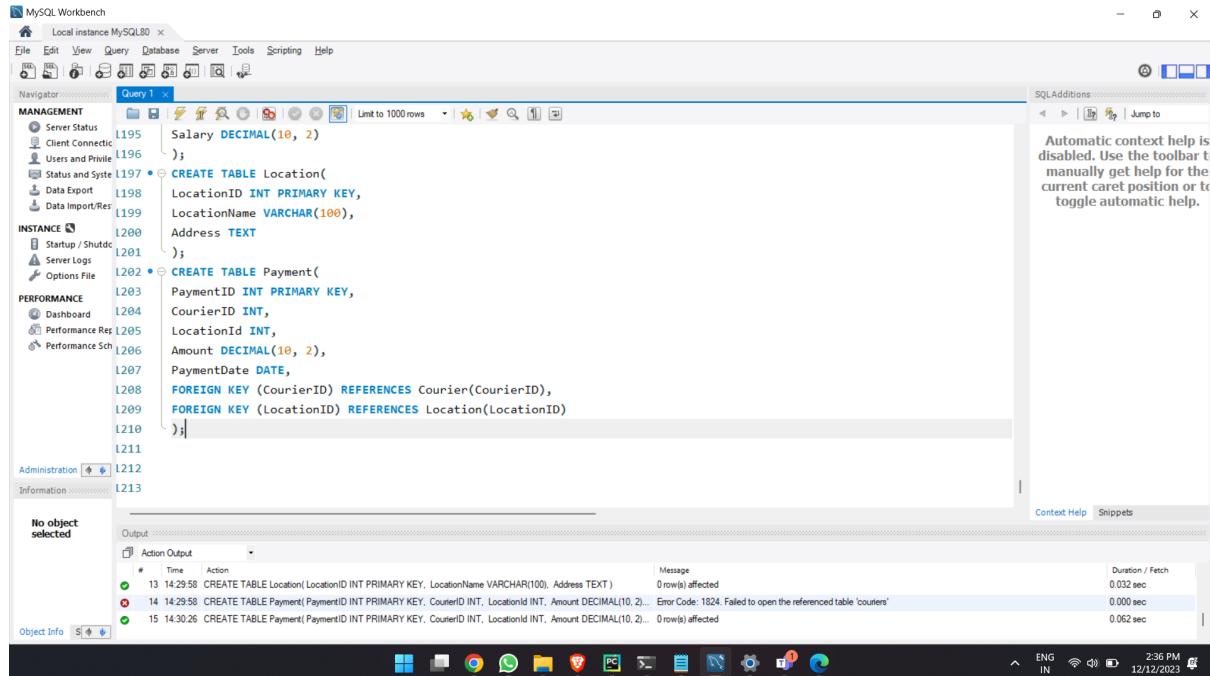


Task1 Database Design

Design a SQL schema for a Courier Management System with tables for Customers, Couriers, Orders, and Parcels. Define the relationships between these tables using appropriate foreign keys.



The screenshot shows the MySQL Workbench interface with a query editor window open. The code in the query editor is as follows:

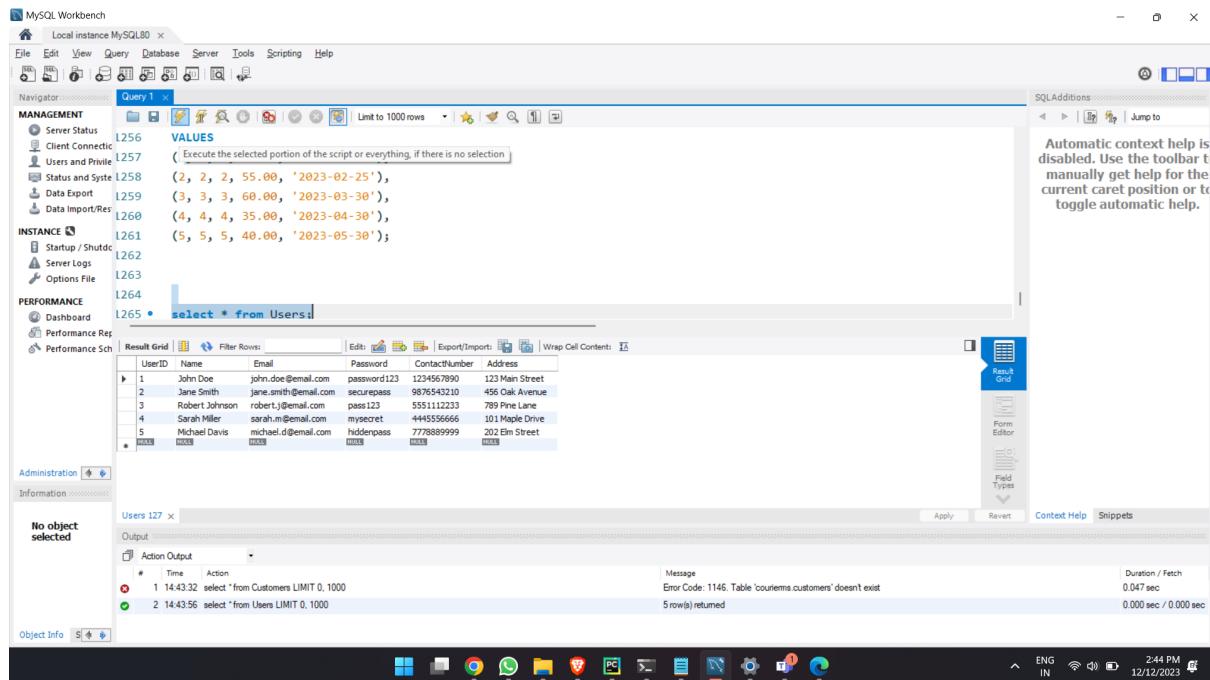
```
MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help
File Edit View Query Database Server Tools Scripting Help
Navigator: Query 1
MANAGEMENT
  Server Status L195   Salary DECIMAL(10, 2)
  Client Connectors L196   );
  Users and Privileges L197 * CREATE TABLE Location(
  Status and System L198   LocationID INT PRIMARY KEY,
  Data Export L199   LocationName VARCHAR(100),
  Data Import/Export L200   Address TEXT
  );
  INSTANCE
    Startup / Shutdown L201   );
    Server Logs L202 * CREATE TABLE Payment(
    Options File L203   PaymentID INT PRIMARY KEY,
    L204   CourierID INT,
    Performance
      Dashboard L205   LocationId INT,
      Performance Reports L206   Amount DECIMAL(10, 2),
      Performance Schedule L207   PaymentDate DATE,
      L208   FOREIGN KEY (CourierID) REFERENCES Courier(CourierID),
      L209   FOREIGN KEY (LocationID) REFERENCES Location(LocationID)
      L210   );
      L211
      Administration L212
      Information L213
      No object selected
      Output
        Action Output
          # Time Action Message Duration / Fetch
          13 14:29:58 CREATE TABLE Location(LocationID INT PRIMARY KEY, LocationName VARCHAR(100), Address TEXT) 0 row(s) affected 0.032 sec
          14 14:29:58 CREATE TABLE Payment(PaymentID INT PRIMARY KEY, CourierID INT, LocationId INT, Amount DECIMAL(10, 2))... Error Code: 1824. Failed to open the referenced table 'couriers' 0.000 sec
          15 14:30:26 CREATE TABLE Payment(PaymentID INT PRIMARY KEY, CourierID INT, LocationId INT, Amount DECIMAL(10, 2)... 0 row(s) affected 0.062 sec
      Object Info S + B
```

The output pane shows the results of the executed queries. The first two queries were successful, while the third query failed due to an error (Error Code: 1824). The status bar at the bottom right indicates the time as 2:36 PM and the date as 12/12/2023.

Task 2: Select,Where Solve the following queries in the Schema that you have created above

Q1. List all customers:



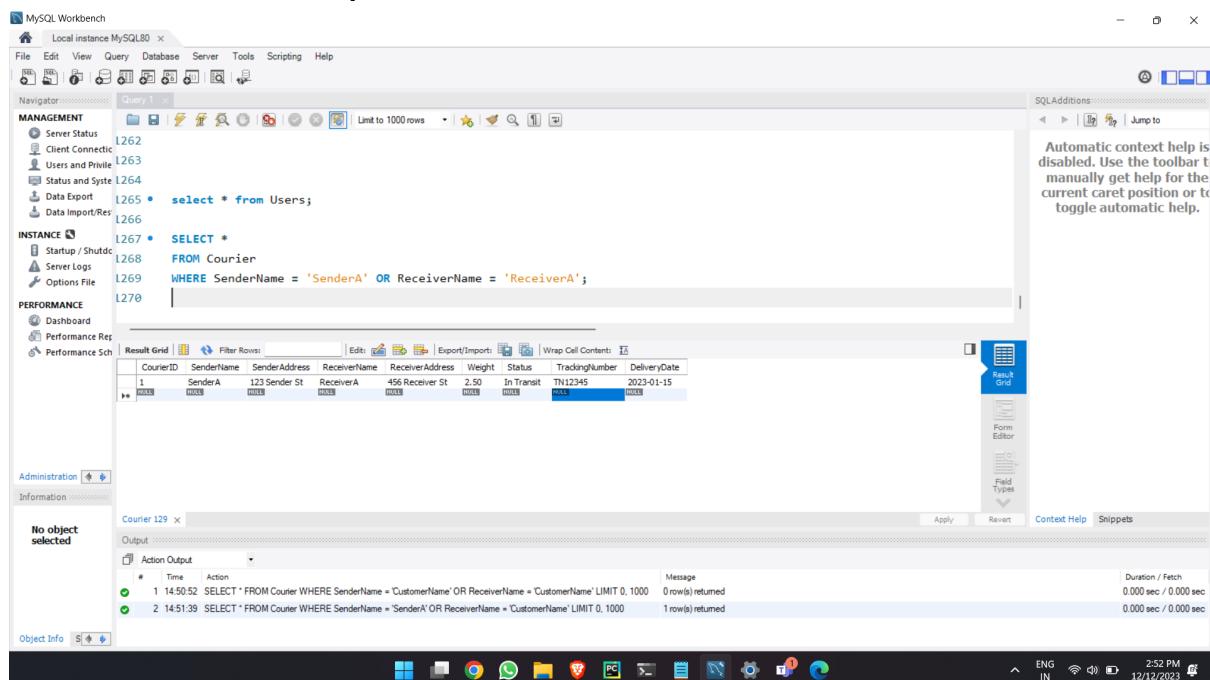
The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL query: `select * from Users;`
- Result Grid:** Displays the following data for the 'Users' table:

User ID	Name	Email	Password	Contact Number	Address
1	John Doe	john.doe@email.com	password123	123 Main Street	
2	Jane Smith	jane.smith@email.com	securypass	9876543210	456 Oak Avenue
3	Robert Johnson	robert.j@email.com	pass123	5551112233	789 Pine Lane
4	Sarah Miller	sarah.m@email.com	mysecret	4445556666	101 Maple Drive
5	Michael Davis	michael.d@email.com	hiddenpass	7778889999	202 Elm Street

- Output Window:** Shows the execution history with two entries:
 - Action: `select * from Users LIMIT 0, 1000` - Duration: 0.047 sec
 - Action: `select * from Users LIMIT 0, 1000` - Duration: 0.000 sec / 0.000 sec

Q2. List all orders for a specific customer:



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

- Query Editor:** Contains the SQL query: `SELECT * FROM Courier WHERE SenderName = 'SenderA' OR ReceiverName = 'ReceiverA';`
- Result Grid:** Displays the following data for the 'Courier' table:

CourierID	SenderId	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	123	SenderA	123 Sender St	ReceiverA	456 Receiver St	2.50	In Transit	TN12345	2023-01-15

- Output Window:** Shows the execution history with two entries:
 - Action: `SELECT * FROM Courier WHERE SenderName = 'CustomerName' OR ReceiverName = 'CustomerName' LIMIT 0, 1000` - Duration: 0.000 sec / 0.000 sec
 - Action: `SELECT * FROM Courier WHERE SenderName = 'CustomerName' OR ReceiverName = 'CustomerName' LIMIT 0, 1000` - Duration: 0.000 sec / 0.000 sec

Q3. List all couriers:

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

- Navigator:** Shows the database structure with nodes like MANAGEMENT, INSTANCE, and PERFORMANCE.
- Query Editor:** Contains the following SQL code:

```
L265 • select * from Users;
L267 • SELECT *
FROM Courier
WHERE SenderName = 'SenderA' OR ReceiverName = 'ReceiverA';
L271 • SELECT * FROM Couriers;
```
- Result Grid:** Displays the results of the last query, showing 5 rows of courier information:| CourierID | SenderName | SenderAddress | ReceiverName | ReceiverAddress | Weight | Status | TrackingNumber | DeliveryDate |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | SenderA | 123 Sender St | ReceiverA | 456 Receiver St | 2.50 | In Transit | TN12345 | 2023-01-15 |
| 2 | SenderB | 789 Sender St | ReceiverB | 101 Receiver St | 3.00 | Delivered | TN67890 | 2023-02-20 |
| 3 | SenderC | 456 Sender St | ReceiverC | 202 Receiver St | 1.80 | Pending | TN34567 | MNULL |
| 4 | SenderD | 101 Sender St | ReceiverD | 303 Receiver St | 2.20 | In Transit | TN23456 | 2023-04-10 |
| 5 | SenderE | 202 Sender St | ReceiverE | 404 Receiver St | 4.50 | Delivered | TN78901 | 2023-05-18 |
- Output Window:** Shows the execution history for the current session:| # | Time | Action | Message | Duration / Fetch |
| --- | --- | --- | --- | --- |
| 1 | 14:50:52 | SELECT * FROM Courier WHERE SenderName = 'CustomerName' OR ReceiverName = 'CustomerName' LIMIT 0, 1000 | 0 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 14:51:39 | SELECT * FROM Courier WHERE SenderName = 'SenderA' OR ReceiverName = 'ReceiverA' LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 14:53:31 | SELECT * FROM Courier LIMIT 0, 1000 | 5 row(s) returned | 0.000 sec / 0.000 sec |

