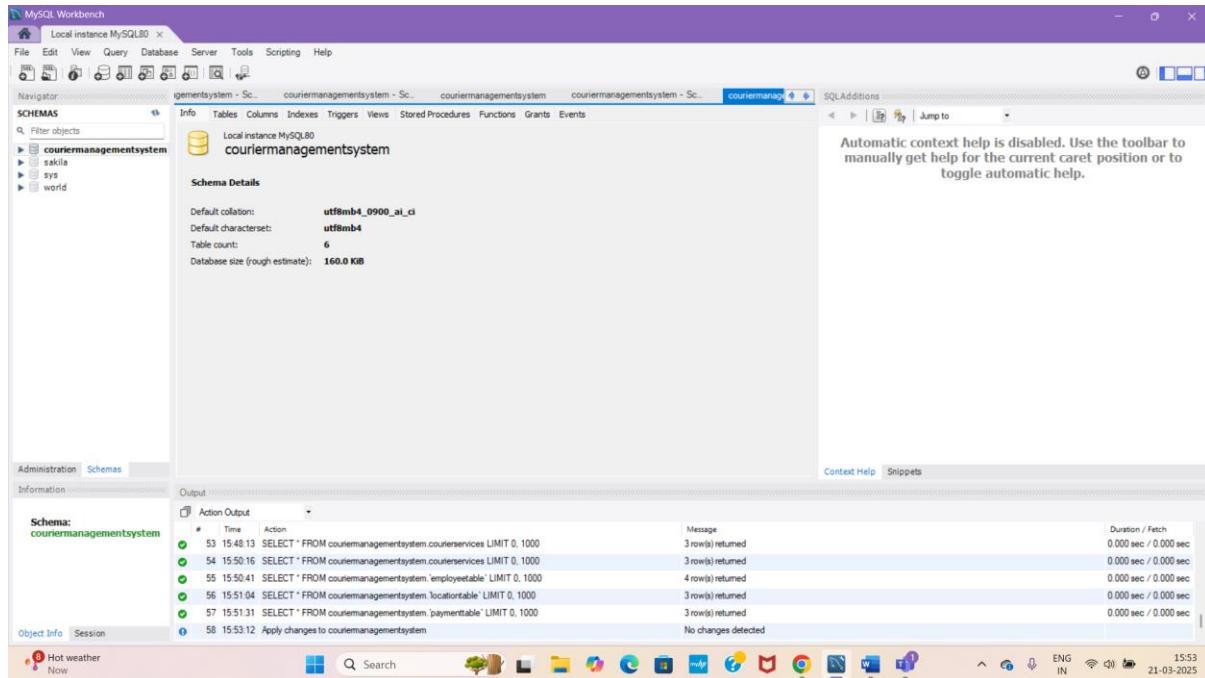


TASK 1:

CREATING A DATABASE:

CREATE SCHEMA `couriermanagementsystem` ;



CREATING TABLES:

User Table:

```
CREATE TABLE `user` (
    `UserID` int NOT NULL,
    `Name` varchar(255) DEFAULT NULL,
    `Email` varchar(255) DEFAULT NULL,
    `Password` varchar(255) DEFAULT NULL,
    `ContactNumber` varchar(20) DEFAULT NULL,
    `Address` text,
    PRIMARY KEY (`UserID`),
    UNIQUE KEY `Email_UNIQUE` (`Email`));
```

Courier Table:

```
CREATE TABLE `courier` (
    `CourierID` int NOT NULL,
    `SenderName` varchar(255) DEFAULT NULL,
```

```
`SenderAddress` text,  
 `ReceiverName` varchar(255) DEFAULT NULL,  
 `ReceiverAddress` text,  
 `Weight` decimal(5,2) DEFAULT NULL,  
 `Status` varchar(50) DEFAULT NULL,  
 `TrackingNumber` varchar(20) DEFAULT NULL,  
 `DeliveryDate` date DEFAULT NULL,  
 PRIMARY KEY (`CourierID`),  
 UNIQUE KEY `TrackingNumber_UNIQUE` (`TrackingNumber`));
```

CourierServices Table:

```
CREATE TABLE `courierservices` (  
 `ServicesId` int NOT NULL,  
 `ServiceName` varchar(100) DEFAULT NULL,  
 `Cost` decimal(8,2) DEFAULT NULL,  
 PRIMARY KEY (`ServicesId`));
```

Employee Table:

```
CREATE TABLE `employeetable` (  
 `EmployeeID` int NOT NULL,  
 `Name` varchar(255) DEFAULT NULL,  
 `Email` varchar(255) DEFAULT NULL,  
 `ContactNumber` varchar(20) DEFAULT NULL,  
 `Role` varchar(50) DEFAULT NULL,  
 `Salary` decimal(10,2) DEFAULT NULL,  
 PRIMARY KEY (`EmployeeID`));
```

Location Table:

```
CREATE TABLE `locationtable` (  
 `LocationID` int NOT NULL,  
 `LocationName` varchar(100) DEFAULT NULL,  
 `Address` text,
```

```
PRIMARY KEY (`LocationID`));
```

Payment Table:

```
CREATE TABLE `paymenttable` (
  `PaymentID` int NOT NULL,
  `CourierID` int DEFAULT NULL,
  `LocationID` int DEFAULT NULL,
  `Amount` decimal(10,2) DEFAULT NULL,
  `PaymentDate` date DEFAULT NULL,
  PRIMARY KEY (`PaymentID`),
  KEY `CourierID_idx` (`CourierID`),
  KEY `LocationID_idx` (`LocationID`),
  CONSTRAINT `CourierID` FOREIGN KEY (`CourierID`) REFERENCES `courier` (`CourierID`),
  CONSTRAINT `LocationID` FOREIGN KEY (`LocationID`) REFERENCES `location table` (`LocationID`);
```

INSERTING VALUES IN USER TABLE:

```
INSERT INTO `couriermanagementsystem`.`user` (`UserID`, `Name`, `Email`, `Password`, `ContactNumber`, `Address`) VALUES ('1', 'Nithya', 'nithya@gmail.com', '111', '1234567890', 'Saidapet');
```

```
INSERT INTO `couriermanagementsystem`.`user` (`UserID`, `Name`, `Email`, `Password`, `ContactNumber`, `Address`) VALUES ('2', 'Deepa', 'deepa@gmail.com', '222', '1234567890', 'MRC Nagar');
```

```
INSERT INTO `couriermanagementsystem`.`user` (`UserID`, `Name`, `Email`, `Password`, `ContactNumber`, `Address`) VALUES ('3', 'Lakshmi', 'lakshmi@gmail.com', '333', '1234567890', 'West Mambalam');
```

```
INSERT INTO `couriermanagementsystem`.`user` (`UserID`, `Name`, `Email`, `Password`, `ContactNumber`, `Address`) VALUES ('4', 'Hashitha', 'hashitha@gmail.com', '444', '1234567890', 'T.Nagar');
```

```
INSERT INTO `couriermanagementsystem`.`user` (`UserID`, `Name`, `Email`, `Password`, `ContactNumber`, `Address`) VALUES ('5', 'Swetha', 'swetha@gmail.com', '555', '1234567890', 'Teynampet');
```

```
INSERT INTO `couriermanagementsystem`.`user` (`UserID`, `Name`, `Email`, `Password`, `ContactNumber`, `Address`) VALUES ('6', 'Anand', 'anand@gmail.com', '666', '1234567890', 'Mylapore');
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'couriermanagementsystem' selected. The main area shows the 'user' table with the following data:

UserID	Name	Email	Password	ContactNumber	Address
1	Nithya	nithya@gmail.com	111	1234567890	Saidapet
2	Deepa	deepa@gmail.com	222	1234567890	MRC Nagar
3	Lakshmi	lakshmi@gmail.com	333	1234567890	West Mambalam
4	Hashima	hashimah@gmail.com	444	1234567890	T.Nagar
5	Sreetha	sreetha@gmail.com	555	1234567890	Chromepet
6	Anand	anand@gmail.com	666	1234567890	Mylapore

The bottom pane shows the history of actions taken on the session:

Action	Time	Action	Message	Duration / Fetch
48	15:44:47	SELECT c.CourierID, AVG(TIMESTAMPDIFF(HOUR, c.PickupTime, c.DeliveryTime)) AS AvgDeliveryTime FROM couriermanagementsystem.user c	Error Code: 1054. Unknown column 'c.PickupTime' in field list'	0.000 sec
49	15:46:48	INSERT INTO couriermanagementsystem.user ('Name', 'Email', 'Password', 'ContactNumber', 'Address')	1364: Field 'UserID' doesn't have a default value	
50	15:47:59	SELECT * FROM couriermanagementsystem.user LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
51	15:48:01	SELECT * FROM couriermanagementsystem.user LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
52	15:48:10	SELECT * FROM couriermanagementsystem.courier LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
53	15:48:13	SELECT * FROM couriermanagementsystem.courierservices LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

INSERTING VALUES IN COURIER TABLE:

```
INSERT INTO `couriermanagementsystem`.`courier` (`CourierID`, `SenderName`, `SenderAddress`, `ReceiverName`, `ReceiverAddress`, `Weight`, `Status`, `TrackingNumber`, `DeliveryDate`) VALUES ('1', 'Kani', 'OMR', 'Amirtha', 'Saidapet', '3', 'Delivered', 'TR1', '2025-03-01');
```

```
INSERT INTO `couriermanagementsystem`.`courier` (`CourierID`, `SenderName`, `SenderAddress`, `ReceiverName`, `ReceiverAddress`, `Weight`, `Status`, `TrackingNumber`, `DeliveryDate`) VALUES ('2', 'Gayathri', 'West Mambalam', 'Nisha', 'Ashok Nagar', '2.5', 'Delivered', 'TR2', '2025-03-10');
```

```
INSERT INTO `couriermanagementsystem`.`courier` (`CourierID`, `SenderName`, `SenderAddress`, `ReceiverName`, `ReceiverAddress`, `Weight`, `Status`, `TrackingNumber`, `DeliveryDate`) VALUES ('3', 'Monica', 'Pallavaram', 'Logitha', 'Chromepet', '5', 'Pending', 'TR3', '2025-03-31');
```

```
INSERT INTO `couriermanagementsystem`.`courier` (`CourierID`, `SenderName`, `SenderAddress`, `ReceiverName`, `ReceiverAddress`, `Weight`, `Status`, `TrackingNumber`, `DeliveryDate`) VALUES ('4', 'Sathvik', 'Porur', 'Pavika', 'Avadi', '1.50', 'Pending', 'TR4', '2025-03-21');
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'couriermanagementsystem' schema with its tables: courierservices, payment table, location table, employee table, user, and courier. The main pane shows a SQL editor with the following query:

```
1 • SELECT * FROM couriermanagementsystem.courierservices;
```

The results grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Kani	OMR	Amirtha	Sedapet	3.00	Delivered	TR 1	2025-03-01
2	Gayathri	West Mambalam	Nisha	Ashok Nagar	2.50	Delivered	TR 2	2025-03-10
3	Monica	Palavaram	Logitha	Chromepet	5.00	Pending	TR 3	2025-03-31
4	Sathvik	Porur	Pavika	Avadi	1.50	Pending	TR 4	2025-03-21

The status bar at the bottom right indicates: 91°F Sunny, ENG IN, 15:48, 21-03-2025.

