

Task - 2

1. List all customers:

The screenshot shows a database application interface. On the left is a 'Navigator' pane with a 'SCHEMAS' tree. The 'hexa' database is expanded, showing 'Tables' (courier, courierservi, employee, location, payment, user) and 'Views' (praveendb, sys). The 'user' table is selected. The main pane shows a query editor with the query: `SELECT * FROM user;`. Below the editor is a 'Result Grid' showing 10 rows of data. The columns are UserID, Name, Email, Password, ContactNumber, and Address. The data includes users like Alice Johnson, Bob Smith, Charlie Brown, David Lee, Eva Green, Frank White, Grace Taylor, Hannah Davis, Ivy Martin, and Jack Brown. The bottom status bar shows 'user 1' and 'No object selected'.

UserID	Name	Email	Password	ContactNumber	Address
1	Alice Johnson	alice@example.com	password1	123-456-7890	123 Main St, City, State, Zip
2	Bob Smith	bob@example.com	password2	234-567-8901	456 Elm St, City, State, Zip
3	Charlie Brown	charlie@example.com	password3	345-678-9012	789 Oak St, City, State, Zip
4	David Lee	david@example.com	password4	456-789-0123	012 Pine St, City, State, Zip
5	Eva Green	eva@example.com	password5	567-890-1234	345 Cedar St, City, State, Zip
6	Frank White	frank@example.com	password6	678-901-2345	678 Maple St, City, State, Zip
7	Grace Taylor	grace@example.com	password7	789-012-3456	901 Birch St, City, State, Zip
8	Hannah Davis	hannah@example.com	password8	890-123-4567	234 Walnut St, City, State, Zip
9	Ivy Martin	ivy@example.com	password9	901-234-5678	567 Pineapple St, City, State, Zip
10	Jack Brown	jack@example.com	password10	012-345-6789	890 Orange St, City, State, Zip

2. List all orders for a specific customer:

The screenshot shows the same database application interface. The 'courier' table is selected in the 'Navigator' pane. The query editor shows the query: `select * from courier where senderName = "Alice Johnson"`. The 'Result Grid' shows 6 rows of data. The columns are CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, and TrackingNumber. The data shows orders sent by Alice Johnson to various recipients like Bob Smith, Charlie Brown, David Lee, Eva Green, Frank White, and Frank White. The bottom status bar shows 'courier'.

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber
1	Alice Johnson	123 Main St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.50	Processing	TRACK123
2	Alice Johnson	123 Main St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.00	Delivered	TRACK134
3	Alice Johnson	123 Main St, City, State, Zip	David Lee	012 Pine St, City, State, Zip	1.80	Cancelled	TRACK135
4	Alice Johnson	123 Main St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.20	In Transit	TRACK136
5	Alice Johnson	123 Main St, City, State, Zip	Frank White	678 Maple St, City, State, Zip	5.10	Processing	TRACK137
15	Alice Johnson	123 Main St, City, State, Zip	Frank White	678 Maple St, City, State, Zip	5.00	Delivered	TRACK147

3. List all couriers:

The screenshot shows a database query tool interface. On the left, a 'SCHEMAS' pane lists tables including 'courier'. The main query editor displays the query: `SELECT * FROM courier;`. The 'Result Grid' shows 13 rows of data with columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, and Tracking.

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	Tracking
1	Alice Johnson	123 Main St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.50	Processing	TRACK1
2	Alice Johnson	123 Main St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.00	Delivered	TRACK1
3	Alice Johnson	123 Main St, City, State, Zip	David Lee	012 Pine St, City, State, Zip	1.80	Cancelled	TRACK1
4	Alice Johnson	123 Main St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.20	In Transit	TRACK1
5	Alice Johnson	123 Main St, City, State, Zip	Frank White	678 Maple St, City, State, Zip	5.10	Processing	TRACK1
6	Bob Smith	456 Elm St, City, State, Zip	Grace Taylor	901 Birch St, City, State, Zip	3.50	Delivered	TRACK1
7	Charlie Brown	789 Oak St, City, State, Zip	Hannah Davis	234 Walnut St, City, State, Zip	2.00	Processing	TRACK1
8	David Lee	012 Pine St, City, State, Zip	Ivy Martin	567 Pineapple St, City, State, Zip	2.80	Delivered	TRACK1
9	Eva Green	345 Cedar St, City, State, Zip	Jack Brown	890 Orange St, City, State, Zip	3.90	Cancelled	TRACK1
10	Frank White	678 Maple St, City, State, Zip	Alice Johnson	123 Main St, City, State, Zip	4.50	In Transit	TRACK1
11	Grace Taylor	901 Birch St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.70	Processing	TRACK1
12	Hannah Davis	234 Walnut St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.40	Delivered	TRACK1
13	Ivy Martin	567 Pineapple St, City, State...	David Lee	012 Pine St, City, State, Zip	1.90	Processing	TRACK1

4. List all packages for a specific order:

The screenshot shows the same database query tool interface. The query editor displays the query: `SELECT * FROM courier where TrackingNumber = "TRACK134";`. The 'Result Grid' shows a single row of data for the package with TrackingNumber 'TRACK134'.

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber
2	Alice Johnson	123 Main St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.00	Delivered	TRACK134

5. List all deliveries for a specific courier:

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'courier', 'courierservi', 'employee', 'location', 'payment', 'user', and views like 'praveendb' and 'sys'. The main query editor displays the following SQL query:

```
1 • SELECT * FROM courier where CourierID = 4;
```

The 'Result Grid' shows the results of the query, displaying a single row of data for CourierID 4:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber
4	Alice Johnson	123 Main St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.20	In Transit	TRACK136

6. List all undelivered packages:

The screenshot shows the same database management tool interface. The query editor displays the following SQL query:

```
1 • SELECT * FROM courier where status <> "Delivered"
```

The 'Result Grid' shows the results of the query, displaying a list of packages that are not in the 'Delivered' status:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingN
1	Alice Johnson	123 Main St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.50	Processing	TRACK123
3	Alice Johnson	123 Main St, City, State, Zip	David Lee	012 Pine St, City, State, Zip	1.80	Cancelled	TRACK135
4	Alice Johnson	123 Main St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.20	In Transit	TRACK136
5	Alice Johnson	123 Main St, City, State, Zip	Frank White	678 Maple St, City, State, Zip	5.10	Processing	TRACK137
7	Charlie Brown	789 Oak St, City, State, Zip	Hannah Davis	234 Walnut St, City, State, Zip	2.00	Processing	TRACK139
9	Eva Green	345 Cedar St, City, State, Zip	Jack Brown	890 Orange St, City, State, Zip	3.90	Cancelled	TRACK141
10	Frank White	678 Maple St, City, State, Zip	Alice Johnson	123 Main St, City, State, Zip	4.50	In Transit	TRACK142
11	Grace Taylor	901 Birch St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.70	Processing	TRACK143
13	Ivy Martin	567 Pineapple St, City, State, Zip	David Lee	012 Pine St, City, State, Zip	1.90	Processing	TRACK145
14	Jack Brown	890 Orange St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.30	In Transit	TRACK146

7. List all packages that are scheduled for delivery today:

Navigation: Query 1 Query 2 customers employee user courier courierservices

SCHEMAS

Filter objects

- hexa
 - Tables
 - courier
 - courierservi
 - employee
 - location
 - payment
 - user
 - Views
 - Stored Procedure
 - Functions
 - praveendb
 - sys

Administration Information

Query 1 Query 2 Limit to 1000 rows

1 • `SELECT * FROM courier where DeliveryDate = CURDATE()`

Result Grid

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNum
2	Alice Johnson	123 Main St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.00	Delivered	TRACK134
5	Alice Johnson	123 Main St, City, State, Zip	Frank White	678 Maple St, City, State, Zip	5.10	Processing	TRACK137
9	Eva Green	345 Cedar St, City, State, Zip	Jack Brown	890 Orange St, City, State, Zip	3.90	Cancelled	TRACK141
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Form Editor

Field Types

Query Stats

8. List all packages with a specific status:

Navigation

SCHEMAS

Filter objects

hexa

Tables

courier

courierservice

employee

location

payment

user

Views

Stored Procedure

Functions

praveendb

sys

Administration

Information

Query 1

Query 1

Query 2

customers

employee

courier

courier

courierservices

Limit to 1000 rows

1 •

SELECT * FROM courier where status = "Processing"

