	3	3	15.00	2023-04-01	3	Jyoti Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit
4	4	4	25.00	2023-04-02	4	Arun Verma	505 Maple Avenue, Pune	Shweta Reddy	606 Elm Street, Ahmedabad	4.50	Delivered
5	5	5	18.00	2023-04-03	5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.20	In Transit
6	6	6	22.00	2023-04-04	6	Rohit Deshmukh	999 Pine Street, Kolkata	Deepika Kapoor	111 Elm Street, Bangalore	6.80	Scheduled
7	7	7	30.00	2023-04-05	7	Raj Sharma	303 Oak Street, Hyderabad	Suran Mehta	404 Maple Avenue, Pune	5.50	Delivered
8	8	8	20.00	2023-04-06	8	Sangeeta Iyer	202 Elm Street, Chennai	Vinod Desai	303 Pine Street, Ahmedabad	8.00	In Transit

Result 12

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:54:03	SELECT p.*, c.* FROM Payments p LEFT JOIN Couriers c ON p.CourierID = c.CourierID LIMIT 0, 1000	8 row(s) returned	0.015 sec / 0.000 sec

27.

The screenshot shows a SQL query in a text editor window. The query is as follows:

```

260
261
262 * /*27*/ SELECT c.CourierID, c.ReceiverName, SUM(p.Amount) AS TotalPayments
263       FROM Couriers c
264       JOIN Payments p ON c.CourierID = p.CourierID
265       GROUP BY c.CourierID, c.ReceiverName;
266

```

Below the query editor, the 'Result Grid' displays the results of the query. It shows a table with three columns: CourierID, ReceiverName, and TotalPayments. The data is as follows:

CourierID	ReceiverName	TotalPayments
1	Suresh Kumar	30.00
2	Arsha Gupta	30.00
3	Alok Gupta	15.00
4	Shweta Reddy	25.00
5	Vinod Dyer	30.00
6	Sneha Kapoor	22.00
7	Suman Mehra	30.00
8	Vivek Desai	30.00

The 'Output' window at the bottom shows the execution details:

Time	Action	Message	Duration / Fetch
1 14:54:30	SELECT c.CourierID, c.ReceiverName, SUM(p.Amount) AS TotalPayments FROM Couriers c JOIN Pay...	8 row(s) returned	0.000 sec / 0.000 sec

28.

The screenshot shows a SQL query in a text editor window. The query is as follows:

```

267
268
269 * /*28*/ SELECT * FROM Payments
270       WHERE PaymentDate = '2023-03-25';
271
272
273

```

Below the query editor, the 'Result Grid' displays the results of the query. It shows a table with five columns: PaymentID, CourierID, LocationID, Amount, and PaymentDate. The data is as follows:

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	1	20.00	2023-03-25

The 'Output' window at the bottom shows the execution details:

Time	Action	Message	Duration / Fetch
1 14:54:08	SELECT * FROM Payments WHERE PaymentDate = '2023-03-25' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

29.

274

275 * /*29*/ SELECT p.*, c.*

276 FROM Payments p

277 LEFT JOIN Couriers c ON p.CourierID = c.CourierID;

278

279

280

PaymentID	CourierID	LocationID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status
1	1	1	30.00	2023-03-25	1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	6.50	In Transit
2	2	2	30.00	2023-04-02	2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	4.00	Delivered
3	3	3	15.00	2023-04-01	3	Jyoti Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit
4	4	4	25.00	2023-04-02	4	Arun Verma	505 Maple Avenue, Pune	Shweta Reddy	606 Elm Street, Ahmedabad	4.50	Delivered
5	5	5	18.00	2023-04-03	5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.20	In Transit
6	6	6	22.00	2023-04-04	6	Rohit Dedeemkh	999 Pine Street, Kolkata	Sneha Kapoor	111 Elm Street, Bangalore	4.80	Scheduled
7	7	7	30.00	2023-04-05	7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Avenue, Pune	5.30	Delivered
8	8	8	30.00	2023-04-06	8	Sangeeta Iyer	202 Elm Street, Chennai	Vivek Desai	303 Pine Street, Ahmedabad	8.00	In Transit

Result 17

Output

Action Output

Time Action Message Duration / Fetch

1 14:56:44 SELECT p.*, c.* FROM Payments p LEFT JOIN Couriers c ON p.CourierID = c.CourierID LIMIT 0, 1000 0 rows returned 0:000 sec / 0:000 sec

30.

280

281

282 * /*30*/ SELECT p.*, l.*

283 FROM Payments p

284 LEFT JOIN Locations l ON p.LocationID = l.LocationID;

285

286

PaymentID	CourierID	LocationID	Amount	PaymentDate	LocationID	LocationName	Address
1	1	1	20.00	2023-03-25	1	Main Office	123 Main Street, Delhi
2	2	2	30.00	2023-04-02	2	Branch Office	456 Oak Avenue, Mumbai
3	3	3	15.00	2023-04-01	3	Warehouse A	777 Pine Street, Hyderabad
4	4	4	25.00	2023-04-02	4	Warehouse B	888 Oak Street, Pune
5	5	5	18.00	2023-04-03	5	Branch Office 2	999 Elm Street, Bangalore
6	6	6	22.00	2023-04-04	6	Branch Office 3	501 Maple Avenue, Chennai
7	7	7	30.00	2023-04-05	7	Warehouse C	222 Oak Street, Kolkata
8	8	8	20.00	2023-04-06	8	Branch Office 4	333 Elm Street, Ahmedabad

Result 18

Output

Action Output

Time Action Message Duration / Fetch

1 14:57:15 SELECT p.*, l.* FROM Payments p LEFT JOIN Locations l ON p.LocationID = l.LocationID LIMIT 0, 1000 0 rows returned 0:000 sec / 0:000 sec

31.