INSERTING VALUES IN COURIERSERVICE TABLE:

```
INSERT INTO `couriermanagementsystem`.`courierservices` (`ServicesId`, `ServiceName`, `Cost`) VALUES ('1', 'Normal Delivery', '100');
```

```
INSERT INTO `couriermanagementsystem`.`courierservices` (`ServicesId`, `ServiceName`, `Cost`) VALUES ('2', 'Express Delivery', '500');
```

```
INSERT INTO `couriermanagementsystem`.`courierservices` (`ServicesId`, `ServiceName`, `Cost`) VALUES ('3', 'Safe Delivery', '800');
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'couriermanagementsystem' schema with its tables: courierservices, payment table, location table, employee table, user, and courier. The main pane shows a SQL editor with the following query:

```
1 • SELECT * FROM couriermanagementsystem.courierservices;
```

The results grid displays the following data:

ServicesId	ServiceName	Cost
1	Normal Delivery	100.00
2	Express Delivery	500.00
3	Safe Delivery	800.00

The status bar at the bottom right indicates: 91°F Sunny, ENG IN, 15:50, 21-03-2025.

INSERTING VALUES IN EMPLOYEE TABLE:

```
INSERT INTO `couriermanagementsystem`.`employeetable` (`EmployeeID`, `Name`, `Email`, `ContactNumber`, `Role`, `Salary`) VALUES ('1', 'Priyanka', 'priya@gmail.com', '1234567890', 'Manager', '50');
```

```
INSERT INTO `couriermanagementsystem`.`employeetable` (`EmployeeID`, `Name`, `Email`, `ContactNumber`, `Role`, `Salary`) VALUES ('2', 'Prasanna', 'prasanna@gmail.com', '1234567890', 'Dispatcher', '25');
```

```
INSERT INTO `couriermanagementsystem`.`employeetable` (`EmployeeID`, `Name`, `Email`, `ContactNumber`, `Role`, `Salary`) VALUES ('3', 'Dhanush', 'dhanush@gmail.com', '1234567890', 'Watchman', '10');
```

```
INSERT INTO `couriermanagementsystem`.`employeetable` (`EmployeeID`, `Name`, `Email`, `ContactNumber`, `Role`, `Salary`) VALUES ('4', 'John', 'john@gmail.com', '1234567890', 'Admin', '30.00');
```

The screenshot shows the MySQL Workbench interface. The top navigation bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the toolbar, the Navigator pane shows the schema 'couriermanagementsystem' containing tables like courierservices, payment table, location table, and employeetable. The employeetable tab is selected, displaying the following data:

EmployeeID	Name	Email	ContactNumber	Role	Salary
1	Priyanka	priya@gmail.com	1234567890	Manager	50.00
2	Prasanna	prasanna@gmail.com	1234567890	Dispatcher	25.00
3	Dhanush	dhanush@gmail.com	1234567890	Watchman	10.00
4	John	john@gmail.com	1234567890	Admin	30.00

The bottom pane shows the 'Action Output' section with a list of recent queries and their execution details. The system status bar at the bottom right indicates it's 15:50 on 21-03-2025, with an ENG IN language setting.

INSERTING VALUES IN LOCATION TABLE:

```
INSERT INTO `couriermanagementsystem`.`locationtable` (`LocationID`, `LocationName`, `Address`) VALUES ('1', 'Shanthi Colony', 'Anna Nagar');
```

```
INSERT INTO `couriermanagementsystem`.`locationtable` (`LocationID`, `LocationName`, `Address`) VALUES ('2', 'Jones Road', 'Saidapet');
```

```
INSERT INTO `couriermanagementsystem`.`locationtable` (`LocationID`, `LocationName`, `Address`) VALUES ('3', 'Cerus', 'Saligramam');
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to 'couriermanagementsystem'. The main window displays the results of a query: 'SELECT * FROM couriermanagementsystem.locationtable;'. The results grid shows three rows of data:

LocationID	LocationName	Address
1	Shenti Colony	Anna Nagar
2	Jones Road	Saidapet
3	Census	Saligramam

Below the results grid is the 'Information' panel, which shows the execution history for the current session. The history includes the following entries:

Action	Time	Message	Duration / Fetch
SELECT * FROM couriermanagementsystem.user	51 15:48:01	6 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.courier	52 15:48:10	4 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.courservices	53 15:48:13	3 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.courierservices	54 15:50:16	3 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.employee	55 15:50:41	4 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.locationtable	56 15:51:04	3 row(s) returned	0.000 sec / 0.000 sec

INSERTING VALUES IN PAYMENT TABLE:

```
INSERT INTO `couriermanagementsystem`.`paymenttable`(`PaymentID`, `CourierID`, `LocationID`, `Amount`, `PaymentDate`) VALUES ('1', '1', '1', '100', '2025-02-14');
```

```
INSERT INTO `couriermanagementsystem`.`paymenttable`(`PaymentID`, `CourierID`, `LocationID`, `Amount`, `PaymentDate`) VALUES ('2', '2', '2', '500', '2025-03-01');
```

```
INSERT INTO `couriermanagementsystem`.`paymenttable`(`PaymentID`, `CourierID`, `LocationID`, `Amount`, `PaymentDate`) VALUES ('3', '3', '3', '800', '2025-03-20');
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to 'couriermanagementsystem'. The main window displays the results of a query: 'SELECT * FROM couriermanagementsystem.paymenttable;'. The results grid shows three rows of data:

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	1	100.00	2025-02-14
2	2	2	500.00	2025-03-01
3	3	3	800.00	2025-03-20

Below the results grid is the 'Information' panel, which shows the execution history for the current session. The history includes the following entries:

Action	Time	Message	Duration / Fetch
SELECT * FROM couriermanagementsystem.courier	52 15:48:10	4 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.courservices	53 15:48:13	3 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.courierservices	54 15:50:16	3 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.employee	55 15:50:41	4 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.locationtable	56 15:51:04	3 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM couriermanagementsystem.paymenttable	57 15:51:31	3 row(s) returned	0.000 sec / 0.000 sec

TASK 2:

1. List all customers: `SELECT * FROM User;`

The screenshot shows the MySQL Workbench interface. In the SQL editor, the query `SELECT * FROM User;` is run. The results are displayed in a grid:

User ID	Name	Email	Password	Contact Number	Address
1	Nithya	nithya@gmail.com	111	1234567890	Saidapet
2	Deepa	deepa@gmail.com	222	1234567890	MRC Nagar
3	Lakshmi	lakshmi@gmail.com	333	1234567890	West Mambalam
4	Hashitha	hashitha@gmail.com	444	1234567890	T.Nagar
5	Swetha	swetha@gmail.com	555	1234567890	Teynampet

The status bar at the bottom right shows the date and time as 21-03-2025 01:16.

2. List all orders for a specific customer: `SELECT * FROM User WHERE UserID = 3;`

The screenshot shows the MySQL Workbench interface. In the SQL editor, the query `SELECT * FROM User WHERE UserID = 3;` is run. The results are displayed in a grid:

User ID	Name	Email	Password	Contact Number	Address
3	Lakshmi	lakshmi@gmail.com	333	1234567890	West Mambalam

The status bar at the bottom right shows the date and time as 21-03-2025 01:35.

3. List all couriers: SELECT * FROM Courier;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema 'couriermanagementsystem' with tables: courier, courierservices, employeetable, locationtable, paymenttable, and user.
- SQL Editor:** Contains the query: `1 • SELECT * FROM Courier;`
- Result Grid:** Displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Kanika	OMR	Amrits	Saidapet	3.00	Delivered	TR1	2025-03-01
2	Gayathri	West Mambalam	Nisha	Ashok Nagar	2.50	Delivered	TR2	2025-03-10
3	Monica	Pallavaram	Logitha	Chromepet	5.00	Pending	TR3	2025-03-31
4	Sathvik	Porur	Pavika	Avadi	1.50	Pending	TR4	2025-03-21

Output: Action Output table showing the execution history of the query.

#	Time	Action	Message	Duration / Fetch
76	01:32:53	SELECT * FROM couriermanagementsystem.courierservices LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
77	01:33:07	SELECT * FROM couriermanagementsystem.courier LIMIT 0, 1000	4 row(s) returned	0.016 sec / 0.000 sec
78	01:33:25	SELECT * FROM couriermanagementsystem.user LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
79	01:35:48	SELECT * FROM User WHERE Userid = 3 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
80	01:37:11	SELECT * FROM couriermanagementsystem.courier LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
81	01:37:32	SELECT * FROM Courier LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

4. List all packages for a specific order: SELECT * FROM Courier WHERE CourierID = 2;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema 'couriermanagementsystem' with tables: courier, courierservices, employeetable, locationtable, paymenttable, and user.
- SQL Editor:** Contains the query: `1 • SELECT * FROM Courier WHERE CourierID = 2;`
- Result Grid:** Displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
2	Gayathri	West Mambalam	Nisha	Ashok Nagar	2.50	Delivered	TR2	2025-03-10

Output: Action Output table showing the execution history of the query.

#	Time	Action	Message	Duration / Fetch
80	01:37:11	SELECT * FROM couriermanagementsystem.courier LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
81	01:37:32	SELECT * FROM Courier LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
82	01:40:44	SELECT * FROM couriermanagementsystem.courier LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
83	01:41:55	SELECT * FROM couriermanagementsystem.courierservices LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
84	01:42:23	SELECT * FROM couriermanagementsystem.paymenttable LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
85	01:43:20	SELECT * FROM Courier WHERE CourierID = 2 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

5. List all deliveries for a specific courier: SELECT * FROM Orders WHERE CourierID = 3;

6. List all undelivered packages: SELECT * FROM Courier WHERE Status != 'Delivered';

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the query `SELECT * FROM Courier WHERE Status != 'Delivered'` is run. The results grid displays two rows of data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
3	Monica	Palavaram	Logitha	Chrompet	5.00	Pending	TR3	2025-03-31
4	Sathvik	Porur	Pavika	Avadi	1.50	Pending	TR4	2025-03-31

The status bar at the bottom right shows the date as 21-03-2025 and the time as 01:56.

7. List all packages that are scheduled for delivery today: `SELECT * FROM Courier WHERE DeliveryDate = CURDATE();`

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the query `SELECT * FROM Courier WHERE DeliveryDate = CURDATE();` is run. The results grid displays one row of data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
4	Sathvik	Porur	Pavika	Avadi	1.50	Pending	TR4	2025-03-21

The status bar at the bottom right shows the date as 21-03-2025 and the time as 01:58.

8. List all packages with a specific status: `SELECT * FROM Courier WHERE Status = 'Delivered'`;

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, there is a single line of SQL code:

```
1 • SELECT * FROM Courier WHERE Status = 'Delivered';
```

The Result Grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Kani	OMR	Amirtha	Sadapet	3.00	Delivered	TR1	2025-03-01
2	Gayathri	West Mambalam	Nisha	Ashok Nagar	2.50	Delivered	TR2	2025-03-10
3	Monica	Palayaram	Logha	Chromepet	5.00	Pending	TR3	2025-03-31

Below the Result Grid, the Action Output pane shows the execution history of the query:

#	Time	Action	Message	Duration / Fetch
88	01:51:16	SELECT * FROM couriermanagementsystem.courier LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
89	01:53:16	SELECT * FROM Courier WHERE CourierID = 2 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
90	01:53:23	SELECT * FROM Courier WHERE CourierID = 3 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
91	01:56:14	SELECT * FROM Courier WHERE Status != 'Delivered' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
92	01:57:53	SELECT * FROM Courier WHERE DeliveryDate > CURDATE() LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
93	02:00:00	SELECT * FROM Courier WHERE Status = 'Delivered' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

9. Calculate the total number of packages for each courier: `SELECT CourierID, COUNT(*) AS TotalPackages FROM Courier GROUP BY CourierID;`

10. Find the average delivery time for each courier: `SELECT CourierID, AVG(DATEDIFF(DeliveryDate, OrderDate)) AS AvgDeliveryTime`

`FROM Courier GROUP BY CourierID;`

11. List all packages with a specific weight range: `SELECT * FROM Courier WHERE Weight BETWEEN 2 AND 5;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, there is a single line of SQL code:

```
1 • SELECT * FROM Courier WHERE Weight BETWEEN 2 AND 5;
```

The Result Grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	Kani	OMR	Amirtha	Sadapet	3.00	Delivered	TR1	2025-03-01
2	Gayathri	West Mambalam	Nisha	Ashok Nagar	2.50	Delivered	TR2	2025-03-10
3	Monica	Palayaram	Logha	Chromepet	5.00	Pending	TR3	2025-03-31

Below the Result Grid, the Action Output pane shows the execution history of the query:

#	Time	Action	Message	Duration / Fetch
89	01:53:16	SELECT * FROM Courier WHERE CourierID = 2 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
90	01:53:23	SELECT * FROM Courier WHERE CourierID = 3 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
91	01:56:14	SELECT * FROM Courier WHERE Status != 'Delivered' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
92	01:57:53	SELECT * FROM Courier WHERE DeliveryDate > CURDATE() LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
93	02:00:00	SELECT * FROM Courier WHERE Status = 'Delivered' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
94	02:08:25	SELECT * FROM Courier WHERE Weight BETWEEN 2 AND 5 LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec

12. Retrieve employees whose names contain 'John': `SELECT * FROM EmployeeTable WHERE Name LIKE '%John%';`

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to "Local instance MySQL80". The left sidebar shows the schema "couriermanagementsystem" with tables like "courier", "courserservices", "employee", "location", "payment", and "user". The main query editor window contains the following SQL code:

```

1 *  SELECT * FROM employeetable WHERE Name LIKE '%John%';
2
3

```

The result grid displays one row of data:

EmployeeID	Name	Email	ContactNumber	Role	Salary
4	John	john@gmail.com	1234567890	Admin	30.00

The status bar at the bottom right indicates the date as 21-03-2025 and the time as 02:17.

13. Retrieve all courier records with payments greater than \$50: `SELECT * FROM PaymentTable WHERE Amount > 50;`

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to "Local instance MySQL80". The left sidebar shows the schema "couriermanagementsystem" with tables like "courier", "courserservices", "employee", "location", "payment", and "user". The main query editor window contains the following SQL code:

```

1 *  SELECT * FROM paymenttable WHERE Amount > 50;
2
3
4

```

The result grid displays three rows of data:

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	1	100.00	2025-02-14
2	2	2	500.00	2025-03-01
3	3	3	800.00	2025-03-20

The status bar at the bottom right indicates the date as 21-03-2025 and the time as 02:18.

TASK 3:

14. Find the total number of couriers handled by each employee: `SELECT EmployeeID, COUNT(CourierID) AS Total_Couriers FROM courier GROUP BY EmployeeID;`

15. Calculate the total revenue generated by each location: `SELECT LocationID, SUM(Amount) AS Total_Revenue FROM paymenttable GROUP BY LocationID;`

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the following query is displayed:

```
1  SELECT LocationID, SUM(Amount) AS Total_Revenue FROM paymenttable GROUP BY LocationID;
```

The results are shown in a grid:

LocationID	Total_Revenue
1	100.00
2	500.00
3	800.00

In the bottom pane, the history shows the execution of this query along with other related statements.

16. Find the total number of couriers delivered to each location: `SELECT LocationID, COUNT(CourierID) AS Total_Couriers FROM courierservices WHERE Status = 'Delivered' GROUP BY LocationID;`

17. Find the courier with the highest average delivery time: `SELECT CourierID, AVG(DATEDIFF(DeliveryDate, OrderDate)) AS Avg_Delivery_Time FROM courierservices GROUP BY CourierID ORDER BY Avg_Delivery_Time DESC LIMIT 1;`

18. Find Locations with Total Payments Less Than a Certain Amount: `SELECT LocationID, SUM(Amount) AS Total_Payments FROM paymenttable GROUP BY LocationID HAVING Total_Payments < 2000;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, the following query is run:

```

1 SELECT LocationID, SUM(Amount) AS Total_Payments
2 FROM paymenttable GROUP BY LocationID HAVING Total_Payments < 2000;

```

The Result Grid shows the output:

LocationID	Total_Payments
1	100.00
2	500.00
3	800.00

In the Output tab, the log shows the following actions:

Action	Time	Message	Duration / Fetch
151	03:32:35	Apply changes to couriermanagementsystem	No changes detected
152	03:35:21	SELECT EmployeeID, COUNT(CourierID) AS Total_Couriers FROM courier GROUP BY EmployeeID LIMIT 0, ...	Error Code: 1054. Unknown column 'EmployeeID' in field list'
153	03:36:12	SELECT LocationID, SUM(Amount) AS Total_Revenue FROM paymenttable GROUP BY LocationID LIMIT 0, ...	3 row(s) returned
154	03:36:44	SELECT LocationID, COUNT(CourierID) AS Total_Couriers FROM courierservices WHERE Status = 'Delivered...'	Error Code: 1054. Unknown column 'LocationID' in field list'
155	03:37:09	SELECT CourierID, AVG(DATEDIFF(DeliveryDate, OrderDate)) AS Avg_Delivery_Time FROM courierservices ...	Error Code: 1054. Unknown column 'CourierID' in field list'
156	03:37:35	SELECT LocationID, SUM(Amount) AS Total_Payments FROM paymenttable GROUP BY LocationID HAVING ...	3 row(s) returned

19. Calculate Total Payments per Location: `SELECT LocationID, SUM(Amount) AS Total_Payments FROM paymenttable GROUP BY LocationID;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, the following query is run:

```

1 SELECT LocationID, SUM(Amount) AS Total_Payments
2 FROM paymenttable GROUP BY LocationID;

```

The Result Grid shows the output:

LocationID	Total_Payments
1	100.00
2	500.00
3	800.00

In the Output tab, the log shows the following actions:

Action	Time	Message	Duration / Fetch
152	03:35:21	SELECT EmployeeID, COUNT(CourierID) AS Total_Couriers FROM courier GROUP BY EmployeeID LIMIT 0, ...	Error Code: 1054. Unknown column 'EmployeeID' in field list'
153	03:36:12	SELECT LocationID, SUM(Amount) AS Total_Revenue FROM paymenttable GROUP BY LocationID LIMIT 0, ...	3 row(s) returned
154	03:36:44	SELECT LocationID, COUNT(CourierID) AS Total_Couriers FROM courierservices WHERE Status = 'Delivered...'	Error Code: 1054. Unknown column 'LocationID' in field list'
155	03:37:09	SELECT CourierID, AVG(DATEDIFF(DeliveryDate, OrderDate)) AS Avg_Delivery_Time FROM courierservices ...	Error Code: 1054. Unknown column 'CourierID' in field list'
156	03:37:35	SELECT LocationID, SUM(Amount) AS Total_Payments FROM paymenttable GROUP BY LocationID HAVING ...	3 row(s) returned
157	03:38:41	SELECT LocationID, SUM(Amount) AS Total_Payments FROM paymenttable GROUP BY LocationID LIMIT 0, ...	3 row(s) returned

20. Retrieve couriers who have received payments totaling more than \$1000 in a specific location(LocationID = X): `SELECT CourierID, LocationID, SUM(Amount) AS Total_Payments`

`FROM paymenttable WHERE LocationID = X GROUP BY CourierID, LocationID HAVING Total_Payments > 1000;`

21. Retrieve couriers who have received payments totaling more than \$1000 after a certain date(PaymentDate > 'YYYY-MM-DD'): `SELECT CourierID, SUM(Amount) AS Total_Payments`

FROM paymenttable WHERE PaymentDate > 'YYYY-MM-DD' GROUP BY CourierID HAVING Total_Payments > 1000;

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the following query is displayed:

```

1 *  SELECT CourierID, SUM(Amount) AS Total_Payments
2   FROM paymenttable WHERE PaymentDate > '2025-01-20' GROUP BY CourierID HAVING Total_Payments > 1000;
3
4

```

The results grid shows the output of the query:

CourierID	Total_Payments

In the status bar at the bottom right, it says "03:40 21-03-2025".

22. Retrieve locations where the total amount received is more than \$5000 before a certain date(PaymentDate > 'YYYY-MM-DD'): SELECT LocationID, SUM(Amount) AS Total_Payments

FROM paymenttable WHERE PaymentDate < 'YYYY-MM-DD' GROUP BY LocationID HAVING Total_Payments > 5000;

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the following query is displayed:

```

1 *  SELECT LocationID, SUM(Amount) AS Total_Payments
2   FROM paymenttable WHERE PaymentDate < '2025-02-14' GROUP BY LocationID HAVING Total_Payments > 5000;
3
4
5

```

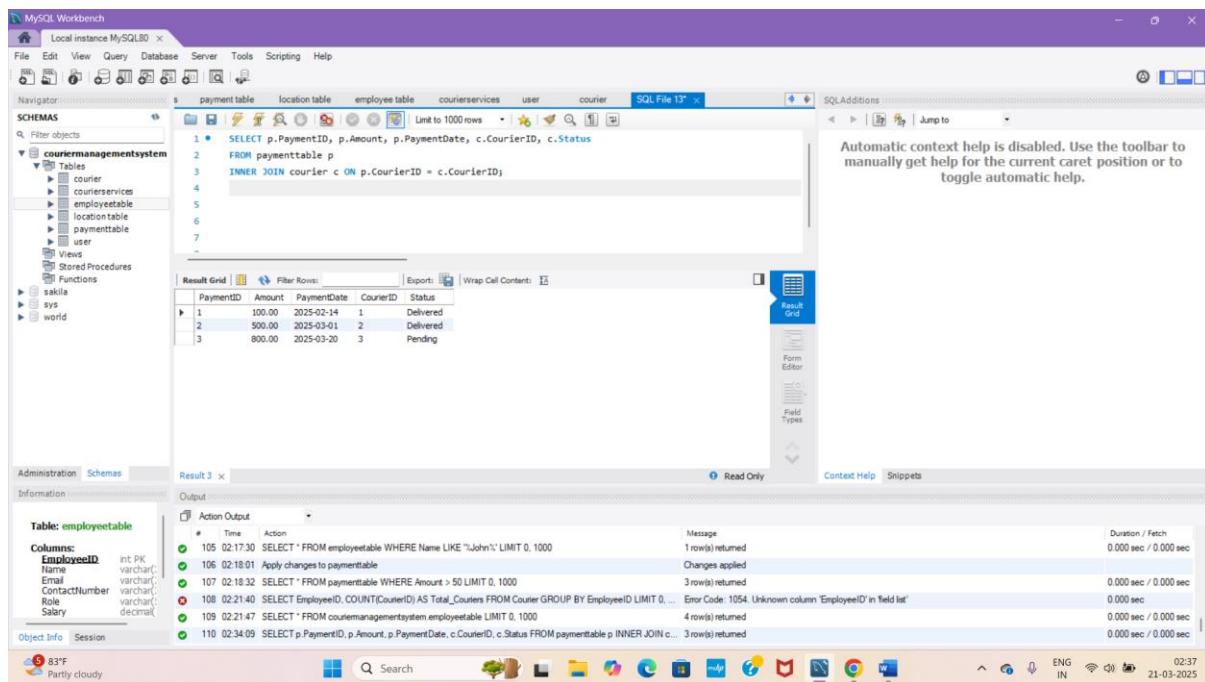
The results grid shows the output of the query:

LocationID	Total_Payments

In the status bar at the bottom right, it says "03:42 21-03-2025".

TASK 4:

23. Retrieve Payments with Courier Information: SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID;



The screenshot shows the MySQL Workbench interface with a query editor and results pane.

Query Editor:

```
1 • 1. SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status
2   FROM paymenttable p
3     INNER JOIN courier c ON p.CourierID = c.CourierID;
4
5
6
7
```

Results Grid:

PaymentID	Amount	PaymentDate	CourierID	Status
1	100.00	2025-02-14	1	Delivered
2	500.00	2025-03-01	2	Delivered
3	800.00	2025-03-20	3	Pending

Action Output:

#	Time	Action	Message	Duration / Fetch
105	02:17:30	SELECT * FROM employetable WHERE Name LIKE '%John%' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
106	02:18:01	Apply changes to paymenttable	Changes applied	
107	02:18:32	SELECT * FROM paymenttable WHERE Amount > 50 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
108	02:21:40	SELECT EmployeeID, COUNT(CourierID) AS Total_Couriers FROM Courier GROUP BY EmployeeID LIMIT 0, 1000	Error Code: 1054. Unknown column 'EmployeeID' in field list'	0.000 sec
109	02:21:47	SELECT * FROM couriermanagementsystem.employetable LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
110	02:34:09	SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID;	3 row(s) returned	0.000 sec / 0.000 sec

24. Retrieve Payments with Location Information: SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID;

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** couriermanagementsystem
- Tables:** paymenttable, locationtable, employeeable, courier
- Query:**

```

1 • SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName
  FROM paymenttable p
  INNER JOIN locationtable l ON p.LocationID = l.LocationID;
  
```
- Result Grid:**

PaymentID	Amount	PaymentDate	LocationID	LocationName
1	100.00	2025-02-14	1	Shanthi Colony
2	500.00	2025-03-01	2	Jones Road
3	800.00	2025-03-20	3	Cerus
- Output:**
 - Action Output:

#	Time	Action	Message	Duration / Fetch
109	02:21:47	SELECT * FROM couriermanagementsystem.employeeable LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
110	02:34:09	SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID INNER JOIN locationtable l ON p.LocationID = l.LocationID;	3 row(s) returned	0.000 sec / 0.000 sec
111	02:39:38	SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID;	Error Code: 1146. Table 'couriermanagementsystem.locationtable' doesn't exist.	0.000 sec
112	02:39:59	SELECT p.PaymentID, p.Amount, p.Date, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID;	Error Code: 1146. Table 'couriermanagementsystem.locationtable' doesn't exist.	0.000 sec
113	02:41:16	Apply changes to locationtable	Changes applied	
114	02:41:29	SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID;	3 row(s) returned	0.016 sec / 0.000 sec

25. Retrieve Payments with Courier and Location Information: SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID INNER JOIN locationtable l ON p.LocationID = l.LocationID;

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** couriermanagementsystem
- Tables:** paymenttable, locationtable, employeeable, courier
- Query:**

```

1 • SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status, l.LocationID, l.LocationName
  FROM paymenttable p
  INNER JOIN courier c ON p.CourierID = c.CourierID
  INNER JOIN locationtable l ON p.LocationID = l.LocationID;
  
```
- Result Grid:**

PaymentID	Amount	PaymentDate	CourierID	Status	LocationID	LocationName
1	100.00	2025-02-14	1	Delivered	1	Shanthi Colony
2	500.00	2025-03-01	2	Delivered	2	Jones Road
3	800.00	2025-03-20	3	Pending	3	Cerus
- Output:**
 - Action Output:

#	Time	Action	Message	Duration / Fetch
110	02:34:09	SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID INNER JOIN locationtable l ON p.LocationID = l.LocationID;	3 row(s) returned	0.000 sec / 0.000 sec
111	02:39:38	SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID;	Error Code: 1146. Table 'couriermanagementsystem.locationtable' doesn't exist.	0.000 sec
112	02:39:59	SELECT p.PaymentID, p.Amount, p.Date, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID;	Error Code: 1146. Table 'couriermanagementsystem.locationtable' doesn't exist.	0.000 sec
113	02:41:16	Apply changes to locationtable	Changes applied	
114	02:41:29	SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID;	3 row(s) returned	0.016 sec / 0.000 sec
115	02:43:21	SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID INNER JOIN locationtable l ON p.LocationID = l.LocationID;	3 row(s) returned	0.000 sec / 0.000 sec

26. List all payments with courier details: SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status FROM paymenttable p LEFT JOIN courier c ON p.CourierID = c.CourierID;

```

SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status
FROM paymenttable p
LEFT JOIN courier c ON p.CourierID = c.CourierID;

```

PaymentID	Amount	PaymentDate	CourierID	Status
1	100.00	2025-02-14	1	Delivered
2	500.00	2025-03-01	2	Delivered
3	800.00	2025-03-20	3	Pending

Action Output:

- 111 02:39:38 SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID; Error Code: 1146. Table 'couriermanagementsystem.locationtable' doesn't exist. 0.000 sec
- 112 02:39:59 SELECT p.PaymentID, p.Amount, p.Date, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID; Error Code: 1146. Table 'couriermanagementsystem.locationtable' doesn't exist. 0.000 sec
- 113 02:41:16 Apply changes to locationtable Changes applied 0.000 sec
- 114 02:41:29 SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID; 3 row(s) returned 0.016 sec / 0.000 sec
- 115 02:43:21 SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status, l.LocationID, l.LocationName FROM paymenttable p LEFT JOIN courier c ON p.CourierID = c.CourierID, locationtable l ON p.LocationID = l.LocationID, paymenttable p1 ON p.PaymentID = p1.PaymentID WHERE p1.PaymentDate = '2025-02-14'; 3 row(s) returned 0.000 sec / 0.000 sec
- 116 02:44:49 SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status FROM paymenttable p LEFT JOIN courier c ON p.CourierID = c.CourierID; 3 row(s) returned 0.000 sec / 0.000 sec

27. Total payments received for each courier: `SELECT c.CourierID, SUM(p.Amount) AS Total_Payments FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID GROUP BY c.CourierID;`

```

SELECT c.CourierID, SUM(p.Amount) AS Total_Payments
FROM paymenttable p
INNER JOIN courier c ON p.CourierID = c.CourierID
GROUP BY c.CourierID;

```

CourierID	Total_Payments
1	100.00
2	500.00
3	800.00

Action Output:

- 111 02:39:38 SELECT p.PaymentID, p.Amount, p.Date, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID; Error Code: 1146. Table 'couriermanagementsystem.locationtable' doesn't exist. 0.000 sec
- 112 02:39:59 SELECT p.PaymentID, p.Amount, p.Date, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID; Error Code: 1146. Table 'couriermanagementsystem.locationtable' doesn't exist. 0.000 sec
- 113 02:41:16 Apply changes to locationtable Changes applied 0.000 sec
- 114 02:41:29 SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID; 3 row(s) returned 0.016 sec / 0.000 sec
- 115 02:43:21 SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.Status, l.LocationID, l.LocationName FROM paymenttable p LEFT JOIN courier c ON p.CourierID = c.CourierID, locationtable l ON p.LocationID = l.LocationID, paymenttable p1 ON p.PaymentID = p1.PaymentID WHERE p1.PaymentDate = '2025-02-14'; 3 row(s) returned 0.000 sec / 0.000 sec
- 116 02:44:49 SELECT c.CourierID, SUM(p.Amount) AS Total_Payments FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID; 3 row(s) returned 0.000 sec / 0.000 sec
- 117 02:45:43 SELECT c.CourierID, SUM(p.Amount) AS Total_Payments FROM paymenttable p INNER JOIN courier c ON p.CourierID = c.CourierID; 3 row(s) returned 0.016 sec / 0.000 sec

28. List payments made on a specific date: `SELECT * FROM paymenttable WHERE PaymentDate = '2025-02-14';`

The screenshot shows the MySQL Workbench interface. In the top panel, a query window displays:

```

1 *  SELECT * FROM paymenttable
2 WHERE PaymentDate = '2025-02-14';
3
4
5
6
7

```

The result grid shows one row of data:

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	1	100.00	2025-02-14

In the bottom panel, the 'paymenttable 8' information window shows the structure of the employabletable:

```

Table: employabletable
Columns:
EmployeeID int PK
Name varchar(50)
Email varchar(50)
MobileNumber varchar(15)
Role varchar(50)
Salary decimal(10,2)

```

The status bar at the bottom right indicates: ENG IN 21-03-2025 02:47.

29. Get Courier Information for Each Payment: LEFT JOIN courier c ON u.UserID = c.CourierID UNION SELECT u.UserID, u.Name, c.CourierID, c.Status FROM user u

RIGHT JOIN courier c ON u.UserID = c.CourierID;

The screenshot shows the MySQL Workbench interface. In the top panel, a query window displays:

```

3 LEFT JOIN courier c ON u.UserID = c.CourierID
4
5 UNION
6
7 SELECT u.UserID, u.Name, c.CourierID, c.Status
8 FROM user u
9 RIGHT JOIN courier c ON u.UserID = c.CourierID;

```

The result grid shows the joined data:

UserID	Name	CourierID	Status
1	Nithya	1	Delivered
2	Deepa	2	Delivered
3	Lakshmi	3	Pending
4	Hasithra	4	Pending
5	Sreetha		

In the bottom panel, the 'Result 9' information window shows the structure of the employabletable:

```

Table: employabletable
Columns:
EmployeeID int PK
Name varchar(50)
Email varchar(50)
MobileNumber varchar(15)
Role varchar(50)
Salary decimal(10,2)

```

The status bar at the bottom right indicates: ENG IN 21-03-2025 02:53.

30. Get Payment Details with Location: SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNER JOIN locationtable l ON p.LocationID = l.LocationID;

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is run:

```

1 *   SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName
2     FROM paymenttable p
3   INNER JOIN locationtable l ON p.LocationID = l.LocationID;

```

The Result Grid displays the following data:

PaymentID	Amount	PaymentDate	LocationID	LocationName
1	100.00	2025-02-14	1	Shanti Colony
2	500.00	2025-03-01	2	Jones Read
3	800.00	2025-03-20	3	Cerus

Below the editor, the Action Output pane shows the execution log:

Time	Action	Message	Duration / Fetch
123 02:51:21	SELECT u.UserID, u.Name, c.CourierID, c.Status FROM user u LEFT JOIN courier c ON u.UserID = c.SenderID	Error Code: 1054. Unknown column 'u.UserID' in field list'	0.000 sec
124 02:51:36	SELECT u.UserID, u.Name, c.CourierID, c.Status FROM user u LEFT JOIN courier c ON u.UserID = c.SenderID	Error Code: 1054. Unknown column 'c.SenderID' in 'on clause'	0.000 sec
125 02:52:31	SELECT u.UserID, u.Name, c.CourierID, c.Status FROM user u LEFT JOIN courier c ON u.UserID = c.SenderID	9 row(s) returned	0.000 sec / 0.000 sec
126 02:54:34	SELECT c.CourierID, c.Status, cs.ServiceID, cs.ServiceName FROM courier c LEFT JOIN courierservices cs ...	Error Code: 1054. Unknown column 'cs.ServiceID' in field list'	0.000 sec
127 02:55:00	SELECT c.CourierID, c.Status, cs.ServiceID, cs.ServiceName FROM courier c LEFT JOIN courierservices cs ...	Error Code: 1054. Unknown column 'cs.CourierID' in 'on clause'	0.000 sec
128 02:58:14	SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNE...	3 row(s) returned	0.000 sec / 0.000 sec

31. Calculating Total Payments for Each Courier: `SELECT c.CourierID, SUM(p.Amount) AS TotalPayments FROM courier c LEFT JOIN paymenttable p ON c.CourierID = p.CourierID GROUP BY c.CourierID;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is run:

```

1 *   SELECT c.CourierID, SUM(p.Amount) AS TotalPayments
2     FROM courier c
3   LEFT JOIN paymenttable p ON c.CourierID = p.CourierID
4   GROUP BY c.CourierID;

```

The Result Grid displays the following data:

CourierID	TotalPayments
1	100.00
2	500.00
3	800.00
4	NULL

Below the editor, the Action Output pane shows the execution log:

Time	Action	Message	Duration / Fetch
124 02:51:36	SELECT u.UserID, u.Name, c.CourierID, c.Status FROM user u LEFT JOIN courier c ON u.UserID = c.SenderID	Error Code: 1054. Unknown column 'c.SenderID' in 'on clause'	0.000 sec
125 02:52:31	SELECT u.UserID, u.Name, c.CourierID, c.Status FROM user u LEFT JOIN courier c ON u.UserID = c.SenderID	9 row(s) returned	0.000 sec / 0.000 sec
126 02:54:34	SELECT c.CourierID, c.Status, cs.ServiceID, cs.ServiceName FROM courier c LEFT JOIN courierservices cs ...	Error Code: 1054. Unknown column 'cs.ServiceID' in field list'	0.000 sec
127 02:55:00	SELECT c.CourierID, c.Status, cs.ServiceID, cs.ServiceName FROM courier c LEFT JOIN courierservices cs ...	Error Code: 1054. Unknown column 'cs.CourierID' in 'on clause'	0.000 sec
128 02:58:14	SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName FROM paymenttable p INNE...	3 row(s) returned	0.000 sec / 0.000 sec
129 02:59:07	SELECT c.CourierID, SUM(p.Amount) AS TotalPayments FROM courier c LEFT JOIN paymenttable p ON c.C...	4 row(s) returned	0.000 sec / 0.000 sec

32. List Payments Within a Date Range: `SELECT * FROM paymenttable WHERE PaymentDate BETWEEN '2025-03-01' AND '2025-03-31';`

```

SELECT * FROM paymenttable
WHERE PaymentDate BETWEEN '2025-03-01' AND '2025-03-31';

```

PaymentID	CourierID	LocationID	Amount	PaymentDate
2	2	2	500.00	2025-03-01
3	3	3	800.00	2025-03-20

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side: `SELECT u.UserID, u.Name, c.CourierID, c.Status FROM user u LEFT JOIN courier c ON u.UserID = c.SenderName UNION SELECT u.UserID, u.Name, c.CourierID, c.Status FROM user u RIGHT JOIN courier c ON u.UserID = c.SenderName;`

```

SELECT u.UserID, u.Name, c.CourierID, c.Status
FROM user u
LEFT JOIN courier c ON u.UserID = c.SenderName
UNION
SELECT u.UserID, u.Name, c.CourierID, c.Status
FROM user u
RIGHT JOIN courier c ON u.UserID = c.SenderName;

```

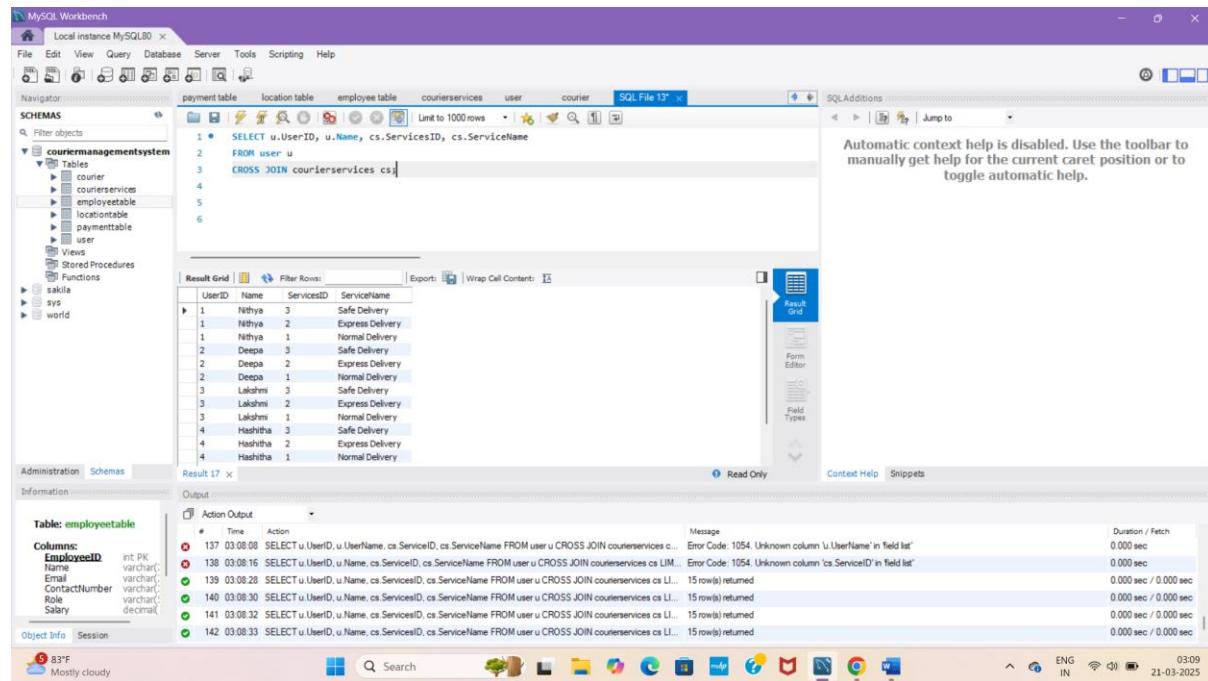
UserID	Name	CourierID	Status
1	Nithya	null	null
2	Deepa	null	null
3	Lakshmi	null	null
4	Hashitha	null	null
5	Svetha	null	null

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

34. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side: `SELECT c.CourierID, c.Status, cs.ServicesID, cs.ServiceName FROM courier c LEFT JOIN courierservices cs ON c.CourierID = cs.CourierID UNION SELECT c.CourierID, c.Status, cs.ServicesID, cs.ServiceName FROM courier c RIGHT JOIN courierservices cs ON c.CourierID = cs.CourierID;`

35. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side: `SELECT e.EmployeeID, e.Name, p.PaymentID, p.Amount FROM employeetable e LEFT JOIN paymenttable p ON e.EmployeeID = p.EmployeeID UNION SELECT e.EmployeeID, e.Name, p.PaymentID, p.Amount FROM employeetable e RIGHT JOIN paymenttable p ON e.EmployeeID = p.EmployeeID;`

36. List all users and all courier services, showing all possible combinations: `SELECT u.UserID, u.Name, cs.ServicesID, cs.ServiceName FROM user u CROSS JOIN courierservices cs;`



The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the 'couriermanagementsystem' database selected, containing tables like 'courier', 'courierservices', 'employeetable', 'locationtable', 'paymenttable', 'user', and 'views'.
- SQL Editor:** Contains the SQL query:


```
1 *  SELECT u.UserID, u.Name, cs.ServicesID, cs.ServiceName
2   FROM user u
3   CROSS JOIN courierservices cs
4
5
6
```
- Result Grid:** Displays the results of the query, showing combinations of users and courier services:

UserID	Name	ServicesID	ServiceName
1	Nithya	3	Safe Delivery
1	Nithya	2	Express Delivery
1	Nithya	1	Normal Delivery
2	Deepa	3	Safe Delivery
2	Deepa	2	Express Delivery
2	Deepa	1	Normal Delivery
3	Lakshmi	3	Safe Delivery
3	Lakshmi	2	Express Delivery
3	Lakshmi	1	Normal Delivery
4	Hashitha	3	Safe Delivery
4	Hashitha	2	Express Delivery
4	Hashitha	1	Normal Delivery
- Output:** Shows the action history with log entries for each query execution.

37. List all employees and all locations, showing all possible combinations: `SELECT e.EmployeeID, e.Name, l.LocationID, l.LocationName FROM employeetable e CROSS JOIN locationtable l;`

MySQL Workbench Screenshot:

SQL File 13*