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content

	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingN
▶	1	Alice Johnson	123 Main St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.50	Processing	TRACK123
	5	Alice Johnson	123 Main St, City, State, Zip	Frank White	678 Maple St, City, State, Zip	5.10	Processing	TRACK137
	7	Charlie Brown	789 Oak St, City, State, Zip	Hannah Davis	234 Walnut St, City, State, Zip	2.00	Processing	TRACK139
	11	Grace Taylor	901 Birch St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.70	Processing	TRACK143
	13	Ivy Martin	567 Pineapple St, City, State, Zip	David Lee	012 Pine St, City, State, Zip	1.90	Processing	TRACK145
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Form Editor

Field Types

Query Stats

9. Calculate the total number of packages for each courier.

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'courier', 'courierservice', 'employee', 'location', 'payment', 'user', 'praveendb', and 'sys'. The main query editor displays the following SQL query:

```
1 SELECT SenderName, COUNT(*) AS TotalPackages
2 FROM Courier
3 GROUP BY SenderName;
```

The 'Result Grid' shows the results of the query:

SenderName	TotalPackages
Alice Johnson	6
Bob Smith	1
Charlie Brown	1
David Lee	1
Eva Green	1
Frank White	1
Grace Taylor	1
Hannah Davis	1
Ivy Martin	1
Jack Brown	1

10. Find the average delivery time for each courier

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure. The main query editor displays the following SQL query:

```
1 SELECT c.CourierID, c.SenderName, AVG(DATEDIFF(c.DeliveryDate, p.PaymentDate)) AS AvgDeliveryTime
2 FROM Courier c
3 JOIN Payment p ON c.CourierID = p.CourierID
4 GROUP BY c.CourierID, c.SenderName;
```

The 'Result Grid' shows the results of the query:

CourierID	SenderName	AvgDeliveryTime
1	Alice Johnson	5.0000
2	Alice Johnson	2.0000
3	Alice Johnson	4.0000
4	Alice Johnson	4.0000
5	Alice Johnson	-1.0000
6	Bob Smith	4.0000
7	Charlie Brown	4.0000
8	David Lee	4.0000
9	Eva Green	-5.0000
10	Frank White	4.0000
11	Grace Taylor	4.0000
12	Hannah Davis	4.0000
13	Ivy Martin	4.0000
14	Jack Brown	4.0000

11. List all packages with a specific weight range:

The screenshot shows a database management interface with a 'Navigator' on the left and a main workspace. The 'Navigator' displays a schema named 'hexa' with tables: courier, courierservice, employee, location, payment, user, and sys. The main workspace shows a query editor with the following SQL query:

```
1 • SELECT * FROM courier where weight >3 and weight <5;
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, and TrackingNum. The results are as follows:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNum
4	Alice Johnson	123 Main St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.20	In Transit	TRACK136
6	Bob Smith	456 Elm St, City, State, Zip	Grace Taylor	901 Birch St, City, State, Zip	3.50	Delivered	TRACK138
9	Eva Green	345 Cedar St, City, State, Zip	Jack Brown	890 Orange St, City, State, Zip	3.90	Cancelled	TRACK141
10	Frank White	678 Maple St, City, State, Zip	Alice Johnson	123 Main St, City, State, Zip	4.50	In Transit	TRACK142
12	Hannah Davis	234 Walnut St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.40	Delivered	TRACK144
14	Jack Brown	890 Orange St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.30	In Transit	TRACK146
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

12. Retrieve employees whose names contain 'John'

The screenshot shows the same database management interface. The 'Navigator' on the left shows the 'hexa' schema with tables: courier, courierservice, employee, location, payment, user, and sys. The main workspace shows a query editor with the following SQL query:

```
1 • SELECT * FROM user where name like "%john%";
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has columns: UserID, Name, Email, Password, ContactNumber, and Address. The results are as follows:

UserID	Name	Email	Password	ContactNumber	Address
1	Alice Johnson	alice@example.com	password1	123-456-7890	123 Main St, City, State, Zip
*	NULL	NULL	NULL	NULL	NULL

13. Retrieve all courier records with payments greater than \$50.

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

```
1 • USE hexa;
2
3 • SELECT c.*, p.Amount
4 FROM Courier c
5 JOIN Payment p ON c.CourierID = p.CourierID
6 WHERE p.Amount > 50.00;
```

The results are displayed in a table with the following columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliveryDate, and Amount. The table contains 10 rows of data.

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	Amount
6	Bob Smith	456 Elm St, City, State, Zip	Grace Taylor	901 Birch St, City, State, Zip	3.50	Delivered	TRACK138	2024-04-19	70.00
7	Charlie Brown	789 Oak St, City, State, Zip	Hannah Davis	234 Walnut St, City, State, Zip	2.00	Processing	TRACK139	2024-04-20	65.00
8	David Lee	012 Pine St, City, State, Zip	Ivy Martin	567 Pineapple St, City, State, Zip	2.80	Delivered	TRACK140	2024-04-21	55.00
10	Frank White	678 Maple St, City, State, Zip	Alice Johnson	123 Main St, City, State, Zip	4.50	In Transit	TRACK142	2024-04-23	90.00
11	Grace Taylor	901 Birch St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.70	Processing	TRACK143	2024-04-24	75.00
12	Hannah Davis	234 Walnut St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.40	Delivered	TRACK144	2024-04-25	85.00
13	Ivy Martin	567 Pineapple St, City, State, Zip	David Lee	012 Pine St, City, State, Zip	1.90	Processing	TRACK145	2024-04-26	100.00
14	Jack Brown	890 Orange St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.30	In Transit	TRACK146	2024-04-27	95.00
15	Alice Johnson	123 Main St, City, State, Zip	Frank White	678 Maple St, City, State, Zip	5.00	Delivered	TRACK147	2024-04-28	110.00

TASK-3

14. Find the total number of couriers handled by each employee.

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

```
1 • USE hexa;
2
3 • SELECT e.EmployeeID, e.Name AS EmployeeName, COUNT(c.CourierID) AS TotalCouriersHandled
4 FROM Employee e
5 LEFT JOIN Courier c ON e.EmployeeID = c.CourierID
6 GROUP BY e.EmployeeID, e.Name;
```

The results are displayed in a table with the following columns: EmployeeID, EmployeeName, and TotalCouriersHandled. The table contains 5 rows of data.

EmployeeID	EmployeeName	TotalCouriersHandled
1	John Doe	1
2	Jane Smith	1
3	Emily Johnson	1
4	Michael Brown	1
5	Sophia Wilson	1

15. Calculate the total revenue generated by each location

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, and user. The central query editor contains the following SQL code:

```
1 • USE hexa;
2 • SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalRevenue
3 FROM Location l
4 LEFT JOIN Payment p ON l.LocationID = p.LocationId
5 GROUP BY l.LocationID, l.LocationName;
```

The 'Result Grid' at the bottom displays the query results:

LocationID	LocationName	TotalRevenue
1	City Center	185.00
2	East End	280.00
3	West Side	270.00
4	North Park	180.00

16. Find the total number of couriers delivered to each location.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, and user. The central query editor contains the following SQL code:

```
1 • USE hexa;
2 • SELECT courier.ReceiverAddress , COUNT(CourierID) AS TotalCouriersDelivered
3 FROM courier
4 GROUP BY courier.ReceiverAddress;
```

The 'Result Grid' at the bottom displays the query results:

ReceiverAddress	TotalCouriersDelivered
456 Elm St, City, State, Zip	2
789 Oak St, City, State, Zip	2
012 Pine St, City, State, Zip	2
345 Cedar St, City, State, Zip	2
678 Maple St, City, State, Zip	2
901 Birch St, City, State, Zip	1
234 Walnut St, City, State, Zip	1
567 Pineapple St, City, State, Zip	1
890 Orange St, City, State, Zip	1
123 Main St, City, State, Zip	1

17. Find the courier with the highest average delivery time:

The screenshot shows a database query editor with a SQL query in the main window. The query is as follows:

```
1 • USE hexa;
2 • SELECT c.SenderName, AVG(DATEDIFF(c.DeliveryDate, p.PaymentDate)) AS AvgDeliveryTime
3 FROM Courier c
4 JOIN Payment p ON c.CourierID = p.CourierID
5 GROUP BY c.SenderName
6 ORDER BY AvgDeliveryTime DESC
7 LIMIT 1;
```

The result grid below the query shows the following data:

SenderName	AvgDeliveryTime
Jack Brown	4.0000

The left sidebar shows the database schema with tables: courier, courierservices, employee, location, payment, user, Views, Stored Procedures, Functions, praveendb, and sys. The bottom of the sidebar has tabs for Administration, Schemas, and Information.