The screenshot shows a SQL query in a text editor window. The query is as follows:

```
285  
286  
287 * /*31*/ SELECT c.CourierID, SUM(p.Amount) AS TotalPayments  
288 FROM Couriers c  
289 LEFT JOIN Payments p ON c.CourierID = p.CourierID  
290 GROUP BY c.CourierID;
```

Below the query editor, the 'Result Grid' displays the results of the query. The columns are 'CourierID' and 'TotalPayments'. The data is as follows:

CourierID	TotalPayments
1	30.00
2	30.00
3	15.00
4	25.00
5	18.00
6	22.00
7	30.00
8	30.00

The 'Output' window at the bottom shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	14:57:58	SELECT c.CourierID, SUM(p.Amount) AS TotalPayments FROM Couriers c LEFT JOIN Payments p ON ...	8 row(s) returned	0.000 sec / 0.000 sec

32.

The screenshot shows a SQL query in a text editor window. The query is as follows:

```
292  
293  
294 * /*32*/ SELECT * FROM Payments  
295 WHERE PaymentDate BETWEEN '2023-03-25' AND '2023-11-30';  
296  
297
```

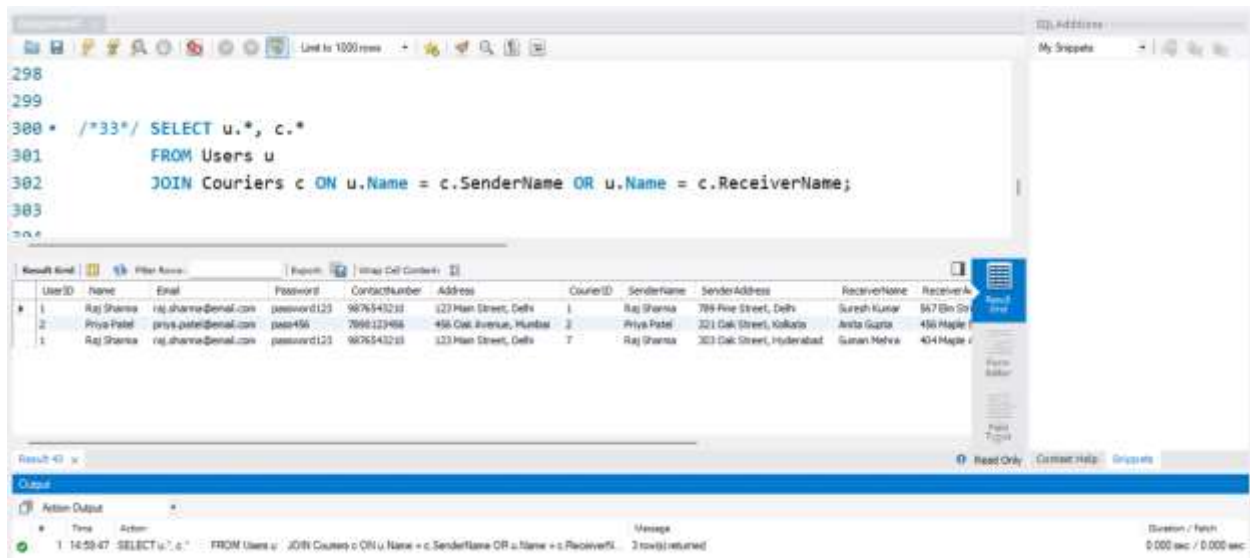
Below the query editor, the 'Result Grid' displays the results of the query. The columns are 'PaymentID', 'CourierID', 'LocationID', 'Amount', and 'PaymentDate'. The data is as follows:

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	1	30.00	2023-03-25
2	2	2	30.00	2023-04-02
3	3	3	15.00	2023-04-01
4	4	4	25.00	2023-04-02
5	5	5	18.00	2023-04-03
6	6	6	22.00	2023-04-04
7	7	7	30.00	2023-04-05
8	8	8	30.00	2023-04-06

The 'Output' window at the bottom shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	14:58:55	SELECT * FROM Payments WHERE PaymentDate BETWEEN '2023-03-25' AND '2023-11-30' LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

33.



The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```

298
299
300 * /*33*/ SELECT u.*, c.*
301         FROM Users u
302         JOIN Couriers c ON u.Name = c.SenderName OR u.Name = c.ReceiverName;
303

```

The results are displayed in a table with the following columns: UserID, Name, Email, Password, ContactNumber, Address, CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress. The table contains 3 rows of data.

UserID	Name	Email	Password	ContactNumber	Address	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress
1	Raj Sharma	raj.sharma@gmail.com	password123	9876543210	123 Main Street, Delhi	1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm St
2	Priya Patel	priya.patel@gmail.com	pass456	7654321098	456 Oak Avenue, Mumbai	2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	432 Maple Lane
3	Raj Sharma	raj.sharma@gmail.com	password123	9876543210	123 Main Street, Delhi	7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Lane

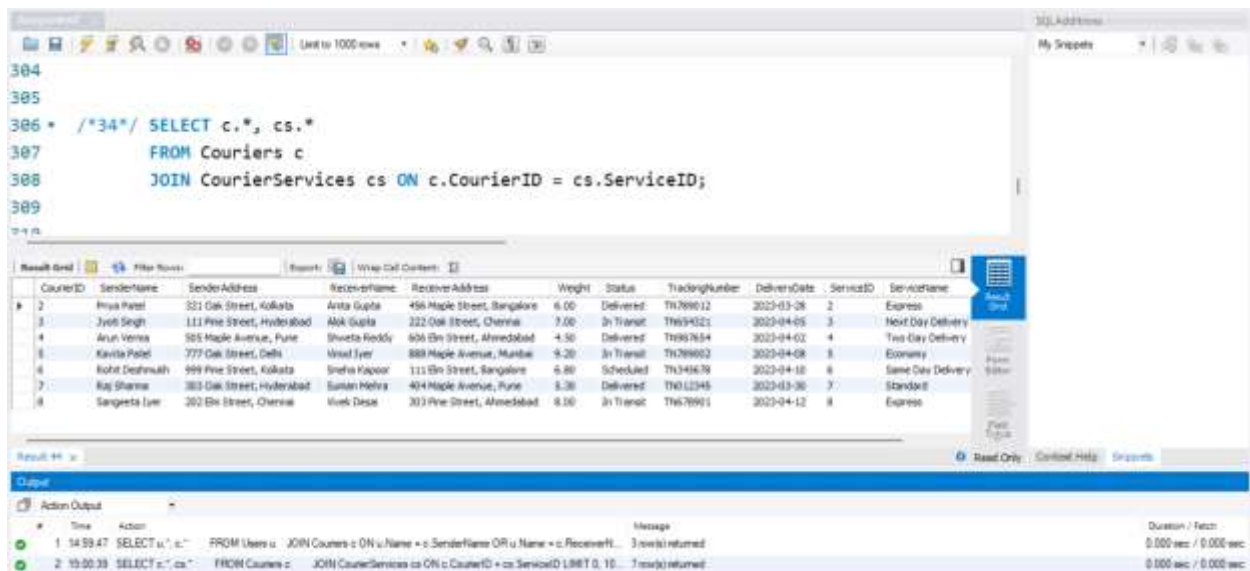
The output section shows the execution details:

```

1 14:59:47 SELECT u.*, c.* FROM Users u JOIN Couriers c ON u.Name = c.SenderName OR u.Name = c.ReceiverName; 3 row(s) returned
Duration / Fetch: 0.000 sec / 0.000 sec

```

34.



The screenshot shows a database query editor with two SQL queries and their results. The first query is as follows:

```

304
305
306 * /*34*/ SELECT c.*, cs.*
307         FROM Couriers c
308         JOIN CourierServices cs ON c.CourierID = cs.ServiceID;
309

```

The results are displayed in a table with the following columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliveredDate, ServiceID, ServiceName. The table contains 8 rows of data.

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveredDate	ServiceID	ServiceName
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered	TH789012	2023-03-28	2	Express
3	Jyoti Singh	111 Pine Street, Hyderabad	Alak Gupta	222 Oak Street, Chennai	7.00	In Transit	TH654321	2023-04-05	3	Next Day Delivery
4	Arun Verma	555 Maple Avenue, Pune	Shweta Reddy	606 Elm Street, Ahmedabad	4.50	Delivered	TH987654	2023-04-02	4	Two Day Delivery
5	Kavita Patel	777 Oak Street, Delhi	Vinod Jyoti	888 Maple Avenue, Mumbai	9.20	In Transit	TH123456	2023-04-08	5	Economy
6	Rohit Deshmukh	999 Pine Street, Kolkata	Shreya Kapoor	111 Elm Street, Bangalore	6.80	Scheduled	TH345678	2023-04-10	6	Same Day Delivery
7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Avenue, Pune	5.30	Delivered	TH012345	2023-03-30	7	Standard
8	Sangeeta Jyoti	202 Elm Street, Chennai	Vivek Desai	303 Pine Street, Ahmedabad	8.10	In Transit	TH678901	2023-04-12	8	Express

The output section shows the execution details for two queries:

```

1 14:59:47 SELECT u.*, c.* FROM Users u JOIN Couriers c ON u.Name = c.SenderName OR u.Name = c.ReceiverName; 3 row(s) returned
Duration / Fetch: 0.000 sec / 0.000 sec

2 15:00:39 SELECT c.*, cs.* FROM Couriers c JOIN CourierServices cs ON c.CourierID = cs.ServiceID LIMIT 0, 10; 7 row(s) returned
Duration / Fetch: 0.000 sec / 0.000 sec

```

35.

The screenshot shows a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query is as follows:

```

311
312 * /*35*/ SELECT e.*, p.*
313         FROM Employees e
314         JOIN Payments p ON e.EmployeeID = p.CourierID;
315
316

```

The results pane displays a table with the following data:

EmployeeID	Name	Email	ContactNumber	Role	Salary	PaymentID	CourierID	LocationID	Amount	PaymentDate
1	Rohan Gupta	rohan.gupta@gmail.com	111-222-3333	Delivery Driver	50000.00	1	1	1	30.00	2023-03-25
2	Chitra Iyer	chitra.iyer@gmail.com	444-333-5555	Customer Service	45000.00	2	2	2	30.00	2023-04-02
3	Vikram Singh	vikram.singh@gmail.com	777-008-1111	Warehouse Manager	33000.00	3	3	3	15.00	2023-04-01
4	Pooja Desai	pooja.desai@gmail.com	444-333-5555	Dispatcher	48000.00	4	4	4	25.00	2023-04-02
5	Kunal Kapoor	kunal.kapoor@gmail.com	111-222-3333	Delivery Driver	52000.00	5	5	5	18.00	2023-04-03
6	Aarti Singh	aarti.singh@gmail.com	666-555-7777	Customer Service	47000.00	6	6	6	22.00	2023-04-04
7	Rajat Mehra	rajat.mehra@gmail.com	888-999-1111	Warehouse Staff	40000.00	7	7	7	30.00	2023-04-05
8	Ananya Iyer	ananya.iyer@gmail.com	222-333-4444	Dispatcher	48000.00	8	8	8	30.00	2023-04-06

The Action Output pane shows the following results:

Time	Action	Message	Duration / Pct
14:59:47	SELECT u.*, s.* FROM Users u JOIN CourierServices cs ON u.Name = cs.SenderName OR s.Name = cs.ReceiverName	3 row(s) returned	0.000 sec / 0.000 sec
15:00:39	SELECT e.*, p.* FROM Employees e JOIN Payments p ON e.EmployeeID = p.CourierID LIMIT 8, 10	7 row(s) returned	0.000 sec / 0.000 sec
15:01:12	SELECT e.*, p.* FROM Employees e JOIN Payments p ON e.EmployeeID = p.CourierID LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

36.

The screenshot shows a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query is as follows:

```

316
317
318 * /*36*/ SELECT u.*, cs.*
319         FROM Users u
320         CROSS JOIN CourierServices cs;
321
322

```

The results pane displays a table with the following data:

UserID	Name	Email	Password	ContactNumber	Address	ServiceID	ServiceName	Cost
6	Meera Iyer	meera.iyer@gmail.com	meera987	3219887654	503 Pine Avenue, Ahmedabad	2	Express	15.00
7	Sanjay Mehra	sanjay.mehra@gmail.com	sanjay789	4321887654	404 Elm Street, Pune	2	Express	15.00
8	Arjun Desai	arjun.desai@gmail.com	arjun456	5432198765	303 Maple Avenue, Chennai	2	Express	15.00
5	Rahul Kapoor	rahul.kapoor@gmail.com	rahul567	6543210987	202 Oak Street, Kolkata	2	Express	15.00
4	Neha Reddy	neha.reddy@gmail.com	neha456	7654321098	101 Pine Avenue, Hyderabad	2	Express	15.00
3	Ankit Verma	ankit.verma@gmail.com	ankit123	8765432109	299 Elm Street, Bangalore	2	Express	15.00
2	Prity Patel	prity.patel@gmail.com	prity456	7890123456	456 Oak Avenue, Mumbai	2	Express	15.00
1	Raj Sharma	raj.sharma@gmail.com	rajsharma123	9876543210	123 Main Street, Delhi	2	Express	15.00
8	Meera Iyer	meera.iyer@gmail.com	meera987	3219887654	503 Pine Avenue, Ahmedabad	3	Next Day Delivery	20.00

The Action Output pane shows the following results:

Time	Action	Message	Duration / Pct
15:01:43	SELECT u.*, cs.* FROM Users u CROSS JOIN CourierServices cs LIMIT 0, 1000	84 row(s) returned	0.313 sec / 0.000 sec

37.

The screenshot shows a database IDE with a SQL query editor at the top and a results pane at the bottom. The query is as follows:

```

322
323
324 * /*37*/ SELECT e.*, l.*
325         FROM Employees e
326         CROSS JOIN Locations l;
327

```

The results pane displays a table with the following columns: EmployeeID, Name, Email, ContactNumber, Role, Salary, LocationID, LocationName, and Address. The table contains 9 rows of data, including employees like Ananya Iyer, Rajat Mehta, and Aarti Singh, each associated with a location (Main Office or Branch Office).

Below the results table, the 'Action Output' pane shows the execution details:

Time	Action	Message	Duration / Fetch
1 15:01:43	SELECT u.*, c.* FROM Users u CROSS JOIN CourierServices cs LIMIT 0, 1000	64 rows returned	0.313 sec / 0.000 sec
2 15:02:06	SELECT e.*, l.* FROM Employees e CROSS JOIN Locations l LIMIT 0, 1000	64 rows returned	0.076 sec / 0.000 sec

38.

The screenshot shows a database IDE with a SQL query editor at the top and a results pane at the bottom. The query is as follows:

```

328
329
330 * /*38*/ SELECT c.*, u.*
331         FROM Couriers c
332         LEFT JOIN Users u ON c.SenderName = u.Name;
333

```

The results pane displays a table with the following columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliverDate, UserID, Name, and Email. The table contains 8 rows of data, showing couriers like Rag Sharma, Riya Patel, and Jyoti Singh, along with their delivery details and the user they are associated with.

Below the results table, the 'Action Output' pane shows the execution details:

Time	Action	Message	Duration / Fetch
1 15:01:43	SELECT u.*, c.* FROM Users u CROSS JOIN CourierServices cs LIMIT 0, 1000	64 rows returned	0.313 sec / 0.000 sec
2 15:02:09	SELECT e.*, l.* FROM Employees e CROSS JOIN Locations l LIMIT 0, 1000	64 rows returned	0.076 sec / 0.000 sec
3 15:02:36	SELECT c.*, u.* FROM Couriers c LEFT JOIN Users u ON c.SenderName = u.Name LIMIT 0, 1000	8 rows returned	0.031 sec / 0.000 sec

39.

334
335
336 * /*39*/ SELECT c.*, u.*
337 FROM Couriers c
338 LEFT JOIN Users u ON c.ReceiverName = u.Name;
339

Result Grid

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TradingNumber	DeliveryDate	UserID	Name	Email	Pa
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	367 Elm Street, Mumbai	8.30	In Transit	TH123456	2023-04-01	0001	0001	0001	0001
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered	TH789012	2023-03-28	0002	0002	0002	0002
3	Yash Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TH54321	2023-04-05	0003	0003	0003	0003
4	Anur Verma	303 Maple Avenue, Pune	Shweta Reddy	604 Elm Street, Ahmedabad	4.30	Delivered	TH987654	2023-04-02	0004	0004	0004	0004
5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.30	In Transit	TH19002	2023-04-08	0005	0005	0005	0005
6	Rohit Deshmukh	999 Pine Street, Kolkata	Neeta Kapoor	111 Elm Street, Bangalore	5.80	Scheduled	TH345678	2023-04-10	0006	0006	0006	0006
7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Avenue, Pune	5.30	Delivered	TH012345	2023-03-30	0007	0007	0007	0007
8	Sangeeta Iyer	202 Elm Street, Chennai	Vivek Desai	303 Pine Street, Ahmedabad	8.00	In Transit	TH678901	2023-04-12	0008	0008	0008	0008

Result 40

Output

Action Output

Time Action Message Duration / Fetch

1 15:03:05 SELECT c.*, u.* FROM Couriers c LEFT JOIN Users u ON c.ReceiverName = u.Name LIMIT 0, 1000 8 row(s) returned 0.000 sec / 0.000 sec

40.

340
341
342 * /*40*/ SELECT c.*, cs.*
343 FROM Couriers c
344 LEFT JOIN CourierServices cs ON c.CourierID = cs.ServiceID;
345

Result Grid

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TradingNumber	DeliveryDate	ServiceID	ServiceName
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	367 Elm Street, Mumbai	8.30	In Transit	TH123456	2023-04-01	0001	Express
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered	TH789012	2023-03-28	2	Express
3	Yash Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TH54321	2023-04-05	3	Next Day Delivery
4	Anur Verma	303 Maple Avenue, Pune	Shweta Reddy	604 Elm Street, Ahmedabad	4.30	Delivered	TH987654	2023-04-02	4	Two-Day Delivery
5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.30	In Transit	TH19002	2023-04-08	5	Economy
6	Rohit Deshmukh	999 Pine Street, Kolkata	Neeta Kapoor	111 Elm Street, Bangalore	5.80	Scheduled	TH345678	2023-04-10	6	Same Day Delivery
7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Avenue, Pune	5.30	Delivered	TH012345	2023-03-30	7	Standard
8	Sangeeta Iyer	202 Elm Street, Chennai	Vivek Desai	303 Pine Street, Ahmedabad	8.00	In Transit	TH678901	2023-04-12	8	Express

Result 33

Output

Action Output

Time Action Message Duration / Fetch

1 15:03:40 SELECT c.*, cs.* FROM Couriers c LEFT JOIN CourierServices cs ON c.CourierID = cs.ServiceID LIMIT 0, 1000 8 row(s) returned 0.000 sec / 0.000 sec

41.

The screenshot shows a SQL query in SQL Studio. The query is as follows:

```

345
346
347 * /*41*/ SELECT e.*, COUNT(c.CourierID) AS TotalCouriersAssigned
348     FROM Employees e
349     LEFT JOIN Couriers c ON e.EmployeeID = c.CourierID
350     GROUP BY e.EmployeeID;

```

The results are displayed in a table with the following columns: EmployeeID, Name, Email, ContactNumber, Role, Salary, and TotalCouriersAssigned. The data is as follows:

EmployeeID	Name	Email	ContactNumber	Role	Salary	TotalCouriersAssigned
1	Rohan Gupta	rohan.gupta@email.com	111-222-3333	Delivery Driver	50000.00	1
2	Chitra Iyer	chitra.iyer@email.com	444-232-5555	Customer Service	45000.00	1
3	Vikram Singh	vikram.singh@email.com	777-888-1111	Warehouse Manager	55000.00	1
4	Pooja Desai	pooja.desai@email.com	444-232-5555	Dispatcher	48000.00	1
5	Kunal Kapoor	kunal.kapoor@email.com	111-222-3333	Delivery Driver	52000.00	1
6	Aarti Singh	aarti.singh@email.com	666-999-7777	Customer Service	47000.00	1
7	Rajat Mehra	rajat.mehra@email.com	888-999-1111	Warehouse Staff	46000.00	1
8	Ananya Iyer	ananya.iyer@email.com	222-333-4444	Dispatcher	49000.00	1

The bottom section shows the execution log with the following entries:

#	Time	Action	Message	Duration / Peak
1	15:03:40	SELECT "c" "c" FROM Couriers c LEFT JOIN CourierServices cs ON c.CourierID = cs.ServiceID LIMIT 5 row(s) returned		0.000 sec / 0.000 sec
2	15:04:02	SELECT e.*, COUNT(c.CourierID) AS TotalCouriersAssigned FROM Employees e LEFT JOIN Couriers c...		0.000 sec / 0.000 sec

42.

The screenshot shows a SQL query in SQL Studio. The query is as follows:

```

352
353 * /*42*/ SELECT l.LocationName, SUM(p.Amount) AS TotalPayments
354     FROM Locations l
355     LEFT JOIN Payments p ON l.LocationID = p.LocationID
356     GROUP BY l.LocationName;
357

```

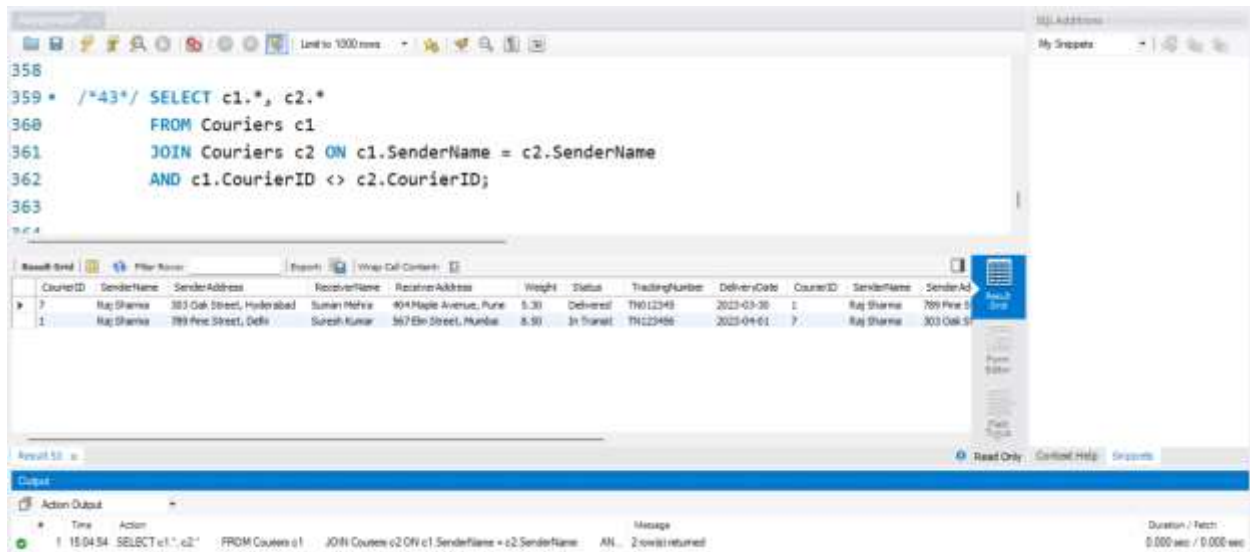
The results are displayed in a table with the following columns: LocationName and TotalPayments. The data is as follows:

LocationName	TotalPayments
Main Office	30.00
Branch Office	30.00
Warehouse A	15.00
Warehouse B	25.00
Branch Office 1	18.00
Branch Office 2	22.00
Warehouse C	30.00
Branch Office 4	20.00

The bottom section shows the execution log with the following entries:

#	Time	Action	Message	Duration / Peak
1	15:03:40	SELECT "c" "c" FROM Couriers c LEFT JOIN CourierServices cs ON c.CourierID = cs.ServiceID LIMIT 5 row(s) returned		0.000 sec / 0.000 sec
2	15:04:03	SELECT e.*, COUNT(c.CourierID) AS TotalCouriersAssigned FROM Employees e LEFT JOIN Couriers c...		0.000 sec / 0.000 sec
3	15:04:26	SELECT LocationName, SUM(p.Amount) AS TotalPayments FROM Locations l LEFT JOIN Payments...		0.000 sec / 0.000 sec

43.



The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```

358
359 * /*43*/ SELECT c1.*, c2.*
360         FROM Couriers c1
361        JOIN Couriers c2 ON c1.SenderName = c2.SenderName
362        AND c1.CourierID <> c2.CourierID;
363