Q4. List all packages for a specific order:

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

- Navigator:** Shows the database structure with nodes like MANAGEMENT, INSTANCE, and PERFORMANCE.
- Query Editor:** Contains the following SQL code:

```
L265 • select * from Users;
L267 • SELECT *
FROM Courier
WHERE SenderName = 'SenderA' OR ReceiverName = 'ReceiverA';
L271 • SELECT * FROM Couriers;
L273 • SELECT * FROM Courier WHERE CourierID = 3;
```
- Result Grid:** Displays the results of the last query, showing 1 row of courier information:| CourierID | SenderName | SenderAddress | ReceiverName | ReceiverAddress | Weight | Status | TrackingNumber | DeliveryDate |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | SenderC | 456 Sender St | ReceiverC | 202 Receiver St | 1.80 | Pending | TN34567 | MNULL |
- Output Window:** Shows the execution history for the current session:| # | Time | Action | Message | Duration / Fetch |
| --- | --- | --- | --- | --- |
| 2 | 14:51:39 | SELECT * FROM Courier WHERE SenderName = 'SenderA' OR ReceiverName = 'ReceiverA' LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 14:53:31 | SELECT * FROM Courier LIMIT 0, 1000 | 5 row(s) returned | 0.000 sec / 0.000 sec |
| 4 | 14:54:39 | SELECT * FROM Courier WHERE CourierID = 3 LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |

Q5. List all deliveries for a specific courier:

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

- Query Editor:** Contains the following SQL code:

```
L267 • SELECT *
FROM Courier
WHERE SenderName = 'SenderA' OR ReceiverName = 'ReceiverA';
L270
L271 • SELECT * FROM Courier;
L272
L273 • SELECT * FROM Courier WHERE CourierID = 3;
L274
L275 • SELECT * FROM Courier WHERE CourierID = 4;
```
- Result Grid:** Displays the results of the last query (CourierID = 4).

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
4	SenderD	101 Sender St	ReceiverD	303 Receiver St	2.20	In Transit	TN23456	2023-04-10
- Information Panel:** Shows the execution history for the current session:

#	Time	Action	Message	Duration / Fetch
3	14:53:31	SELECT * FROM Courier LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	14:54:39	SELECT * FROM Courier WHERE CourierID = 3 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	14:56:03	SELECT * FROM Courier WHERE CourierID = 4 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Q6. List all undelivered packages:

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

- Query Editor:** Contains the following SQL code:

```
L268 FROM Courier
L269 WHERE SenderName = 'SenderA' OR ReceiverName = 'ReceiverA';
L270
L271 • SELECT * FROM Courier;
L272
L273 • SELECT * FROM Courier WHERE CourierID = 3;
L274
L275 • SELECT * FROM Courier WHERE CourierID = 4;
L276
L277 • SELECT * FROM Courier WHERE Status = 'In Transit';
```
- Result Grid:** Displays the results of the last query (Status = 'In Transit').

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	SenderA	123 Sender St	ReceiverA	456 Receiver St	2.50	In Transit	TN12345	2023-01-15
4	SenderD	101 Sender St	ReceiverD	303 Receiver St	2.20	In Transit	TN23456	2023-04-10
- Information Panel:** Shows the execution history for the current session:

#	Time	Action	Message	Duration / Fetch
6	14:56:54	SELECT * FROM Courier WHERE Status = 'Undelivered' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
7	14:57:09	SELECT * FROM Courier LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
8	14:58:03	SELECT * FROM Courier WHERE Status = 'In Transit' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Q7. List all packages that are scheduled for delivery today:

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
L271 • SELECT * FROM Courier;
L272
L273 • SELECT * FROM Courier WHERE CourierID = 3;
L274
L275 • SELECT * FROM Courier WHERE CourierID = 4;
L276
L277 • SELECT * FROM Courier WHERE Status = 'In Transit';
L278
L279 • SELECT * FROM Courier WHERE DeliveryDate = CURDATE();
L280
```

The results grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The status bar at the bottom right indicates the time as 3:02 PM and the date as 12/12/2023.

Q8. List all packages with a specific status:

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
L272
L273 • SELECT * FROM Courier WHERE CourierID = 3;
L274
L275 • SELECT * FROM Courier WHERE CourierID = 4;
L276
L277 • SELECT * FROM Courier WHERE Status = 'In Transit';
L278
L279 • SELECT * FROM Courier WHERE DeliveryDate = CURDATE();
L280 • SELECT * FROM Courier WHERE Status = 'Delivered';
L281
```

The results grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
2	SenderB	789 Sender St	ReceiverB	101 Receiver St	3.00	Delivered	TN67890	2023-02-20
5	SenderE	202 Sender St	ReceiverE	404 Receiver St	4.50	Delivered	TN78901	2023-05-18

The status bar at the bottom right indicates the time as 3:07 PM and the date as 12/12/2023.