18. Find Locations with Total Payments Less Than a Certain Amount

The screenshot shows a database query editor with a SQL query in the main window. The query is as follows:

```
1 • USE hexa;
2 • SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPayments
3 FROM Location l
4 LEFT JOIN Payment p ON l.LocationID = p.LocationId
5 GROUP BY l.LocationID, l.LocationName
6 HAVING TotalPayments < 1000;
7
```

The result grid below the query shows the following data:

LocationID	LocationName	TotalPayments
1	City Center	185.00
2	East End	280.00
3	West Side	270.00
4	North Park	180.00

The left sidebar shows the database schema with tables: courier, courierservices, employee, location, payment, user, Views, Stored Procedures, Functions, praveendb, and sys. The bottom of the sidebar has tabs for Administration, Schemas, and Information.

19. Calculate Total Payments per Location

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, user, praveendb, and sys. The right pane shows a query window with the following SQL code:

```
1 • USE hexa;
2 • SELECT l.LocationName, SUM(p.Amount) AS TotalPayments
3 FROM Location l
4 LEFT JOIN Payment p ON l.LocationID = p.LocationID
5 GROUP BY l.LocationID;
```

Below the query editor, the 'Result Grid' shows the following data:

LocationName	TotalPayments
City Center	185.00
East End	280.00
West Side	270.00
North Park	180.00

20. Retrieve couriers who have received payments totaling more than \$1000 in a specific location (LocationID = X):

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, user, praveendb, and sys. The right pane shows a query window with the following SQL code:

```
1 select courier.courierid, sum(amount) as total, senderaddress from courier
2 join payment on courier.CourierID = payment.CourierID
3 join location on courier.SenderAddress = location.Address
4 where LocationName = "North Park"
5 group by CourierID having total>=1000
6
```

Below the query editor, the 'Result Grid' shows the following data:

courierid	total	senderaddress
10	1200.00	400 Pine St, City, State, Zip

21. Retrieve couriers who have received payments totaling more than \$1000 after a certain date
(PaymentDate > 'YYYY-MM-DD'):

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, user, and Views. The right pane shows a query window with the following SQL code:

```
1 SELECT c.CourierID, c.SenderName, SUM(p.Amount) AS TotalPayments
2 FROM Courier c
3 JOIN Payment p ON c.CourierID = p.CourierID
4 WHERE p.PaymentDate > '2024-04-10'
5 GROUP BY c.CourierID, c.SenderName
6 HAVING TotalPayments > 1000;
```

The 'Result Grid' shows the following data:

CourierID	SenderName	TotalPayments
2	Alice Johnson	1400.00
7	Charlie Brown	1400.00
10	Frank White	1200.00

22. Retrieve locations where the total amount received is more than \$5000 before a certain date
(PaymentDate > 'YYYY-MM-DD')

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, user, and Views. The right pane shows a query window with the following SQL code:

```
1 SELECT c.SenderName, SUM(p.Amount) AS TotalPayments
2 FROM Courier c
3 JOIN Payment p ON c.CourierID = p.CourierID
4 WHERE p.PaymentDate > '2024-04-01'
5 GROUP BY c.SenderName
6 HAVING TotalPayments > 5000;
```

The 'Result Grid' shows the following data:

SenderName	TotalPayments
Alice Johnson	5880.00

Task 4:

23. Retrieve Payments with Courier Information

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema, including tables like 'courier', 'courservices', 'employee', 'location', 'payment', and 'user'. The right pane shows a query window with the following SQL code:

```
1 SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.SenderName, c.SenderAddress, c.ReceiverName, c.ReceiverAddress, c.  
2 FROM Payment p  
3 JOIN Courier c ON p.CourierID = c.CourierID;  
4  
5  
6
```

The query results are displayed in a grid with the following columns: PaymentID, Amount, PaymentDate, CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, and Status. The results show 11 rows of data.

PaymentID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status
1	4000.00	2024-04-10	1	Alice Johnson	123 Main St, City, State, Zip	Bob Smith	400 Pine St, City, State, Zip	2.50	Prox
2	1400.00	2024-04-11	2	Alice Johnson	123 Main St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.00	Deliv
3	300.00	2024-04-12	3	Alice Johnson	123 Main St, City, State, Zip	David Lee	400 Pine St, City, State, Zip	1.80	Can
4	40.00	2024-04-13	4	Alice Johnson	123 Main St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.20	In T
5	30.00	2024-04-14	5	Alice Johnson	123 Main St, City, State, Zip	Frank White	400 Pine St, City, State, Zip	5.10	Prox
6	70.00	2024-04-15	6	Bob Smith	456 Elm St, City, State, Zip	Grace Taylor	901 Birch St, City, State, Zip	3.50	Deliv
7	1400.00	2024-04-16	7	Charlie Brown	789 Oak St, City, State, Zip	Hannah Davis	234 Walnut St, City, State, Zip	2.00	Prox
8	55.00	2024-04-17	8	David Lee	012 Pine St, City, State, Zip	Ivy Martin	567 Pineapple St, City, State, Zip	2.80	Deliv
9	40.00	2024-04-18	9	Eva Green	345 Cedar St, City, State, Zip	Jack Brown	400 Pine St, City, State, Zip	3.90	Can
10	1200.00	2024-04-19	10	Frank White	400 Pine St, City, State, Zip	Alice Johnson	123 Main St, City, State, Zip	4.50	In T
11	75.00	2024-04-20	11	Grace Taylor	901 Birch St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.70	Prox

24. Retrieve Payments with Location Information

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema, including tables like 'courier', 'courservices', 'employee', 'location', 'payment', and 'user'. The right pane shows a query window with the following SQL code:

```
1 SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName, l.Address AS LocationAddress  
2 FROM Payment p  
3 JOIN Location l ON p.LocationID = l.LocationID;  
4  
5  
6  
7
```

The query results are displayed in a grid with the following columns: PaymentID, Amount, PaymentDate, LocationID, LocationName, and LocationAddress. The results show 15 rows of data.

PaymentID	Amount	PaymentDate	LocationID	LocationName	LocationAddress
1	4000.00	2024-04-10	1	City Center	100 Main St, City, State, Zip
5	30.00	2024-04-14	1	City Center	100 Main St, City, State, Zip
9	40.00	2024-04-18	1	City Center	100 Main St, City, State, Zip
13	100.00	2024-04-22	1	City Center	100 Main St, City, State, Zip
2	1400.00	2024-04-11	2	East End	200 Elm St, City, State, Zip
6	70.00	2024-04-15	2	East End	200 Elm St, City, State, Zip
10	1200.00	2024-04-19	2	East End	200 Elm St, City, State, Zip
14	95.00	2024-04-23	2	East End	200 Elm St, City, State, Zip
3	300.00	2024-04-12	3	West Side	300 Oak St, City, State, Zip
7	1400.00	2024-04-16	3	West Side	300 Oak St, City, State, Zip
11	75.00	2024-04-20	3	West Side	300 Oak St, City, State, Zip
15	110.00	2024-04-24	3	West Side	300 Oak St, City, State, Zip

25. Retrieve Payments with Courier and Location Information

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema, including tables like 'courier', 'courierservices', 'employee', 'location', 'payment', and 'user'. The right pane shows a query window with the following SQL code:

```
1 SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName, l.Address AS LocationAddress
2 FROM Payment p
3 JOIN Location l ON p.LocationID = l.LocationID;
```

The 'Result Grid' at the bottom displays the query results with columns: PaymentID, Amount, PaymentDate, LocationID, LocationName, and LocationAddress. The results show 15 rows of payment data, including details like PaymentID, Amount, PaymentDate, LocationID, LocationName, and LocationAddress.