```

The results table shows the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	CourierID	SenderName	SenderAd
1	Raj Sharma	303 Oak Street, Hyderabad	Suman Mohra	404 Maple Avenue, Pune	5.30	Delivered	TN012345	2023-03-30	1	Raj Sharma	303 Oak S
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit	TN223456	2023-04-01	1	Raj Sharma	303 Oak S

The Action Output section shows the following message:

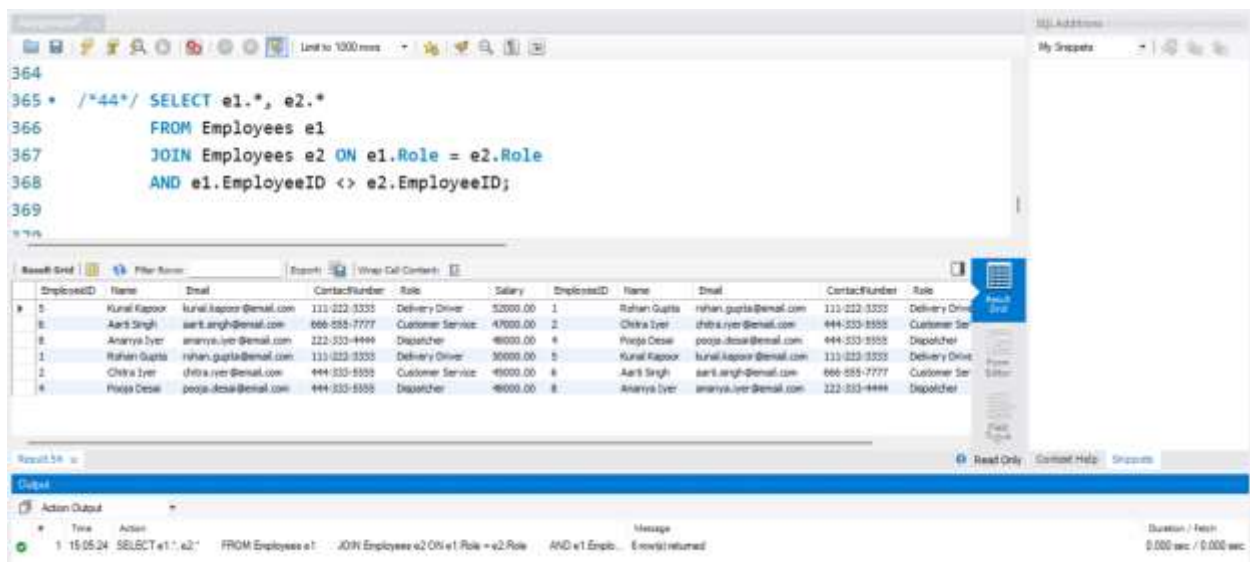
```

1 15:04:54 SELECT c1.*, c2.* FROM Couriers c1 JOIN Couriers c2 ON c1.SenderName = c2.SenderName AND c1.CourierID <> c2.CourierID; 2 rows returned

```

Duration / Fetch: 0.000 sec / 0.000 sec

44.



The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```

364
365 * /*44*/ SELECT e1.*, e2.*
366         FROM Employees e1
367        JOIN Employees e2 ON e1.Role = e2.Role
368        AND e1.EmployeeID <> e2.EmployeeID;
369

```

The results table shows the following data:

EmployeeID	Name	Email	ContactNumber	Role	Salary	EmployeeID	Name	Email	ContactNumber	Role
5	Kunal Kapoor	kunal.kapoor@gmail.com	111-222-3333	Delivery Driver	52000.00	1	Rohan Gupta	rohan.gupta@gmail.com	111-222-3333	Delivery Driver
6	Aarh Singh	aarh.singh@gmail.com	666-555-7777	Customer Service	47000.00	2	Chitra Iyer	chitra.iyer@gmail.com	444-555-8888	Customer Service
8	Ananya Iyer	ananya.iyer@gmail.com	222-333-4444	Dispatcher	48000.00	4	Pooja Desai	pooja.desai@gmail.com	444-555-8888	Dispatcher
1	Rohan Gupta	rohan.gupta@gmail.com	111-222-3333	Delivery Driver	50000.00	5	Kunal Kapoor	kunal.kapoor@gmail.com	111-222-3333	Delivery Driver
2	Chitra Iyer	chitra.iyer@gmail.com	444-555-8888	Customer Service	49000.00	6	Aarh Singh	aarh.singh@gmail.com	666-555-7777	Customer Service
4	Pooja Desai	pooja.desai@gmail.com	444-555-8888	Dispatcher	46000.00	8	Ananya Iyer	ananya.iyer@gmail.com	222-333-4444	Dispatcher

The Action Output section shows the following message:

```

1 15:05:24 SELECT e1.*, e2.* FROM Employees e1 JOIN Employees e2 ON e1.Role = e2.Role AND e1.EmployeeID <> e2.EmployeeID; 6 rows returned

```

Duration / Fetch: 0.000 sec / 0.000 sec

45.

The screenshot shows the SQL Server Enterprise Manager interface. The query editor at the top contains the following SQL code:

```

370
371 * /*45*/ SELECT p.*, l.*
372     FROM Payments p
373     JOIN Couriers c ON p.CourierID = c.CourierID
374     JOIN Locations l ON c.CourierID = l.LocationID;
375

```

The Results grid below the query editor displays the following data:

PaymentID	CourierID	LocationID	Amount	PaymentDate	LocationID	LocationName	Address
1	1	1	30.00	2023-03-26	1	Main Office	123 Main Street, Delhi
2	2	2	30.00	2023-04-01	2	Branch Office	456 Oak Avenue, Mumbai
3	3	3	15.00	2023-04-01	3	Warehouse A	777 Pine Street, Hyderabad
4	4	4	25.00	2023-04-01	4	Warehouse B	888 Oak Street, Pune
5	5	5	18.00	2023-04-01	5	Branch Office 2	999 Elm Street, Bangalore
6	6	6	22.00	2023-04-04	6	Branch Office 3	101 Maple Avenue, Chennai
7	7	7	30.00	2023-04-01	7	Warehouse C	222 Oak Street, Kolkata
8	8	8	30.00	2023-04-06	8	Branch Office 4	333 Elm Street, Ahmedabad

The Output pane at the bottom shows the execution details:

```

1 15:55:55 SELECT p.*, l.* FROM Payments p JOIN Couriers c ON p.CourierID = c.CourierID JOIN Locations l ON c.CourierID = l.LocationID;

```

Duration / Fetch: 0.000 sec / 0.000 sec

46.

The screenshot shows the SQL Server Enterprise Manager interface. The query editor at the top contains the following SQL code:

```

376
377 * /*46*/ SELECT c1.*, c2.*
378     FROM Couriers c1
379     JOIN Couriers c2 ON c1.SenderAddress = c2.SenderAddress
380     AND c1.CourierID <> c2.CourierID;
381

```

The Results grid below the query editor is empty, indicating that no rows were returned by the query.

The Output pane at the bottom shows the execution details:

```

1 15:56:27 SELECT c1.*, c2.* FROM Couriers c1 JOIN Couriers c2 ON c1.SenderAddress = c2.SenderAddress AND c1.CourierID <> c2.CourierID;

```

Duration / Fetch: 0.000 sec / 0.000 sec

47.

The screenshot shows a SQL IDE window with a query editor and a results grid. The query is as follows:

```

382
383 * /*47*/ SELECT e.Name AS EmployeeName, COUNT(c.CourierID) AS TotalCouriersDelivered
384         FROM Employees e
385        LEFT JOIN Couriers c ON e.EmployeeID = c.CourierID
386        GROUP BY e.Name;
387

```

The results grid displays the following data:

EmployeeName	TotalCouriersDelivered
Rohan Gupta	1
Chitra Tyer	1
Vinay Singh	1
Pooja Desai	1
Kunal Kapoor	1
Aarti Singh	1
Rajesh Mehta	1
Ananya Tyer	1

The output pane shows the execution message: "SELECT e.Name AS EmployeeName, COUNT(c.CourierID) AS TotalCouriersDelivered FROM Employees e ... 1 row(s) returned".

48.

The screenshot shows a SQL IDE window with a query editor and a results grid. The query is as follows:

```

388
389 * /*48*/ SELECT c.*, cs.*
390        FROM Couriers c
391       JOIN CourierServices cs ON c.CourierID = cs.ServiceID
392       JOIN Payments p ON c.CourierID = p.CourierID
393      WHERE p.Amount > cs.Cost;

```

The results grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	ServiceID	ServiceName
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.50	Delivered	TH0799012	2023-03-28	2	Express
4	Arun Verma	905 Maple Avenue, Pune	Shweta Reddy	606 Elm Street, Ahmedabad	4.50	Delivered	TH087654	2023-04-03	4	Two Day Delivery
6	Kavita Patel	777 Oak Street, Delhi	Vinay Tyer	888 Maple Avenue, Mumbai	9.20	In Transit	TH099002	2023-04-08	6	Economy
7	Raj Sharma	303 Oak Street, Hyderabad	Susan Mehta	404 Maple Avenue, Pune	5.30	Delivered	TH012345	2023-03-30	7	Standard

The output pane shows the execution message: "SELECT c.*, cs.* FROM Couriers c JOIN CourierServices cs ON c.CourierID = cs.ServiceID JOIN Payments p ON c.CourierID = p.CourierID WHERE p.Amount > cs.Cost ... 4 row(s) returned".

Scope:

49.

The screenshot shows a SQL File 2 window with a query that filters couriers by weight. The query is as follows:

```
394  
395 /*-----Scope-----*/  
396 /*49*/ SELECT * FROM Couriers  
397 WHERE Weight > (SELECT AVG(Weight) FROM Couriers);  
398  
399
```

The result grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit	TH123456	2023-04-01
3	Jyoti Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TH987654	2023-04-05
5	Kavita Patel	777 Oak Street, Delhi	Vinod Gyer	888 Maple Avenue, Mumbai	9.20	In Transit	TH654321	2023-04-08
8	Sargenta Iyer	202 Elm Street, Chennai	Waseem Dsouza	303 Pine Street, Ahmedabad	8.00	In Transit	TH456789	2023-04-12

The output section shows the execution details:

Time	Action	Message	Duration / Fetch
1 15:12:01	SELECT * FROM Couriers WHERE Weight > (SELECT AVG(Weight) FROM Couriers) LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

50.

The screenshot shows a SQL File 2 window with a query that filters employees by salary. The query is as follows:

```
399  
400  
401 /*50*/ SELECT Name FROM Employees  
402 WHERE Salary > (SELECT AVG(Salary) FROM Employees);  
403  
404
```

The result grid displays the following data:

Name
Rohan Gupta
Vikram Singh
Kunal Kapoor

The output section shows the execution details:

Time	Action	Message	Duration / Fetch
1 15:12:27	SELECT Name FROM Employees WHERE Salary > (SELECT AVG(Salary) FROM Employees) LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

51.

SQL File #1

```

104
105 * /*51*/ SELECT SUM(Cost) AS TotalCost
106       FROM CourierServices
107       WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);
108
109

```

Result Grid

TotalCost
122.00

Output

Action Output

Time	Action	Message	Duration / Fetch
15:12:55	SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);	1 row(s) returned	0.000 sec / 0.000 sec

52.

SQL File #1

```

109
110 * /*52*/ SELECT c.*
111       FROM Couriers c
112       WHERE EXISTS
113       (SELECT * FROM Payments p WHERE c.CourierID = p.CourierID);
114

```

Result Grid

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	347 Elm Street, Mumbai	8.50	In Transit	TH123456	2023-04-01
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered	TH789012	2023-03-28
3	Yash Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TH654321	2023-04-05
4	Anita Verma	505 Maple Avenue, Pune	Shweta Reddy	606 Elm Street, Ahmedabad	4.50	Delivered	TH987654	2023-04-02
5	Kavita Patel	777 Oak Street, Delhi	Vinod Iyer	888 Maple Avenue, Mumbai	9.20	In Transit	TH123456	2023-04-08
6	Rohit Deshmukh	999 Pine Street, Kolkata	Neeta Kapoor	111 Elm Street, Bangalore	6.80	Scheduled	TH345678	2023-04-10
7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Avenue, Pune	5.30	Delivered	TH012345	2023-03-30
8	Sangeeta Iyer	202 Elm Street, Chennai	Vishal Desai	303 Pine Street, Ahmedabad	8.00	In Transit	TH678901	2023-04-12

Output

Action Output

Time	Action	Message	Duration / Fetch
15:13:25	SELECT c.* FROM Couriers c WHERE EXISTS (SELECT * FROM Payments p WHERE c.CourierID = p.CourierID);	8 row(s) returned	0.016 sec / 0.000 sec

53.

The screenshot shows the SQL File Editor interface. The query is as follows:

```

114
115 * /*53*/ SELECT l.*
116     FROM Locations l
117     JOIN Payments P ON P.locationID = l.locationID
118     WHERE Amount = (SELECT MAX(Amount) FROM Payments);
119

```

The Results Grid displays the following data:

LocationID	LocationName	Address
2	Branch Office	456 Oak Avenue, Mumbai
7	Warehouse C	222 Oak Street, Kolkata

The Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	15:15:34	SELECT l.* FROM Locations l JOIN Payments P ON P.locationID = l.locationID WHERE Amount = (SELECT MAX(Amount) FROM Payments);		0.016 sec / 0.000 sec

54.

The screenshot shows the SQL File Editor interface. The query is as follows:

```

120
121 * /*54*/ SELECT c1.*
122     FROM Couriers c1
123     WHERE Weight > ALL
124     (SELECT Weight FROM Couriers c2 WHERE c2.SenderName = 'SenderName');
125

```

The Results Grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Raj Sharma	789 Pine Street, Delhi	Suresh Kumar	567 Elm Street, Mumbai	8.50	In Transit	TH123456	2023-04-01
2	Priya Patel	321 Oak Street, Kolkata	Anita Gupta	456 Maple Street, Bangalore	6.00	Delivered	TH789012	2023-03-28
3	Yash Singh	111 Pine Street, Hyderabad	Alok Gupta	222 Oak Street, Chennai	7.00	In Transit	TH654321	2023-04-05
4	Arun Verma	555 Maple Avenue, Pune	Shweta Reddy	606 Elm Street, Ahmedabad	4.50	Delivered	TH987654	2023-04-02
5	Kavita Patel	777 Oak Street, Delhi	Wood Iyer	888 Maple Avenue, Mumbai	9.20	In Transit	TH765002	2023-04-08
6	Rohit Deshmukh	999 Pine Street, Kolkata	Sneha Kapoor	111 Elm Street, Bangalore	6.80	Scheduled	TH345678	2023-04-30
7	Raj Sharma	303 Oak Street, Hyderabad	Suman Mehra	404 Maple Avenue, Pune	5.30	Delivered	TH012345	2023-03-30
8	Sangeeta Iyer	202 Elm Street, Chennai	Vivek Datta	303 Pine Street, Ahmedabad	8.00	In Transit	TH678901	2023-04-12

The Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	15:16:09	SELECT c1.* FROM Couriers c1 WHERE Weight > ALL (SELECT Weight FROM Couriers c2 WHERE c2.SenderName = 'SenderName');		0.000 sec / 0.000 sec