Q9. Calculate the total number of packages for each courier.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The SQL code is:

```
L274 •    SELECT * FROM Courier WHERE CourierID = 4;
L275 •    SELECT * FROM Courier WHERE Status = 'In Transit';
L276
L277 •    SELECT * FROM Courier WHERE DeliveryDate = CURDATE();
L278
L279 •    SELECT * FROM Courier WHERE Status = 'Delivered';
L280
L281 •    SELECT CourierID, COUNT(*) AS TotalPackages FROM Courier GROUP BY CourierID;
L282
L283 •    SELECT CourierID, COUNT(*) AS TotalPackages FROM Courier GROUP BY CourierID LIMIT 0, 1000;
```

The results grid shows the following data:

CourierID	TotalPackages
1	1
2	1
3	1
4	1
5	1

The status bar at the bottom right indicates the time is 3:09 PM and the date is 12/12/2023.

Q10. Find the average delivery time for each courier

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The SQL code is:

```
L280
L281 •    SELECT * FROM Courier WHERE Status = 'Delivered';
L282
L283 •    SELECT CourierID, COUNT(*) AS TotalPackages FROM Courier GROUP BY CourierID;
L284
L285
L286 •    SELECT CourierID, AVG(DATEDIFF(DeliveryDate, '2022-12-25')) AS AverageDeliveryTime
L287     FROM Courier GROUP BY CourierID;
L288
```

The results grid shows the following data:

CourierID	AverageDeliveryTime
1	21.0000
2	57.0000
3	106.0000
4	144.0000

The status bar at the bottom right indicates the time is 3:13 PM and the date is 12/12/2023.

Q11. List all packages with a specific weight range:

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

- Query Editor:** Contains the following SQL code:

```
L281 • SELECT * FROM Courier WHERE Status = 'Delivered';
L282
L283 • SELECT CourierID, COUNT(*) AS TotalPackages FROM Courier GROUP BY CourierID;
L284
L285
L286 • SELECT CourierID, AVG(DATEDIFF(DeliveryDate, '2022-12-25')) AS AverageDeliveryTime
L287     FROM Courier GROUP BY CourierID;
L288
L289 • SELECT * FROM Courier WHERE Weight BETWEEN 1.00 AND 2.00;
```
- Result Grid:** Displays the results of the last query, showing a single row of data for a package with CourierID 3.

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
3	SenderC	456 Sender St	ReceiverC	202 Receiver St	1.80	Pending	TN34567	NULL
- Output Panel:** Shows the execution log:

```
1 15:24:11 15:24:11 SELECT * FROM Courier WHERE Weight BETWEEN 1.00 AND 2.00 LIMIT 0, 1000
2 15:24:11 15:24:11 SELECT * FROM Courier WHERE Weight BETWEEN 1.00 AND 2.00;
```
- System Bar:** Shows the date and time as 12/12/2023 3:24 PM.

Q12. Retrieve employees whose names contain 'John'