26. List all payments with courier details

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema, including tables like 'courier', 'courierservices', 'employee', 'location', 'payment', and 'user'. The right pane shows a query window with the following SQL code:

```
1 SELECT p.PaymentID, p.Amount, p.PaymentDate,
2 c.CourierID, c.SenderName, c.SenderAddress,
3 c.ReceiverName, c.ReceiverAddress, c.Weight,
4 c.Status, c.TrackingNumber, c.DeliveryDate
5 FROM Payment p
6 JOIN Courier c ON p.CourierID = c.CourierID;
```

The 'Result Grid' at the bottom displays the query results with columns: PaymentID, Amount, PaymentDate, CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, and Status. The results show 15 rows of payment data, including details like PaymentID, Amount, PaymentDate, CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, and Status.

27. Total payments received for each courier

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' schema with tables: courier, courierservices, employee, location, payment, user, and views. The right pane shows a query window with the following SQL:

```
1 SELECT c.SenderName, SUM(p.Amount) AS TotalPayments
2 FROM Courier c
3 JOIN Payment p ON c.CourierID = p.CourierID
4 GROUP BY c.SenderName
```

The 'Result Grid' shows the following data:

SenderName	TotalPayments
Alice Johnson	5880.00
Bob Smith	70.00
Charlie Brown	1400.00
David Lee	55.00
Eva Green	40.00
Frank White	1200.00
Grace Taylor	75.00
Hannah Davis	85.00
Ivy Martin	100.00
Jack Brown	95.00

The status bar indicates 'Schema: hexa' and 'Result 21'.

28. List payments made on a specific date

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' schema with tables: courier, courierservices, employee, location, payment, user, and views. The right pane shows a query window with the following SQL:

```
1 SELECT *
2 FROM Payment
3 WHERE PaymentDate = '2024-04-14';
```

The 'Result Grid' shows the following data:

PaymentID	CourierID	LocationID	Amount	PaymentDate
5	5	1	30.00	2024-04-14

The status bar indicates 'Schema: hexa' and 'Payment 22'.

29. Get Courier Information for Each Payment

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, and user. The 'location' table has columns: LocationID, LocationName, and Address. The 'payment' table has columns: PaymentID, Amount, PaymentDate, CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, and Status. The 'courier' table has columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, and DeliveryDate. The 'employee' table has columns: EmployeeID, Name, Address, and Status. The 'courierservices' table has columns: ServiceID, Name, Address, and Status. The 'user' table has columns: UserID, Name, Address, and Status. The 'sys' table has columns: SysID, Name, Address, and Status. The 'Administration' table has columns: AdminID, Name, Address, and Status. The 'Information' table has columns: InfoID, Name, Address, and Status. The 'Schemas' table has columns: SchemaID, Name, Address, and Status. The 'Tables' table has columns: TableID, Name, Address, and Status. The 'Views' table has columns: ViewID, Name, Address, and Status. The 'Stored Procedures' table has columns: SPID, Name, Address, and Status. The 'Functions' table has columns: FID, Name, Address, and Status. The 'Triggers' table has columns: TrID, Name, Address, and Status. The 'Indexes' table has columns: IndID, Name, Address, and Status. The 'Foreign Keys' table has columns: FKID, Name, Address, and Status. The 'Fields' table has columns: FieldID, Name, Address, and Status. The 'Types' table has columns:TypeID, Name, Address, and Status. The 'Query' pane shows the following SQL query:

```
1 SELECT p.PaymentID, p.Amount, p.PaymentDate,
2       c.CourierID, c.SenderName, c.SenderAddress,
3       c.ReceiverName, c.ReceiverAddress, c.Weight,
4       c.Status, c.TrackingNumber, c.DeliveryDate
5 FROM Payment p
6 JOIN Courier c ON p.CourierID = c.CourierID;
```

The 'Result Grid' shows the following data:

PaymentID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status
1	4000.00	2024-04-10	1	Alice Johnson	123 Main St, City, State, Zip	Bob Smith	400 Pine St, City, State, Zip	2.50	Proc
2	1400.00	2024-04-11	2	Alice Johnson	123 Main St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.00	Delv
3	300.00	2024-04-12	3	Alice Johnson	123 Main St, City, State, Zip	David Lee	400 Pine St, City, State, Zip	1.80	Can
4	40.00	2024-04-13	4	Alice Johnson	123 Main St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.20	In T
5	30.00	2024-04-14	5	Alice Johnson	123 Main St, City, State, Zip	Frank White	400 Pine St, City, State, Zip	5.10	Proc
6	70.00	2024-04-15	6	Bob Smith	456 Elm St, City, State, Zip	Grace Taylor	901 Birch St, City, State, Zip	3.50	Delv
7	1400.00	2024-04-16	7	Charlie Brown	789 Oak St, City, State, Zip	Hannah Davis	234 Walnut St, City, State, Zip	2.00	Proc
8	55.00	2024-04-17	8	David Lee	012 Pine St, City, State, Zip	Ivy Martin	567 Pineapple St, City, State, Zip	2.80	Delv
9	40.00	2024-04-18	9	Eva Green	345 Cedar St, City, State, Zip	Jack Brown	400 Pine St, City, State, Zip	3.90	Can
10	1200.00	2024-04-19	10	Frank White	400 Pine St, City, State, Zip	Alice Johnson	123 Main St, City, State, Zip	4.50	In T
11	75.00	2024-04-20	11	Grace Taylor	901 Birch St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.70	Proc

30. Get Payment Details with Location

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, and user. The 'location' table has columns: LocationID, LocationName, and Address. The 'payment' table has columns: PaymentID, Amount, PaymentDate, CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, and Status. The 'courier' table has columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, and DeliveryDate. The 'employee' table has columns: EmployeeID, Name, Address, and Status. The 'courierservices' table has columns: ServiceID, Name, Address, and Status. The 'user' table has columns: UserID, Name, Address, and Status. The 'sys' table has columns: SysID, Name, Address, and Status. The 'Administration' table has columns: AdminID, Name, Address, and Status. The 'Information' table has columns: InfoID, Name, Address, and Status. The 'Schemas' table has columns: SchemaID, Name, Address, and Status. The 'Tables' table has columns: TableID, Name, Address, and Status. The 'Views' table has columns: ViewID, Name, Address, and Status. The 'Stored Procedures' table has columns: SPID, Name, Address, and Status. The 'Functions' table has columns: FID, Name, Address, and Status. The 'Triggers' table has columns: TrID, Name, Address, and Status. The 'Indexes' table has columns: IndID, Name, Address, and Status. The 'Foreign Keys' table has columns: FKID, Name, Address, and Status. The 'Fields' table has columns: FieldID, Name, Address, and Status. The 'Types' table has columns:TypeID, Name, Address, and Status. The 'Query' pane shows the following SQL query:

```
1 SELECT p.PaymentID, p.Amount, p.PaymentDate,
2       c.CourierID, c.SenderName, c.SenderAddress,
3       c.ReceiverName, c.ReceiverAddress, c.Weight,
4       c.Status, c.TrackingNumber, c.DeliveryDate,
5       l.LocationID, l.LocationName, l.Address AS LocationAddress
6 FROM Payment p
7 JOIN Courier c ON p.CourierID = c.CourierID
8 JOIN Location l ON p.LocationID = l.LocationID;
```

The 'Result Grid' shows the following data:

PaymentID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status
1	4000.00	2024-04-10	1	Alice Johnson	123 Main St, City, State, Zip	Bob Smith	400 Pine St, City, State, Zip	2.50	Prc
5	30.00	2024-04-14	5	Alice Johnson	123 Main St, City, State, Zip	Frank White	400 Pine St, City, State, Zip	5.10	Prc
9	40.00	2024-04-18	9	Eva Green	345 Cedar St, City, State, Zip	Jack Brown	400 Pine St, City, State, Zip	3.90	Ca
13	100.00	2024-04-22	13	Ivy Martin	567 Pineapple St, City, State, Zip	David Lee	012 Pine St, City, State, Zip	1.90	Prc
2	1400.00	2024-04-11	2	Alice Johnson	123 Main St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.00	De
6	70.00	2024-04-15	6	Bob Smith	456 Elm St, City, State, Zip	Grace Taylor	901 Birch St, City, State, Zip	3.50	De
10	1200.00	2024-04-19	10	Frank White	400 Pine St, City, State, Zip	Alice Johnson	123 Main St, City, State, Zip	4.50	In
14	95.00	2024-04-23	14	Jack Brown	890 Orange St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.30	In
3	300.00	2024-04-12	3	Alice Johnson	123 Main St, City, State, Zip	David Lee	400 Pine St, City, State, Zip	1.80	Ca
7	1400.00	2024-04-16	7	Charlie Brown	789 Oak St, City, State, Zip	Hannah Davis	234 Walnut St, City, State, Zip	2.00	Prc
11	75.00	2024-04-20	11	Grace Taylor	901 Birch St, City, State, Zip	Bob Smith	456 Elm St, City, State, Zip	2.70	Prc