```

1
2 •  SELECT e.EmployeeID, e.Name, l.LocationID, l.LocationName
3   FROM employeeable e
4   CROSS JOIN locationtable l;
5
6
7

```

Result Grid

EmployeeID	Name	LocationID	LocationName
1	Priyanka	3	Cerus
1	Priyanka	2	Jones Road
1	Priyanka	1	Shanti Colony
2	Prasanna	3	Cerus
2	Prasanna	2	Jones Road
2	Prasanna	1	Shanti Colony
3	Dhanush	3	Cerus
3	Dhanush	2	Jones Road
3	Dhanush	1	Shanti Colony
4	John	3	Cerus
4	John	2	Jones Road
4	John	1	Shanti Colony

Action Output

Time	Action	Message	Duration / Fetch
139 03:08:28	SELECT u.UserID, u.Name, cs.ServiceID, cs.ServiceName FROM user u CROSS JOIN courierservices cs LI...	15 row(s) returned	0.000 sec / 0.000 sec
140 03:08:30	SELECT u.UserID, u.Name, cs.ServiceID, cs.ServiceName FROM user u CROSS JOIN courierservices cs LI...	15 row(s) returned	0.000 sec / 0.000 sec
141 03:08:32	SELECT u.UserID, u.Name, cs.ServiceID, cs.ServiceName FROM user u CROSS JOIN courierservices cs LI...	15 row(s) returned	0.000 sec / 0.000 sec
142 03:08:33	SELECT u.UserID, u.Name, cs.ServiceID, cs.ServiceName FROM user u CROSS JOIN courierservices cs LI...	15 row(s) returned	0.000 sec / 0.000 sec
143 03:15:27	SELECT e.EmployeeID, e.EmployeeName, l.LocationID, l.LocationName FROM employeeable e CROSS JOI...	Error Code: 1054. Unknown column 'e.EmployeeName' in field list'	0.000 sec
144 03:15:50	SELECT e.EmployeeID, e.Name, l.LocationID, l.LocationName FROM employeeable e CROSS JOIN locationt...	12 row(s) returned	0.000 sec / 0.000 sec

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

```

SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, u.UserID AS
SenderID, u.Name AS SenderName, u.Address AS SenderAddress FROM courier c LEFT JOIN
user u ON c.SenderName = u.Name;

```

MySQL Workbench Screenshot:

SQL File 14*

```

1 •  SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status,
2   u.UserID AS SenderID, u.name AS SenderName, u.Address AS SenderAddress
3   FROM courier c
4   LEFT JOIN user u ON c.SenderName = u.Name;
5
6
7

```

Result Grid

CourierID	SenderName	ReceiverName	Weight	Status	SenderId	SenderName	SenderAddress
1	Kani	Amritha	3.00	Delivered	NULL	NULL	NULL
2	Gayathri	Nisha	2.50	Delivered	NULL	NULL	NULL
3	Monica	Loptha	5.00	Pending	NULL	NULL	NULL
4	Sathvik	Pavika	1.50	Pending	NULL	NULL	NULL

Action Output

Time	Action	Message	Duration / Fetch
36 15:36:19	SELECT e.EmployeeID, e.name, COUNT(c.CourierID) AS NumberOfCouriers FROM employeeable e LEFT JO...	Error Code: 1054. Unknown column 'e.AssignedEmployeeID' in 'on clause'	0.000 sec
37 15:36:35	SELECT e.EmployeeID, e.name, COUNT(c.CourierID) AS NumberOfCouriers FROM employeeable e LEFT JO...	Error Code: 1054. Unknown column 'e.EmployeeID' in 'on clause'	0.000 sec
38 15:37:17	SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, cs.ServiceID, cs.ServiceTyp...	Error Code: 1054. Unknown column 'cs.ServiceID' in 'field list'	0.000 sec
39 15:37:48	SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, cs.ServiceID, cs.ServiceNa...	Error Code: 1054. Unknown column 'cs.ServiceID' in 'on clause'	0.000 sec
40 15:38:22	SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, u.UserID AS ReceiverID, u.N...	4 row(s) returned	0.000 sec / 0.000 sec
41 15:39:37	SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, u.UserID AS SenderID, u.N...	4 row(s) returned	0.000 sec / 0.000 sec

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

```

SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, u.UserID AS
ReceiverID, u.Name AS ReceiverName, u.Address AS ReceiverAddress FROM courier c LEFT
JOIN user u ON c.ReceiverName = u.Name;

```

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the schema 'couriermanagementsystem' with tables like courierservices, payment table, location table, employee table, user, and courier. The SQL Editor tab contains the following query:

```

1 •  SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status,
2     u.UserID AS ReceiverID, u.Name AS ReceiverName, u.Address AS ReceiverAddress
3   FROM courier c
4  LEFT JOIN user u ON c.ReceiverName = u.Name;
5
6
7

```

The Results Grid shows the following data:

CourierID	SenderName	ReceiverName	Weight	Status	ReceiverID	ReceiverName	ReceiverAddress
1	Kari	Amrtha	3.00	Delivered	1	Kari	
2	Gayathri	Nisha	2.50	Delivered	2	Gayathri	
3	Monica	Logtha	5.00	Pending	3	Monica	
4	Sathvik	Pavika	1.50	Pending	4	Sathvik	

The Session History (Output) pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
35	15:33:58	SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPayment FROM locationtable l JOIN paymenttable p ON l.LocationID = p.LocationID GROUP BY l.LocationID ORDER BY TotalPayment DESC;	3 row(s) returned	0.000 sec / 0.000 sec
36	15:36:19	SELECT e.EmployeeID, e.EmployeeName, COUNT(c.CourierID) AS NumberOfCouriers FROM employeetable e LEFT JOIN courier c ON e.EmployeeID = c.AssignedEmployeeID GROUP BY e.EmployeeID, e.EmployeeName;	Error Code: 1054. Unknown column 'c.AssignedEmployeeID' in 'on clause'	0.000 sec
37	15:36:35	SELECT e.EmployeeID, e.EmployeeName, COUNT(c.CourierID) AS NumberOfCouriers FROM employeetable e LEFT JOIN courier c ON e.EmployeeID = c.AssignedEmployeeID GROUP BY e.EmployeeID, e.EmployeeName;	Error Code: 1054. Unknown column 'c.EmployeeID' in 'on clause'	0.000 sec
38	15:37:17	SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, cs.ServicesID, cs.ServiceType, cs.ServiceName	Error Code: 1054. Unknown column 'cs.ServiceID' in field list'	0.000 sec
39	15:37:48	SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, cs.ServicesID, cs.ServiceType, cs.ServiceName, u.UserID AS ReceiverID, u.Name AS ReceiverName, u.Address AS ReceiverAddress	Error Code: 1054. Unknown column 'cs.CourierID' in 'on clause'	0.000 sec
40	15:38:22	SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, cs.ServicesID, cs.ServiceType, cs.ServiceName, u.UserID AS ReceiverID, u.Name AS ReceiverName, u.Address AS ReceiverAddress	4 row(s) returned	0.000 sec / 0.000 sec

40. Retrieve a list of couriers along with the courier service details (if available): `SELECT c.CourierID, c.Status, cs.ServicesID, cs.ServiceName FROM courier c LEFT JOIN courierservices cs ON c.CourierID = cs.CourierID;`

41. Retrieve a list of employees and the number of couriers assigned to each employee:
`SELECT e.EmployeeID, e.EmployeeName, COUNT(c.CourierID) AS TotalCouriers FROM employeetable e LEFT JOIN courier c ON e.EmployeeID = c.AssignedEmployeeID GROUP BY e.EmployeeID, e.EmployeeName;`

42. Retrieve a list of locations and the total payment amount received at each location:
`SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPayment`

`FROM locationtable l JOIN paymenttable p ON l.LocationID = p.LocationID GROUP BY l.LocationID, l.LocationName ORDER BY TotalPayment DESC;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is run against the 'couriermanagementsystem' database:

```

1 • SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPayment
  FROM locationtable l
  JOIN paymenttable p ON l.LocationID = p.LocationID
  GROUP BY l.LocationID, l.LocationName
  ORDER BY TotalPayment DESC;

```

The results are displayed in the Result Grid:

LocationID	LocationName	TotalPayment
3	Cerus	800.00
2	Jones Road	500.00
1	Shanti Colony	100.00

Below the SQL Editor, the Log tab shows the execution history:

Action	Time	Message	Duration / Fetch
SELECT c1.*	30 15:25:40	FROM courier c1 JOIN courier c2 ON c1.SenderAddress = c2.SenderAddress WHERE c1.Count... 0 row(s) returned	0.000 sec / 0.000 sec
SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.SenderAddress FROM paymenttable p JOIN...	31 15:25:58	0 row(s) returned	0.000 sec / 0.000 sec
SELECT e1.EmployeeID, e1.EmployeeName, e1.Role FROM employeeable e1 WHERE e1.Role IN (32 15:29:55	SELE... Error Code: 1054. Unknown column 'e1.EmployeeName' in field list'	0.000 sec
SELECT e1.EmployeeID, e1.Name, e1.Role FROM employeeable e1 WHERE e1.Role IN (33 15:30:11	SELE... 0 row(s) returned	0.000 sec / 0.000 sec
SELECT c1.CourierID, c1.SenderName, c1.ReceiverName, c1.Weight, c1.Status FROM courier c1 WHERE c...	34 15:32:10	0 row(s) returned	0.000 sec / 0.000 sec
SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPayment FROM locationtable l JOIN payment...	35 15:33:58	0 row(s) returned	0.000 sec / 0.000 sec

43. Retrieve all couriers sent by the same sender (based on SenderName): `SELECT c1.CourierID, c1.SenderName, c1.ReceiverName, c1.Weight, c1.Status`

`FROM courier c1 WHERE c1.SenderName IN (SELECT SenderName FROM courier GROUP BY SenderName HAVING COUNT(*) > 1) ORDER BY c1.SenderName, c1.CourierID;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is run against the 'couriermanagementsystem' database:

```

1 • SELECT c1.CourierID, c1.SenderName, c1.ReceiverName, c1.Weight, c1.Status
  FROM courier c1
  WHERE c1.SenderName IN (
    SELECT SenderName FROM courier GROUP BY SenderName HAVING COUNT(*) > 1
  )
  ORDER BY c1.SenderName, c1.CourierID;

```

The results are displayed in the Result Grid:

CourierID	SenderName	ReceiverName	Weight	Status

Below the SQL Editor, the Log tab shows the execution history:

Action	Time	Message	Duration / Fetch
SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCountersDelivered FROM employeeable e...	29 15:24:04	4 row(s) returned	0.000 sec / 0.000 sec
SELECT c1.*	30 15:25:40	FROM courier c1 JOIN courier c2 ON c1.SenderAddress = c2.SenderAddress WHERE c1.Count... 0 row(s) returned	0.000 sec / 0.000 sec
SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.SenderAddress FROM paymenttable p JOIN...	31 15:25:58	0 row(s) returned	0.000 sec / 0.000 sec
SELECT e1.EmployeeID, e1.EmployeeName, e1.Role FROM employeeable e1 WHERE e1.Role IN (32 15:29:55	SELE... Error Code: 1054. Unknown column 'e1.EmployeeName' in field list'	0.000 sec
SELECT e1.EmployeeID, e1.Name, e1.Role FROM employeeable e1 WHERE e1.Role IN (33 15:30:11	SELE... 0 row(s) returned	0.000 sec / 0.000 sec
SELECT c1.CourierID, c1.SenderName, c1.ReceiverName, c1.Weight, c1.Status FROM courier c1 WHERE c...	34 15:32:10	0 row(s) returned	0.000 sec / 0.000 sec

44. List all employees who share the same role: `SELECT e1.EmployeeID, e1.Name, e1.Role`

`FROM employeeable e1 WHERE e1.Role IN (SELECT Role FROM employeeable GROUP BY Role HAVING COUNT(*) > 1) ORDER BY e1.Role, e1.Name;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, there is a query:

```

1 • SELECT e1.EmployeeID, e1.Name, e1.Role
  FROM employeestable e1
 WHERE e1.Role IN (
   SELECT Role FROM employeestable GROUP BY Role HAVING COUNT(*) > 1
 )
 ORDER BY e1.Role, e1.Name;

```

The Result Grid shows the output of the query:

EmployeeID	Name	Role

In the bottom pane, the Action Output shows the execution log:

#	Time	Action	Message	Duration / Fetch
28	15:23:03	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeeable e LEFT JOIN courier c ON e.EmployeeID = c.CourierID WHERE c.SenderAddress IN (SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1) ORDER BY c.SenderAddress, e.PaymentDate;	Error Code: 1054. Unknown column 'c.DeliveredBy' in 'on clause'	0.000 sec
29	15:24:04	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeeable e LEFT JOIN courier c ON e.EmployeeID = c.CourierID WHERE c.SenderAddress IN (SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1) ORDER BY c.SenderAddress, e.PaymentDate;	4 row(s) returned	0.000 sec / 0.000 sec
30	15:25:40	SELECT c1.* FROM courier c1 JOIN courier c2 ON c1.SenderAddress = c2.SenderAddress WHERE c1.CourierID <> c2.CourierID ORDER BY c1.SenderAddress, c2.SenderAddress;	0 row(s) returned	0.000 sec / 0.000 sec
31	15:26:58	SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.SenderAddress FROM paymenttable p JOIN courier c ON p.CourierID = c.CourierID WHERE c.SenderAddress IN (SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1) ORDER BY c.SenderAddress, p.PaymentDate;	0 row(s) returned	0.000 sec / 0.000 sec
32	15:29:55	SELECT e1.EmployeeID, e1.EmployeeName, e1.Role FROM employeeable e1 WHERE e1.Role IN (SELECT Role FROM employeestable GROUP BY Role HAVING COUNT(*) > 1) ORDER BY e1.Role, e1.Name;	Error Code: 1054. Unknown column 'e1.EmployeeName' in 'field list'	0.000 sec
33	15:30:11	SELECT e1.EmployeeID, e1.Name, e1.Role FROM employeeable e1 WHERE e1.Role IN (SELECT Role FROM employeestable GROUP BY Role HAVING COUNT(*) > 1) ORDER BY e1.Role, e1.Name;	0 row(s) returned	0.000 sec / 0.000 sec

45. Retrieve all payments made for couriers sent from the same location: SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.SenderAddress FROM paymenttable p JOIN courier c ON p.CourierID = c.CourierID WHERE c.SenderAddress IN (SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1) ORDER BY c.SenderAddress, p.PaymentDate;

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, there is a query:

```

1 • SELECT c.CourierID, p.Amount, cs.Count FROM courier c JOIN paymenttable p ON c.CourierID = p.CourierID WHERE c.SenderAddress IN ( SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1 ) ORDER BY c.SenderAddress, p.PaymentDate;

```

The Result Grid shows the output of the query:

CourierID	Amount	PaymentDate	CourierID	SenderAddress

In the bottom pane, the Action Output shows the execution log:

#	Time	Action	Message	Duration / Fetch
26	15:21:47	SELECT c.CourierID, p.Amount, cs.Count FROM courier c JOIN paymenttable p ON c.CourierID = p.CourierID WHERE c.SenderAddress IN (SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1) ORDER BY c.SenderAddress, p.PaymentDate;	Error Code: 1054. Unknown column 'c.ServicesID' in 'on clause'	0.000 sec
27	15:22:59	SELECT * FROM courermanagementsystem.locationtable LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
28	15:23:03	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeeable e LEFT JOIN courier c ON e.EmployeeID = c.CourierID WHERE c.SenderAddress IN (SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1) ORDER BY c.SenderAddress, e.PaymentDate;	Error Code: 1054. Unknown column 'c.DeliveredBy' in 'on clause'	0.000 sec
29	15:24:04	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeeable e LEFT JOIN courier c ON e.EmployeeID = c.CourierID WHERE c.SenderAddress IN (SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1) ORDER BY c.SenderAddress, e.PaymentDate;	4 row(s) returned	0.000 sec / 0.000 sec
30	15:25:40	SELECT c1.* FROM courier c1 JOIN courier c2 ON c1.SenderAddress = c2.SenderAddress WHERE c1.CourierID <> c2.CourierID ORDER BY c1.SenderAddress, c2.SenderAddress;	0 row(s) returned	0.000 sec / 0.000 sec
31	15:26:58	SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.SenderAddress FROM paymenttable p JOIN courier c ON p.CourierID = c.CourierID WHERE c.SenderAddress IN (SELECT SenderAddress FROM courier GROUP BY SenderAddress HAVING COUNT(*) > 1) ORDER BY c.SenderAddress, p.PaymentDate;	0 row(s) returned	0.000 sec / 0.000 sec

46. Retrieve all couriers sent from the same location (based on SenderAddress): SELECT c1.* FROM courier c1 JOIN courier c2 ON c1.SenderAddress = c2.SenderAddress WHERE c1.CourierID <> c2.CourierID ORDER BY c1.SenderAddress;

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is written:

```

1 *  SELECT c1.*
2   FROM courier c1
3   JOIN courier c2 ON c1.SenderAddress = c2.SenderAddress
4   WHERE c1.CourierID <> c2.CourierID
5   ORDER BY c1.SenderAddress;
6
7

```

The Result Grid shows the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate

Output window (Action Output):

#	Time	Action	Message	Duration / Fetch
25	15:21:13	SELECT c.CourierID, p.Amount, cs.Cost FROM courier c JOIN paymenttable p ON c.CourierID = p.CourierID JOIN courierservices cs ON c.CourierID = cs.CourierID	Error Code: 1054. Unknown column 'cs.ServiceID' in 'on clause'	0.000 sec
26	15:21:47	SELECT c.CourierID, p.Amount, cs.Cost FROM courier c JOIN paymenttable p ON c.CourierID = p.CourierID JOIN courierservices cs ON c.CourierID = cs.CourierID	Error Code: 1054. Unknown column 'cs.ServicesID' in 'on clause'	0.000 sec
27	15:22:59	SELECT * FROM couriermanagementsystem..locationtable LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
28	15:23:03	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeetable e LEFT JOIN courier c ON e.EmployeeID = c.Status GROUP BY e.EmployeeID, e.Name;	Error Code: 1054. Unknown column 'c.DeliveredBy' in 'on clause'	0.000 sec
29	15:24:04	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeetable e LEFT JOIN courier c ON e.EmployeeID = c.Status GROUP BY e.EmployeeID, e.Name;	4 row(s) returned	0.000 sec / 0.000 sec
30	15:25:40	SELECT c1.* FROM courier c1 JOIN courier c2 ON c1.SenderAddress = c2.SenderAddress WHERE c1.CourierID <> c2.CourierID ORDER BY c1.SenderAddress;	0 row(s) returned	0.000 sec / 0.000 sec

47. List employees and the number of couriers they have delivered: `SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeetable e LEFT JOIN courier c ON e.EmployeeID = c.Status GROUP BY e.EmployeeID, e.Name;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is written:

```

1 *  SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered
2   FROM employeetable e
3   LEFT JOIN courier c ON e.EmployeeID = c.Status
4   GROUP BY e.EmployeeID, e.Name;
5
6

```

The Result Grid shows the following data:

EmployeeID	Name	TotalCouriersDelivered
1	Priyanka	0
2	Prasanna	0
3	Dhanush	0
4	John	0

Output window (Action Output):

#	Time	Action	Message	Duration / Fetch
24	15:20:32	SELECT * FROM couriermanagementsystem..paymenttable LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
25	15:21:13	SELECT c.CourierID, p.Amount, cs.Cost FROM courier c JOIN paymenttable p ON c.CourierID = p.CourierID JOIN courierservices cs ON c.CourierID = cs.CourierID	Error Code: 1054. Unknown column 'cs.ServiceID' in 'on clause'	0.000 sec
26	15:21:47	SELECT c.CourierID, p.Amount, cs.Cost FROM courier c JOIN paymenttable p ON c.CourierID = p.CourierID JOIN courierservices cs ON c.CourierID = cs.CourierID	Error Code: 1054. Unknown column 'cs.ServicesID' in 'on clause'	0.000 sec
27	15:22:59	SELECT * FROM couriermanagementsystem..locationtable LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
28	15:23:03	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeetable e LEFT JOIN courier c ON e.EmployeeID = c.Status GROUP BY e.EmployeeID, e.Name;	Error Code: 1054. Unknown column 'c.DeliveredBy' in 'on clause'	0.000 sec
29	15:24:04	SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersDelivered FROM employeetable e LEFT JOIN courier c ON e.EmployeeID = c.Status GROUP BY e.EmployeeID, e.Name;	4 row(s) returned	0.000 sec / 0.000 sec

48. Find couriers that were paid an amount greater than the cost of their respective courier services: `SELECT c.CourierID, c.ServiceCost, p.Amount FROM courier c INNER JOIN paymenttable p ON c.CourierID = p.CourierID WHERE p.Amount > c.ServiceCost;`

49. Find couriers that have a weight greater than the average weight of all couriers: `SELECT CourierID, Weight FROM courier WHERE Weight > (SELECT AVG(Weight) FROM courier);`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is run:

```

1 • SELECT CourierID, Weight
  FROM courier
 WHERE Weight > (SELECT AVG(Weight) FROM courier);

```

The Result Grid shows the output:

CourierID	Weight
3	5.00

In the Output tab, the execution log is displayed:

#	Time	Action	Message	Duration / Fetch		
17	15:16:32	SELECT SUM(Cost) AS TotalCost FROM courierservices WHERE Cost < (SELECT MAX(Cost) FROM courierservices)	1 row(s) returned	0.000 sec / 0.000 sec		
18	15:17:19	SELECT EmployeeName, Salary FROM employeetable WHERE Salary > (SELECT AVG(Salary) FROM employeetable)	Error Code: 1054. Unknown column 'EmployeeName' in 'field list'	0.000 sec		
19	15:17:27	SELECT * FROM couriermanagementsystem.employee <table border="1"> <tr><td>table</td></tr> </table> LIMIT 0, 1000	table	Error Code: 1146. Table 'couriermanagementsystem.employee <table border="1"> <tr><td>table</td></tr> </table> ' doesn't exist	table	0.000 sec
table						
table						
20	15:17:40	SELECT * FROM couriermanagementsystem.employeetable LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec		
21	15:18:07	SELECT Name, Salary FROM employeetable WHERE Salary > (SELECT AVG(Salary) FROM employeetable) L...	2 row(s) returned	0.000 sec / 0.000 sec		
22	15:19:17	SELECT CourierID, Weight FROM courier WHERE Weight > (SELECT AVG(Weight) FROM courier) LIMIT 0, ...	1 row(s) returned	0.000 sec / 0.000 sec		

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

SELECT Name, Salary FROM employeetable WHERE Salary > (SELECT AVG(Salary) FROM employeetable);

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is run:

```

1 • SELECT Name, Salary
  FROM employeetable
 WHERE Salary > (SELECT AVG(Salary) FROM employeetable);

```

The Result Grid shows the output:

Name	Salary
Priyanka	50.00
John	30.00

In the Output tab, the execution log is displayed:

#	Time	Action	Message	Duration / Fetch		
16	15:14:13	SELECT c.CourierID, c.Status, p.Amount, p.PaymentDate FROM courier c JOIN paymenttable p ON c.CourierID = p.CourierID	3 row(s) returned	0.016 sec / 0.000 sec		
17	15:16:32	SELECT SUM(Cost) AS TotalCost FROM courierservices WHERE Cost < (SELECT MAX(Cost) FROM courierservices)	1 row(s) returned	0.000 sec / 0.000 sec		
18	15:17:19	SELECT EmployeeName, Salary FROM employeetable WHERE Salary > (SELECT AVG(Salary) FROM employeetable)	Error Code: 1054. Unknown column 'EmployeeName' in 'field list'	0.000 sec		
19	15:17:27	SELECT * FROM couriermanagementsystem.employee <table border="1"> <tr><td>table</td></tr> </table> LIMIT 0, 1000	table	Error Code: 1146. Table 'couriermanagementsystem.employee <table border="1"> <tr><td>table</td></tr> </table> ' doesn't exist	table	0.000 sec
table						
table						
20	15:17:40	SELECT * FROM couriermanagementsystem.employeetable LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec		
21	15:18:07	SELECT Name, Salary FROM employeetable WHERE Salary > (SELECT AVG(Salary) FROM employeetable) L...	2 row(s) returned	0.000 sec / 0.000 sec		

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

SELECT SUM(ServiceCost) AS TotalCost FROM courierservices WHERE ServiceCost < (SELECT MAX(ServiceCost) FROM courierservices);

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is written:

```

1 *  SELECT SUM(Cost) AS TotalCost
  FROM courierservices
 WHERE Cost < (SELECT MAX(Cost) FROM courierservices);

```

The Result Grid shows the output of the query:

TotalCost
600.00

In the bottom status bar, it says "15:16 21-03-2025".

52. Find all couriers that have been paid for: `SELECT c.CourierID, c.Status, p.Amount, p.PaymentDate FROM courier c JOIN paymenttable p ON c.CourierID = p.CourierID;`

The screenshot shows the MySQL Workbench interface. In the SQL Editor tab, a query is written:

```

1 *  SELECT c.CourierID, c.Status, p.Amount, p.PaymentDate
  FROM courier c
 JOIN paymenttable p ON c.CourierID = p.CourierID;

```

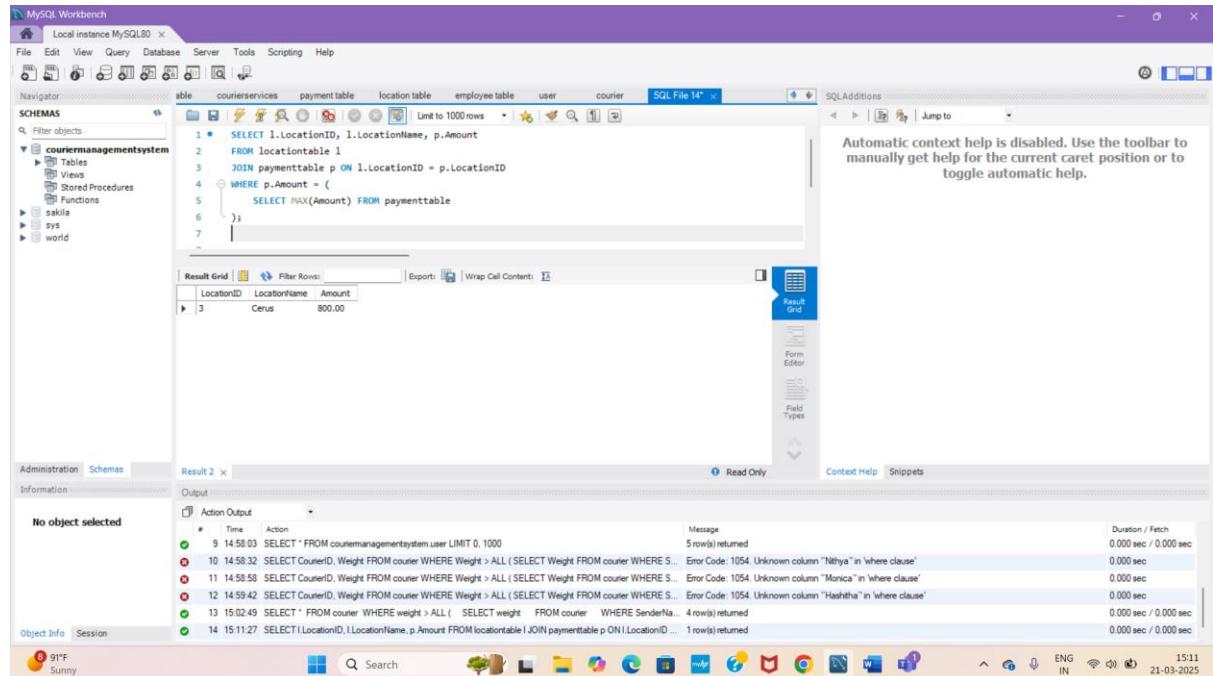
The Result Grid shows the output of the query:

CourierID	Status	Amount	PaymentDate
1	Delivered	100.00	2025-02-14
2	Delivered	500.00	2025-03-01
3	Pending	800.00	2025-03-20

In the bottom status bar, it says "15:14 21-03-2025".

53. Find the locations where the maximum payment amount was made: `SELECT l.LocationID, l.LocationName, p.Amount FROM locationtable l JOIN paymenttable p ON l.LocationID = p.LocationID WHERE p.Amount = (SELECT MAX(Amount) FROM`

paymenttable);



The screenshot shows the MySQL Workbench interface with the following details:

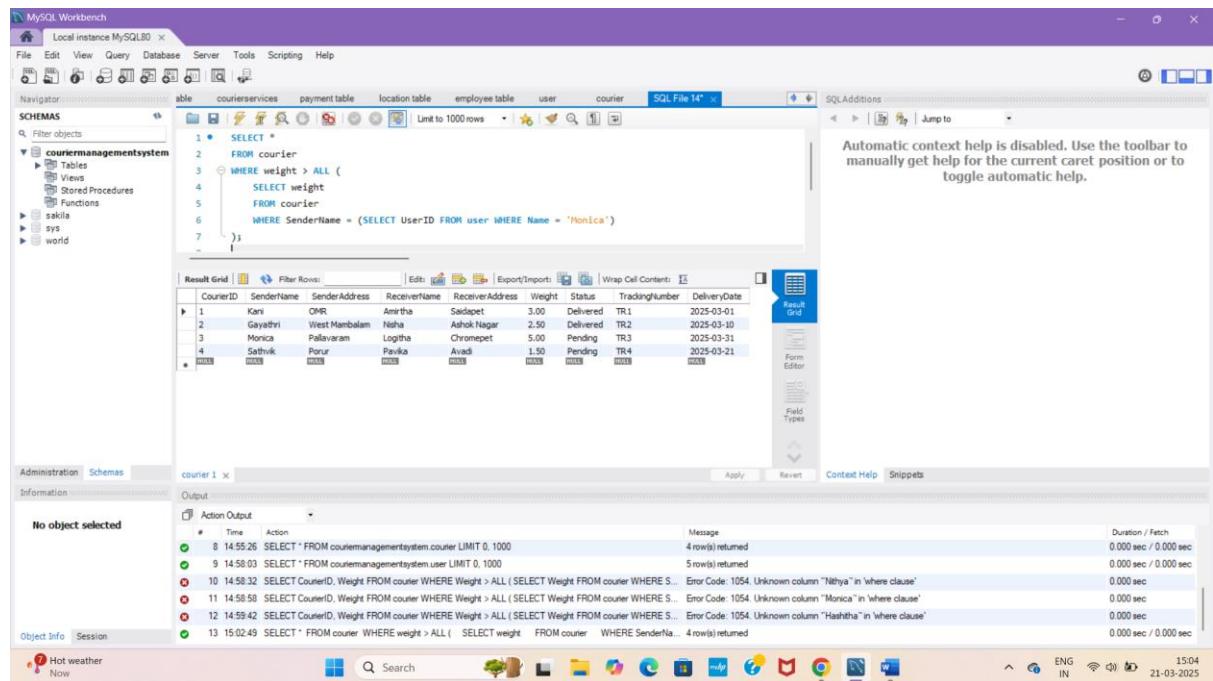
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** couriermanagementsystem, sakila, sys, world.
- SQL Editor:**

```

1 *   SELECT l.LocationID, l.LocationName, p.Amount
2     FROM locationtable
3   JOIN paymenttable p ON l.LocationID = p.LocationID
4   WHERE p.Amount = (
5       SELECT MAX(Amount) FROM paymenttable
6   );
    
```
- Result Grid:**

LocationID	LocationName	Amount
3	Cerus	800.00
- Output Tab:** Action Output showing the history of SQL statements executed.
- System Bar:** Shows the date (21-03-2025), time (15:11), and system status (ENG IN).

54. Find all couriers whose weight is greater than the weight of all couriers sent by a specific sender (e.g., 'SenderName'): `SELECT * FROM courier WHERE weight > ALL (SELECT weight FROM courier WHERE SenderName = (SELECT UserID FROM user WHERE Name = 'Monica'));`



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** couriermanagementsystem, sakila, sys, world.
- SQL Editor:**

```

1 *   SELECT *
2     FROM courier
3   WHERE weight > ALL (
4       SELECT weight
5         FROM courier
6       WHERE SenderName = (SELECT UserID FROM user WHERE Name = 'Monica'))
    
```
- Result Grid:**

CourierID	SenderId	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	NULL	Kani	OMR	Amirtha	Saiyedpet	3.00	Delivered	TR-1	2025-03-01
2	NULL	Gayathri	West Mambalam	Nisha	Ashok Nagar	2.50	Delivered	TR-2	2025-03-10
3	NULL	Monica	Palavaram	Logitha	Chromepet	5.00	Pending	TR-3	2025-03-31
4	NULL	Sathvik	Porur	Pavika	Avadi	1.50	Pending	TR-4	2025-03-21
- Output Tab:** Action Output showing the history of SQL statements executed.
- System Bar:** Shows the date (21-03-2025), time (15:04), and system status (ENG IN).