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

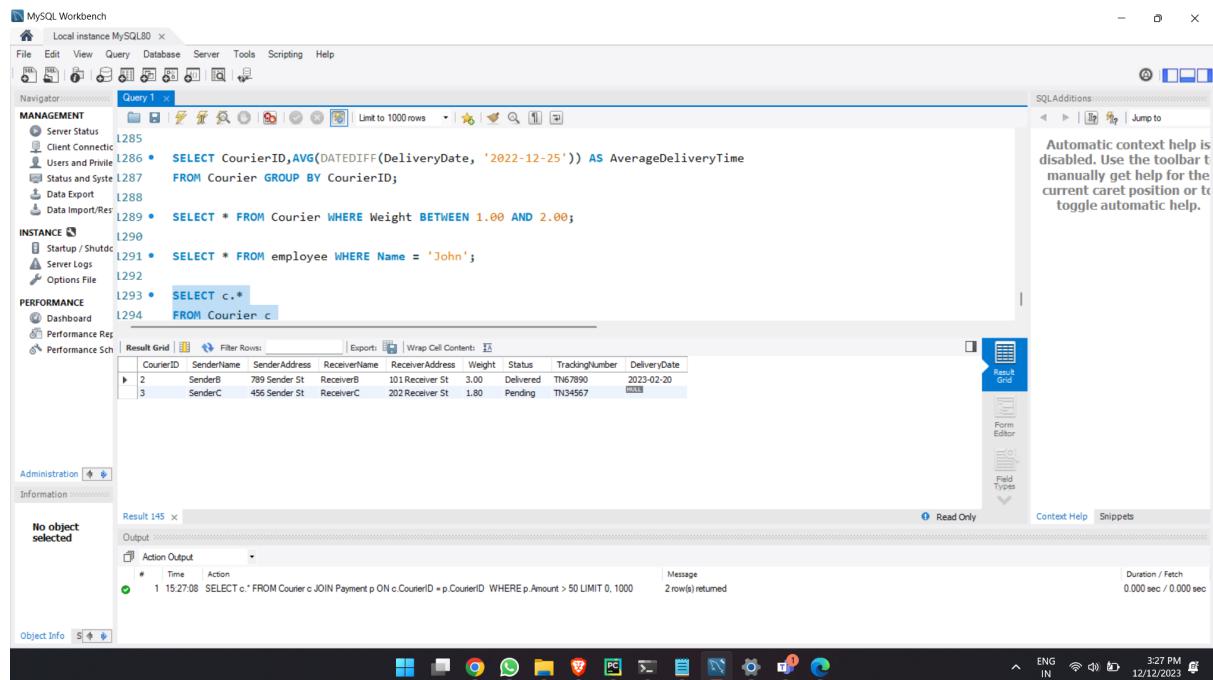
- Query Editor:** Contains the following SQL code:

```
L283 • SELECT CourierID, COUNT(*) AS TotalPackages FROM Courier GROUP BY CourierID;
L284
L285
L286 • SELECT CourierID, AVG(DATEDIFF(DeliveryDate, '2022-12-25')) AS AverageDeliveryTime
L287     FROM Courier GROUP BY CourierID;
L288
L289 • SELECT * FROM Courier WHERE Weight BETWEEN 1.00 AND 2.00;
L290
L291 • SELECT * FROM employee WHERE Name = 'John';
```
- Result Grid:** Displays the results of the last query, showing a single row of data for an employee named John.

EmployeeID	Name	Email	ContactNumber	Role	Salary
*	John	NULL	NULL	NULL	NULL
- Output Panel:** Shows the execution log:

```
1 15:24:11 15:24:11 SELECT * FROM Courier WHERE Weight BETWEEN 1.00 AND 2.00 LIMIT 0, 1000
2 15:25:56 15:25:56 SELECT * FROM employee WHERE Name = 'John' LIMIT 0, 1000
```
- System Bar:** Shows the date and time as 12/12/2023 3:25 PM.

Q13. Retrieve all courier records with payments greater than \$50.



The screenshot shows the MySQL Workbench interface with a query editor window titled "Query1". The SQL code entered is:

```
L286 • SELECT CourierID, AVG(DATEDIFF(DeliveryDate, '2022-12-25')) AS AverageDeliveryTime
L287     FROM Courier GROUP BY CourierID;
L288
L289 • SELECT * FROM Courier WHERE Weight BETWEEN 1.00 AND 2.00;
L290
L291 • SELECT * FROM employee WHERE Name = 'John';
L292
L293 • SELECT c.*
L294     FROM Courier c
```

The results grid displays the following data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
2	SenderB	789 Sender St	ReceiverB	101 Receiver St	3.00	Delivered	TN67890	2023-02-20
3	SenderC	456 Sender St	ReceiverC	202 Receiver St	1.80	Pending	TN34567	NULL

The status bar at the bottom right indicates the session is "Read Only".

Task 3: GroupBy, Aggregate Functions, Having, Order By, where

Q14. Find the total number of couriers handled by each employee.

There is no column to connect with employeeID in Couriers

Q15. Calculate the total revenue generated by each location

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
L348 • SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalAmountReceived
  FROM Location l
  JOIN Payment p ON l.LocationID = p.LocationID
 WHERE p.PaymentDate > '2023-05-30'
 GROUP BY l.LocationID, l.LocationName
 HAVING TotalAmountReceived > 50;
```

The result grid shows the following data:

LocationID	LocationName	TotalRevenue
1	LocationA	35.00
2	LocationB	55.00
3	LocationC	60.00
4	LocationD	35.00
5	LocationE	40.00

The status bar at the bottom right indicates the time as 4:34 PM and the date as 12/12/2023.

Q16. Find the total number of couriers delivered to each location.

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
L365 1.LocationID, 1.LocationName;
L368 • SELECT 1.LocationID, 1.LocationName, COUNT(c.CourierID) AS TotalCouriersDelivered
  FROM Location 1
  LEFT JOIN Payment p ON 1.LocationID = p.LocationID
  LEFT JOIN Courier c ON p.CourierID = c.CourierID
 GROUP BY 1.LocationID, 1.LocationName;
```

The result grid shows the following data:

LocationID	LocationName	TotalCouriersDelivered
1	LocationA	1
2	LocationB	1
3	LocationC	1
4	LocationD	1
5	LocationE	1

The status bar at the bottom right indicates the time as 4:35 PM and the date as 12/13/2023.