31. Calculating Total Payments for Each Courier

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, and user. The right pane shows a query window with the following SQL code:

```
1 SELECT c.CourierID, c.SenderName,  
2 SUM(p.Amount) AS TotalPayments  
3 FROM Courier c  
4 LEFT JOIN Payment p ON c.CourierID = p.CourierID  
5 GROUP BY c.CourierID, c.SenderName  
6 ORDER BY TotalPayments DESC;  
7  
8  
9
```

Below the query, the 'Result Grid' displays the results of the query:

CourierID	SenderName	TotalPayments
1	Alice Johnson	4000.00
2	Alice Johnson	1400.00
7	Charlie Brown	1400.00
10	Frank White	1200.00
3	Alice Johnson	300.00
15	Alice Johnson	110.00
13	Ivy Martin	100.00
14	Jack Brown	95.00
12	Hannah Davis	85.00
11	Grace Taylor	75.00
6	Bob Smith	70.00
8	David Lee	55.00

32. List Payments Within a Date Range

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, and user. The right pane shows a query window with the following SQL code:

```
1 SELECT *  
2 FROM Payment  
3 WHERE PaymentDate BETWEEN '2024-04-01' AND '2024-04-13';  
4  
5  
6  
7  
8  
9
```

Below the query, the 'Result Grid' displays the results of the query:

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	1	4000.00	2024-04-10
2	2	2	1400.00	2024-04-11
3	3	3	300.00	2024-04-12
4	4	4	40.00	2024-04-13
5	5	5	40.00	2024-04-13

33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side

The screenshot shows a database management tool interface. On the left is a 'SCHEMAS' pane with a tree view of tables including 'courier', 'courierservices', 'employee', 'location', 'payment', and 'user'. The 'user' table is selected, showing columns like 'UserID', 'Name', 'Email', 'Password', 'ContactNumber', and 'Address'. The main pane displays a SQL query:

```

1 SELECT u.UserID, u.Name AS UserName,
2       c.CourierID, c.SenderName, c.SenderAddress,
3       c.ReceiverName, c.ReceiverAddress, c.Weight,
4       c.Status, c.TrackingNumber, c.DeliveryDate
5 FROM User u
6 LEFT JOIN Courier c ON u.name = c.sendername
7 ORDER BY u.UserID, c.CourierID;

```

Below the query is a 'Result Grid' showing 28 rows of data. The columns are: UserID, UserName, CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, and TrackingNumber. The results show a left join where all users are listed, and couriers are listed only if they match the user's name.

UserID	UserName	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber
1	Alice Johnson	1	Alice Johnson	123 Main St, City, State, Zip	Bob Smith	400 Pine St, City, State, Zip	2.50	Processing	TRAC
1	Alice Johnson	2	Alice Johnson	123 Main St, City, State, Zip	Charlie Brown	789 Oak St, City, State, Zip	3.00	Delivered	TRAC
1	Alice Johnson	3	Alice Johnson	123 Main St, City, State, Zip	David Lee	400 Pine St, City, State, Zip	1.80	Cancelled	TRAC
1	Alice Johnson	4	Alice Johnson	123 Main St, City, State, Zip	Eva Green	345 Cedar St, City, State, Zip	4.20	In Transit	TRAC
1	Alice Johnson	5	Alice Johnson	123 Main St, City, State, Zip	Frank White	400 Pine St, City, State, Zip	5.10	Processing	TRAC
1	Alice Johnson	15	Alice Johnson	123 Main St, City, State, Zip	Frank White	678 Maple St, City, State, Zip	5.00	Delivered	TRAC
2	Bob Smith	6	Bob Smith	456 Elm St, City, State, Zip	Grace Taylor	901 Birch St, City, State, Zip	3.50	Delivered	TRAC
3	Charlie Brown	7	Charlie Brown	789 Oak St, City, State, Zip	Hannah Davis	234 Walnut St, City, State, Zip	2.00	Processing	TRAC
4	David Lee	8	David Lee	012 Pine St, City, State, Zip	Ivy Martin	567 Pineapple St, City, State, Zip	2.80	Delivered	TRAC
5	Eva Green	9	Eva Green	345 Cedar St, City, State, Zip	Jack Brown	400 Pine St, City, State, Zip	3.90	Cancelled	TRAC
6	Frank White	10	Frank White	400 Pine St, City, State, Zip	Alice Johnson	123 Main St, City, State, Zip	4.50	In Transit	TRAC

34. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side

The screenshot shows the same database management tool interface. The 'SCHEMAS' pane shows the 'courierservices' table selected. The main pane displays a SQL query:

```

1 SELECT c.CourierID, c.SenderName,
2       cs.ServiceID, cs.ServiceName, cs.Cost
3 FROM Courier c
4 join payment p on p.courierid = c.courierid
5 left JOIN CourierServices cs ON amount = cs.cost
6 ORDER BY c.CourierID, cs.ServiceID;

```

Below the query is a 'Result Grid' showing 12 rows of data. The columns are: CourierID, SenderName, ServiceID, ServiceName, and Cost. The results show a left join where all couriers are listed, and services are listed only if they match the courier's payment amount.

CourierID	SenderName	ServiceID	ServiceName	Cost
1	Alice Johnson	NULL	NULL	NULL
2	Alice Johnson	NULL	NULL	NULL
3	Alice Johnson	NULL	NULL	NULL
4	Alice Johnson	NULL	NULL	NULL
5	Alice Johnson	3	Overnight	30.00
6	Bob Smith	NULL	NULL	NULL
7	Charlie Brown	NULL	NULL	NULL
8	David Lee	NULL	NULL	NULL
9	Eva Green	NULL	NULL	NULL
10	Frank White	NULL	NULL	NULL
11	Grace Taylor	NULL	NULL	NULL
12	Hannah Davis	NULL	NULL	NULL

36. List all users and all courier services, showing all possible combinations.

The screenshot shows a database management interface with a 'Navigator' pane on the left and a 'Query 1' editor on the right. The 'Navigator' pane shows a schema named 'hexa' with tables 'courier', 'courierservices', 'employee', and 'location'. The 'Query 1' editor contains the following SQL query:

```

1 SELECT u.UserID, u.Name AS UserName,
2       cs.ServiceID, cs.ServiceName, cs.Cost
3 FROM User u
4 CROSS JOIN CourierServices cs
5 ORDER BY u.UserID, cs.ServiceID;
6

```

The 'Result Grid' shows the following data:

UserID	UserName	ServiceID	ServiceName	Cost
1	Alice Johnson	1	Standard	10.00
1	Alice Johnson	2	Express	20.00
1	Alice Johnson	3	Overnight	30.00
1	Alice Johnson	4	International	50.00
2	Bob Smith	1	Standard	10.00
2	Bob Smith	2	Express	20.00
2	Bob Smith	3	Overnight	30.00
2	Bob Smith	4	International	50.00
3	Charlie Brown	1	Standard	10.00
3	Charlie Brown	2	Express	20.00
3	Charlie Brown	3	Overnight	30.00
3	Charlie Brown	4	International	50.00

37. List all employees and all locations, showing all possible combinations:

The screenshot shows a database management interface with a 'Navigator' pane on the left and a 'Query 1' editor on the right. The 'Navigator' pane shows a schema named 'hexa' with tables 'courier', 'courierservices', 'employee', and 'location'. The 'Query 1' editor contains the following SQL query:

```

1 SELECT e.EmployeeID, e.Name AS EmployeeName,
2       l.LocationID, l.LocationName, l.Address
3 FROM Employee e
4 CROSS JOIN Location l
5 ORDER BY e.EmployeeID, l.LocationID;
6
7

```

The 'Result Grid' shows the following data:

EmployeeID	EmployeeName	LocationID	LocationName	Address
1	John Doe	1	City Center	100 Main St, City, State, Zip
1	John Doe	2	East End	200 Elm St, City, State, Zip
1	John Doe	3	West Side	300 Oak St, City, State, Zip
1	John Doe	4	North Park	400 Pine St, City, State, Zip
2	Jane Smith	1	City Center	100 Main St, City, State, Zip
2	Jane Smith	2	East End	200 Elm St, City, State, Zip
2	Jane Smith	3	West Side	300 Oak St, City, State, Zip
2	Jane Smith	4	North Park	400 Pine St, City, State, Zip
3	Emily Johnson	1	City Center	100 Main St, City, State, Zip
3	Emily Johnson	2	East End	200 Elm St, City, State, Zip
3	Emily Johnson	3	West Side	300 Oak St, City, State, Zip
3	Emily Johnson	4	North Park	400 Pine St, City, State, Zip

38. Retrieve a list of couriers and their corresponding sender information (if available)

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'courier', 'courierservices', 'employee', 'location', 'payment', and 'user'. The 'user' table is selected, showing its columns: UserID, Name, Email, Password, ContactNumber, and Address. The main query editor displays the following SQL query:

```
1 • SELECT c.CourierID, c.SenderName, c.SenderAddress,  
2       u.UserID AS SenderUserID, u.Name AS SenderName,  
3       u.Email AS SenderEmail, u.ContactNumber AS SenderContact  
4 FROM Courier c  
5 LEFT JOIN User u ON c.SenderName = u.Name  
6 ORDER BY c.CourierID;  
7  
8  
9
```

The 'Result Grid' shows the results of the query, displaying columns: CourierID, SenderName, SenderAddress, SenderUserID, SenderName, SenderEmail, and SenderContact. The results are as follows:

CourierID	SenderName	SenderAddress	SenderUserID	SenderName	SenderEmail	SenderContact
1	Alice Johnson	123 Main St, City, State, Zip	1	Alice Johnson	alice@example.com	123-456-7890
2	Alice Johnson	123 Main St, City, State, Zip	1	Alice Johnson	alice@example.com	123-456-7890
3	Alice Johnson	123 Main St, City, State, Zip	1	Alice Johnson	alice@example.com	123-456-7890
4	Alice Johnson	123 Main St, City, State, Zip	1	Alice Johnson	alice@example.com	123-456-7890
5	Alice Johnson	123 Main St, City, State, Zip	1	Alice Johnson	alice@example.com	123-456-7890
6	Bob Smith	456 Elm St, City, State, Zip	2	Bob Smith	bob@example.com	234-567-8901
7	Charlie Brown	789 Oak St, City, State, Zip	3	Charlie Brown	charlie@example.com	345-678-9012
8	David Lee	012 Pine St, City, State, Zip	4	David Lee	david@example.com	456-789-0123
9	Eva Green	345 Cedar St, City, State, Zip	5	Eva Green	eva@example.com	567-890-1234
10	Frank White	400 Pine St, City, State, Zip	6	Frank White	frank@example.com	678-901-2345
11	Grace Taylor	901 Birch St, City, State, Zip	7	Grace Taylor	grace@example.com	789-012-3456
12	Hannah Davis	234 Walnut St, City, State, Zip	8	Hannah Davis	hannah@example.com	890-123-4567

39. Retrieve a list of couriers and their corresponding receiver information (if available):

The screenshot shows the same database management tool interface. The 'SCHEMAS' pane is the same. The main query editor displays the following SQL query:

```
1 • SELECT c.CourierID, c.ReceiverName, c.ReceiverAddress,  
2       u.UserID AS ReceiverUserID, u.Name AS ReceiverName,  
3       u.Email AS ReceiverEmail, u.ContactNumber AS ReceiverContact  
4 FROM Courier c  
5 LEFT JOIN User u ON c.ReceiverName = u.Name  
6 ORDER BY c.CourierID;  
7  
8  
9
```

The 'Result Grid' shows the results of the query, displaying columns: CourierID, ReceiverName, ReceiverAddress, ReceiverUserID, ReceiverName, ReceiverEmail, and ReceiverContact. The results are as follows:

CourierID	ReceiverName	ReceiverAddress	ReceiverUserID	ReceiverName	ReceiverEmail	ReceiverContact
1	Bob Smith	400 Pine St, City, State, Zip	2	Bob Smith	bob@example.com	234-567-8901
2	Charlie Brown	789 Oak St, City, State, Zip	3	Charlie Brown	charlie@example.com	345-678-9012
3	David Lee	400 Pine St, City, State, Zip	4	David Lee	david@example.com	456-789-0123
4	Eva Green	345 Cedar St, City, State, Zip	5	Eva Green	eva@example.com	567-890-1234
5	Frank White	400 Pine St, City, State, Zip	6	Frank White	frank@example.com	678-901-2345
6	Grace Taylor	901 Birch St, City, State, Zip	7	Grace Taylor	grace@example.com	789-012-3456
7	Hannah Davis	234 Walnut St, City, State, Zip	8	Hannah Davis	hannah@example.com	890-123-4567
8	Ivy Martin	567 Pineapple St, City, State, Zip	9	Ivy Martin	ivy@example.com	901-234-5678
9	Jack Brown	400 Pine St, City, State, Zip	10	Jack Brown	jack@example.com	012-345-6789
10	Alice Johnson	123 Main St, City, State, Zip	1	Alice Johnson	alice@example.com	123-456-7890
11	Bob Smith	456 Elm St, City, State, Zip	2	Bob Smith	bob@example.com	234-567-8901
12	Charlie Brown	789 Oak St, City, State, Zip	3	Charlie Brown	charlie@example.com	345-678-9012

42. Retrieve a list of locations and the total payment amount received at each location:

The screenshot shows a database management tool interface. On the left is a 'SCHEMAS' pane with a tree view showing tables like 'courier', 'courierservices', 'employee', 'location', 'payment', and 'user'. The 'location' table is selected. The main area displays 'Query 1' with the following SQL code:

```

1 SELECT l.LocationID, l.LocationName, l.Address,
2       SUM(p.Amount) AS TotalPaymentAmount
3 FROM Location l
4 LEFT JOIN Payment p ON l.LocationID = p.LocationID
5 GROUP BY l.LocationID, l.LocationName, l.Address
6 ORDER BY l.LocationID;
7

```

Below the query, the 'Result Grid' shows the results of the query:

LocationID	LocationName	Address	TotalPaymentAmount
1	City Center	100 Main St, City, State, Zip	4170.00
2	East End	200 Elm St, City, State, Zip	2765.00
3	West Side	300 Oak St, City, State, Zip	1885.00
4	North Park	400 Pine St, City, State, Zip	180.00

43. Retrieve all couriers sent by the same sender (based on SenderName).

The screenshot shows the same database management tool interface. The 'SCHEMAS' pane is the same. The main area displays 'Query 1' with the following SQL code:

```

1 SELECT c1.CourierID AS CourierID1, c1.SenderName AS SenderName,
2       c2.CourierID AS CourierID2, c2.SenderName AS SenderName,
3       c2.ReceiverName, c2.ReceiverAddress
4 FROM Courier c1
5 LEFT JOIN Courier c2 ON c1.SenderName = c2.SenderName AND c1.CourierID <> c2.CourierID
6 ORDER BY c1.SenderName, c1.CourierID, c2.CourierID;
7
8

```

Below the query, the 'Result Grid' shows the results of the query:

CourierID1	SenderName	CourierID2	SenderName	ReceiverName	ReceiverAddress
1	Alice Johnson	2	Alice Johnson	Charlie Brown	789 Oak St, City, State, Zip
1	Alice Johnson	3	Alice Johnson	David Lee	400 Pine St, City, State, Zip
1	Alice Johnson	4	Alice Johnson	Eva Green	345 Cedar St, City, State, Zip
1	Alice Johnson	5	Alice Johnson	Frank White	400 Pine St, City, State, Zip
1	Alice Johnson	15	Alice Johnson	Frank White	678 Maple St, City, State, Zip
1	Alice Johnson	1	Alice Johnson	Bob Smith	400 Pine St, City, State, Zip
2	Alice Johnson	3	Alice Johnson	David Lee	400 Pine St, City, State, Zip
2	Alice Johnson	4	Alice Johnson	Eva Green	345 Cedar St, City, State, Zip
2	Alice Johnson	5	Alice Johnson	Frank White	400 Pine St, City, State, Zip
2	Alice Johnson	15	Alice Johnson	Frank White	678 Maple St, City, State, Zip
3	Alice Johnson	1	Alice Johnson	Bob Smith	400 Pine St, City, State, Zip
3	Alice Johnson	2	Alice Johnson	Charlie Brown	789 Oak St, City, State, Zip

44. List all employees who share the same role.

The screenshot shows a database management tool interface. On the left is a 'SCHEMAS' tree with a tree view of the database structure. The main area displays a SQL query in 'Query 1' and its results in a 'Result Grid'.

Query 1:

```

1 SELECT e1.EmployeeID AS EmployeeID1, e1.Name AS EmployeeName1, e1.Role AS SharedRole,
2       e2.EmployeeID AS EmployeeID2, e2.Name AS EmployeeName2
3 FROM Employee e1
4 LEFT JOIN Employee e2 ON e1.Role = e2.Role AND e1.EmployeeID <> e2.EmployeeID
5 ORDER BY e1.Role, e1.EmployeeID, e2.EmployeeID;

```

Result Grid:

EmployeeID1	EmployeeName1	SharedRole	EmployeeID2	EmployeeName2
5	Sophia Wilson	Admin	10	James Jones
10	James Jones	Admin	5	Sophia Wilson
3	Emily Johnson	Customer Service	8	William Wilson
3	Emily Johnson	Customer Service	13	Mia Thompson
8	William Wilson	Customer Service	3	Emily Johnson
8	William Wilson	Customer Service	13	Mia Thompson
13	Mia Thompson	Customer Service	3	Emily Johnson
13	Mia Thompson	Customer Service	8	William Wilson
2	Jane Smith	Delivery Staff	7	Emma Davis
2	Jane Smith	Delivery Staff	12	Aiden Davis
7	Emma Davis	Delivery Staff	2	Jane Smith
7	Emma Davis	Delivery Staff	12	Aiden Davis

45. Retrieve all payments made for couriers sent from the same location.

The screenshot shows a database management tool interface. On the left is a 'SCHEMAS' tree. The main area displays a SQL query in 'Query 1' and its results in a 'Result Grid'.

Query 1:

```

2 p1.PaymentID AS PaymentID1, p1.Amount AS PaymentAmount1,
3 c2.CourierID AS CourierID2, c2.SenderName, c2.SenderAddress,
4 p2.PaymentID AS PaymentID2, p2.Amount AS PaymentAmount2
5 FROM Courier c1
6 LEFT JOIN Payment p1 ON c1.CourierID = p1.CourierID
7 LEFT JOIN Courier c2 ON c1.SenderAddress = c2.SenderAddress AND c1.CourierID <> c2.CourierID
8 LEFT JOIN Payment p2 ON c2.CourierID = p2.CourierID
9 ORDER BY c1.SenderAddress, c1.CourierID, c2.CourierID;
10

```

Result Grid:

CourierID1	SenderName	SenderAddress	PaymentID1	PaymentAmount1	CourierID2	SenderName	SenderAddress	PaymentID2
8	David Lee	012 Pine St, City, State, Zip	8	55.00	NULL	NULL	NULL	NULL
1	Alice Johnson	123 Main St, City, State, Zip	1	4000.00	2	Alice Johnson	123 Main St, City, State, Zip	2
1	Alice Johnson	123 Main St, City, State, Zip	1	4000.00	3	Alice Johnson	123 Main St, City, State, Zip	3
1	Alice Johnson	123 Main St, City, State, Zip	1	4000.00	4	Alice Johnson	123 Main St, City, State, Zip	4
1	Alice Johnson	123 Main St, City, State, Zip	1	4000.00	5	Alice Johnson	123 Main St, City, State, Zip	5
1	Alice Johnson	123 Main St, City, State, Zip	1	4000.00	15	Alice Johnson	123 Main St, City, State, Zip	15
2	Alice Johnson	123 Main St, City, State, Zip	2	1400.00	1	Alice Johnson	123 Main St, City, State, Zip	1
2	Alice Johnson	123 Main St, City, State, Zip	2	1400.00	3	Alice Johnson	123 Main St, City, State, Zip	3
2	Alice Johnson	123 Main St, City, State, Zip	2	1400.00	4	Alice Johnson	123 Main St, City, State, Zip	4
2	Alice Johnson	123 Main St, City, State, Zip	2	1400.00	5	Alice Johnson	123 Main St, City, State, Zip	5
2	Alice Johnson	123 Main St, City, State, Zip	2	1400.00	15	Alice Johnson	123 Main St, City, State, Zip	15

46. Retrieve all couriers sent from the same location (based on SenderAddress).

Query 1

```

1 SELECT c1.CourierID AS CourierID1, c1.SenderName, c1.SenderAddress,
2       c2.CourierID AS CourierID2, c2.SenderName, c2.ReceiverName, c2.ReceiverAddress
3 FROM Courier c1
4 LEFT JOIN Courier c2 ON c1.SenderAddress = c2.SenderAddress AND c1.CourierID <> c2.CourierID
5 ORDER BY c1.SenderAddress, c1.CourierID, c2.CourierID;

```

CourierID1	SenderName	SenderAddress	CourierID2	SenderName	ReceiverName	ReceiverAddress
8	David Lee	012 Pine St, City, State, Zip	2	Alice Johnson	Charlie Brown	789 Oak St, City, State, Zip
1	Alice Johnson	123 Main St, City, State, Zip	3	Alice Johnson	David Lee	400 Pine St, City, State, Zip
1	Alice Johnson	123 Main St, City, State, Zip	4	Alice Johnson	Eva Green	345 Cedar St, City, State, Zip
1	Alice Johnson	123 Main St, City, State, Zip	5	Alice Johnson	Frank White	400 Pine St, City, State, Zip
1	Alice Johnson	123 Main St, City, State, Zip	15	Alice Johnson	Frank White	678 Maple St, City, State, Zip
2	Alice Johnson	123 Main St, City, State, Zip	1	Alice Johnson	Bob Smith	400 Pine St, City, State, Zip
2	Alice Johnson	123 Main St, City, State, Zip	3	Alice Johnson	David Lee	400 Pine St, City, State, Zip
2	Alice Johnson	123 Main St, City, State, Zip	4	Alice Johnson	Eva Green	345 Cedar St, City, State, Zip
2	Alice Johnson	123 Main St, City, State, Zip	5	Alice Johnson	Frank White	400 Pine St, City, State, Zip
2	Alice Johnson	123 Main St, City, State, Zip	15	Alice Johnson	Frank White	678 Maple St, City, State, Zip
3	Alice Johnson	123 Main St, City, State, Zip	1	Alice Johnson	Bob Smith	400 Pine St, City, State, Zip

49. Find couriers that have a weight greater than the average weight of all couriers

Query 1

```

1 SELECT CourierID, SenderName, ReceiverName, Weight
2 FROM Courier
3 WHERE Weight > (
4     SELECT AVG(Weight)
5     FROM Courier
6 )
7 ORDER BY CourierID;

```

CourierID	SenderName	ReceiverName	Weight
4	Alice Johnson	Eva Green	4.20
5	Alice Johnson	Frank White	5.10
6	Bob Smith	Grace Taylor	3.50
9	Eva Green	Jack Brown	3.90
10	Frank White	Alice Johnson	4.50
12	Hannah Davis	Charlie Brown	3.40
14	Jack Brown	Eva Green	4.30
15	Alice Johnson	Frank White	5.00

Scope: Inner Queries, Non Equi Joins, Equi joins, Exist, Any, All

50. Find the names of all employees who have a salary greater than the average salary:

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, and user. The 'employee' table is selected. The right pane shows a query window with the following SQL code:

```
1 SELECT CourierID, SenderName, ReceiverName, Weight
2 FROM Courier
3 WHERE Weight > (
4     SELECT AVG(Weight)
5     FROM Courier
6 )
7 ORDER BY CourierID;
```

The 'Result Grid' shows the following data:

CourierID	SenderName	ReceiverName	Weight
4	Alice Johnson	Eva Green	4.20
5	Alice Johnson	Frank White	5.10
6	Bob Smith	Grace Taylor	3.50
9	Eva Green	Jack Brown	3.90
10	Frank White	Alice Johnson	4.50
12	Hannah Davis	Charlie Brown	3.40
14	Jack Brown	Eva Green	4.30
15	Alice Johnson	Frank White	5.00
NULL	NULL	NULL	NULL

51. Find the total cost of all courier services where the cost is less than the maximum cost

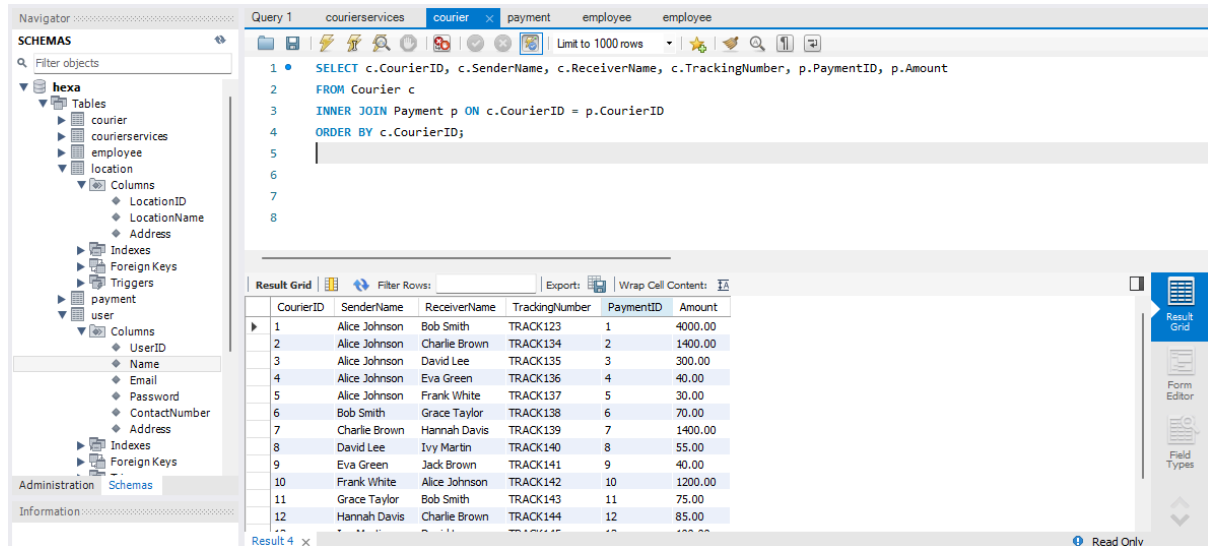
The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'hexa' database schema with tables: courier, courierservices, employee, location, payment, and user. The 'courierservices' table is selected. The right pane shows a query window with the following SQL code:

```
1 SELECT SUM(Cost) AS TotalCost
2 FROM CourierServices
3 WHERE Cost < (
4     SELECT MAX(Cost)
5     FROM CourierServices
6 );
```

The 'Result Grid' shows the following data:

TotalCost
60.00

52. Find all couriers that have been paid for



Query 1 courierservices courier payment employee employee

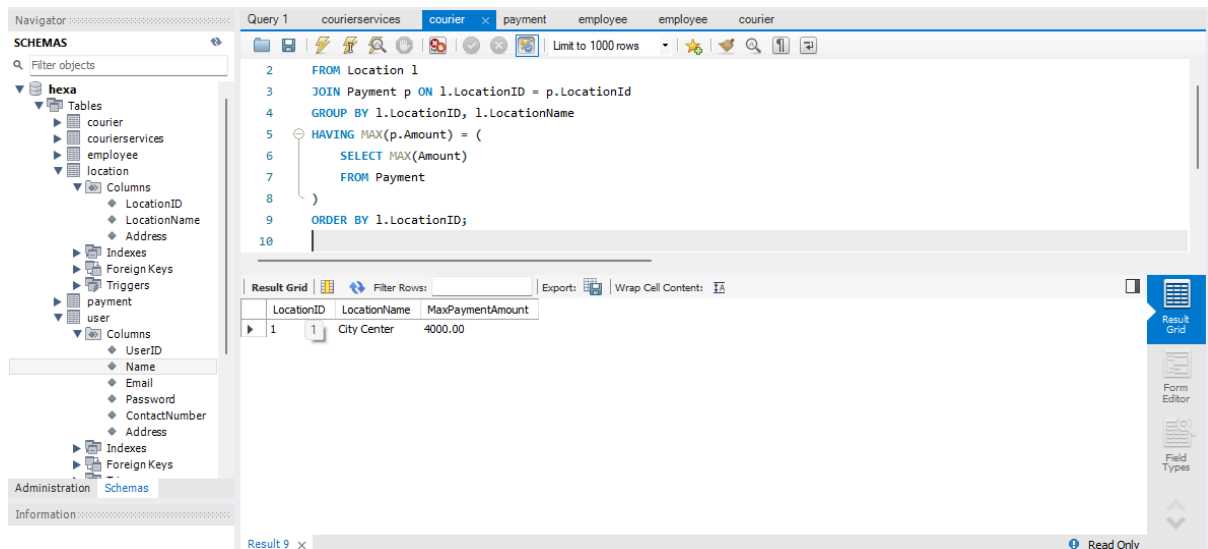
```

1 • SELECT c.CourierID, c.SenderName, c.ReceiverName, c.TrackingNumber, p.PaymentID, p.Amount
2 FROM Courier c
3 INNER JOIN Payment p ON c.CourierID = p.CourierID
4 ORDER BY c.CourierID;
5
6
7
8

```

CourierID	SenderName	ReceiverName	TrackingNumber	PaymentID	Amount
1	Alice Johnson	Bob Smith	TRACK123	1	4000.00
2	Alice Johnson	Charlie Brown	TRACK134	2	1400.00
3	Alice Johnson	David Lee	TRACK135	3	300.00
4	Alice Johnson	Eva Green	TRACK136	4	40.00
5	Alice Johnson	Frank White	TRACK137	5	30.00
6	Bob Smith	Grace Taylor	TRACK138	6	70.00
7	Charlie Brown	Hannah Davis	TRACK139	7	1400.00
8	David Lee	Ivy Martin	TRACK140	8	55.00
9	Eva Green	Jack Brown	TRACK141	9	40.00
10	Frank White	Alice Johnson	TRACK142	10	1200.00
11	Grace Taylor	Bob Smith	TRACK143	11	75.00
12	Hannah Davis	Charlie Brown	TRACK144	12	85.00

53. Find the locations where the maximum payment amount was made



Query 1 courierservices courier payment employee employee courier

```

2 FROM Location l
3 JOIN Payment p ON l.LocationID = p.LocationID
4 GROUP BY l.LocationID, l.LocationName
5 HAVING MAX(p.Amount) = (
6 SELECT MAX(Amount)
7 FROM Payment
8 )
9 ORDER BY l.LocationID;
10

```

LocationID	LocationName	MaxPaymentAmount
1	City Center	4000.00

54. Find all couriers whose weight is greater than the weight of all couriers sent by a specific sender (e.g., 'SenderName'):

The screenshot shows a database management tool interface. On the left is a 'SCHEMAS' tree view showing a database named 'hexa' with tables: courier, courierservices, employee, location, payment, and user. The 'courier' table is selected. The main area displays a SQL query in 'Query 1':

```
1
2 SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight
3 FROM Courier c
4 WHERE c.Weight > (
5     SELECT MAX(Weight)
6     FROM Courier
7     WHERE SenderName = 'Charlie Brown'
8 )
9 ORDER BY c.CourierID;
```

Below the query is a 'Result Grid' showing 15 rows of data. The columns are CourierID, SenderName, ReceiverName, and Weight. The results are ordered by CourierID.

CourierID	SenderName	ReceiverName	Weight
1	Alice Johnson	Bob Smith	2.50
2	Alice Johnson	Charlie Brown	3.00
4	Alice Johnson	Eva Green	4.20
5	Alice Johnson	Frank White	5.10
6	Bob Smith	Grace Taylor	3.50
8	David Lee	Ivy Martin	2.80
9	Eva Green	Jack Brown	3.90
10	Frank White	Alice Johnson	4.50
11	Grace Taylor	Bob Smith	2.70
12	Hannah Davis	Charlie Brown	3.40
14	Jack Brown	Eva Green	4.30
15	Alice Johnson	Frank White	5.00

At the bottom of the result grid, there is a tab labeled 'Courier 7' and buttons for 'Apply' and 'Revert'.