Q17. Find the courier with the highest average delivery time:

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

- Query Editor:** Displays the SQL query:


```
L312 • SELECT c.CourierID, AVG(DATEDIFF(c.DeliveryDate,'2023-01-01')) AS AverageDeliveryTime
      FROM Courier c
      WHERE c.DeliveryDate IS NOT NULL
      GROUP BY c.CourierID
      ORDER BY AverageDeliveryTime DESC
```
- Result Grid:** Shows the output of the query:

CourierID	AverageDeliveryTime
137	137.0000
- Output Window:** Shows the execution log:


```
1 16:23:24 SELECT c.CourierID, AVG(DATEDIFF(c.DeliveryDate,'2023-01-01')) AS AverageDeliveryTime FROM Courier c WHERE ... 1 row(s) returned
```

Q18. Find Locations with Total Payments Less Than a Certain Amount

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

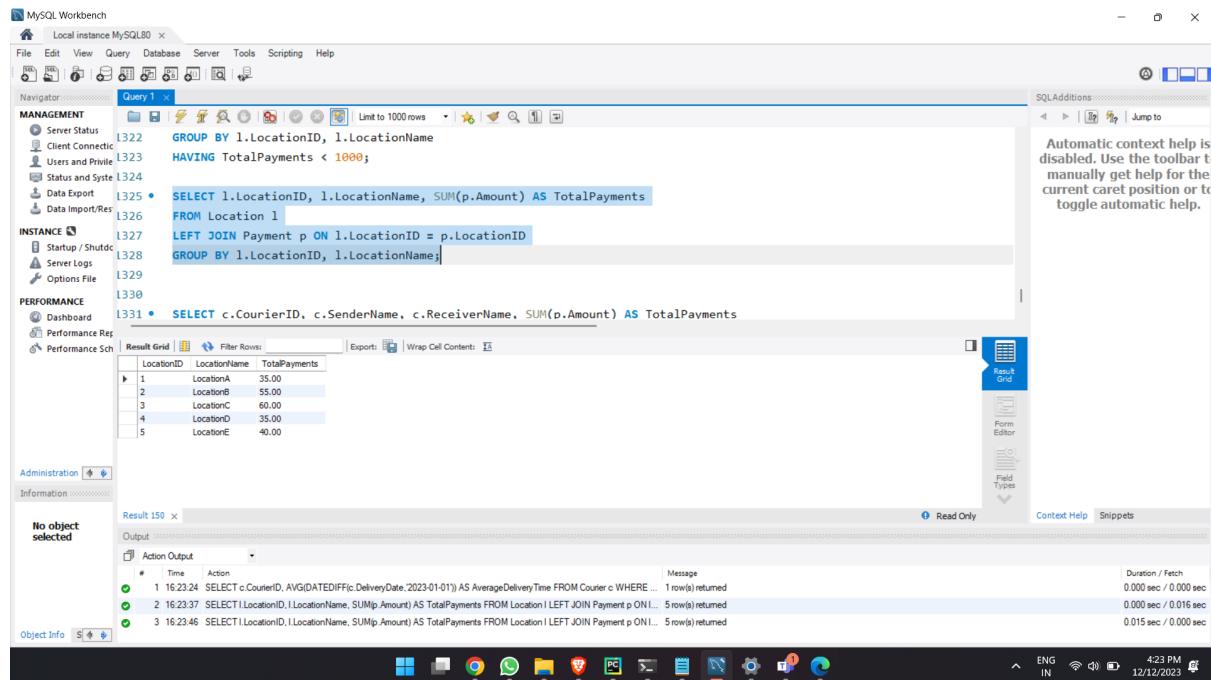
- Query Editor:** Displays the SQL query:


```
L368 • SELECT l.LocationID, l.LocationName, COUNT(c.CourierID) AS TotalCouriersDelivered
      FROM Location l
      LEFT JOIN Payment p ON l.LocationID = p.LocationID
      LEFT JOIN Courier c ON p.CourierID = c.CourierID
      GROUP BY l.LocationID, l.LocationName;
      
      L374 • SELECT l.LocationID, l.LocationName, COALESCE(SUM(p.Amount), 0) AS TotalPayments
      FROM Location l
```
- Result Grid:** Shows the output of the query:

LocationID	LocationName	TotalPayments
1	LocationA	35.00
4	LocationD	35.00
- Output Window:** Shows the execution log:


```
13 16:34:11 SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalRevenue FROM Location l LEFT JOIN Payment p ... 5 row(s) returned
      14 16:35:11 SELECT l.LocationID, l.LocationName, COUNT(c.CourierID) AS TotalCouriersDelivered FROM Location l LE ... 5 row(s) returned
      15 16:37:28 SELECT l.LocationID, l.LocationName, COALESCE(SUM(p.Amount), 0) AS TotalPayments FROM Location l LEFT JO ... 2 row(s) returned
```

Q19. Calculate Total Payments per Location



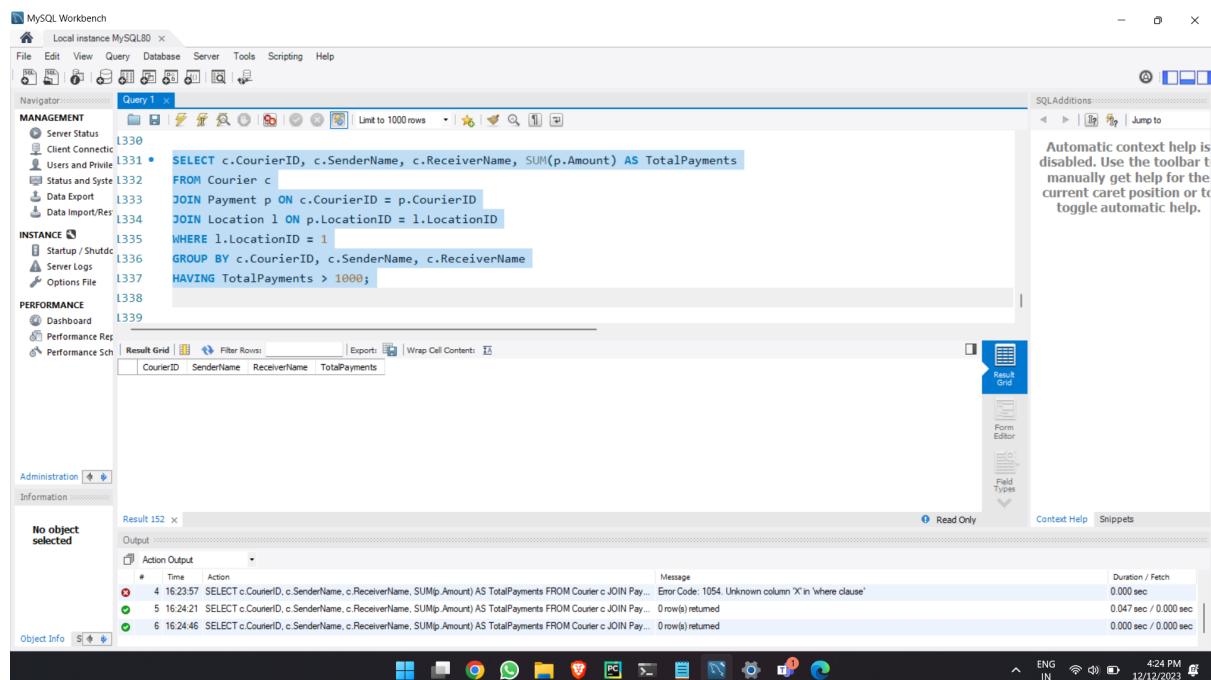
The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains three SQL statements labeled L322, L323, L324, L325, L326, L327, L328, L329, and L330. The last statement (L330) is highlighted.
- Result Grid:** Displays the output of the query:

LocationID	LocationName	TotalPayments
1	LocationA	35.00
2	LocationB	55.00
3	LocationC	60.00
4	LocationD	35.00
5	LocationE	40.00
- Output Window:** Shows the execution log with three entries:

#	Time	Action	Message	Duration / Fetch
1	16:23:24	SELECT c.CourierID, AVG(DATEDIFF(c.DeliveryDate, '2023-01-01')) AS AverageDeliveryTime FROM Courier c WHERE ... 1 row(s) returned		0.000 sec / 0.000 sec
2	16:23:37	SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPayments FROM Location l LEFT JOIN Payment p ON l.LocationID = p.LocationID ... 5 row(s) returned		0.000 sec / 0.016 sec
3	16:23:46	SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID ... 5 row(s) returned		0.015 sec / 0.000 sec

Q20. Retrieve couriers who have received payments totaling more than \$1000 in a specific location (LocationID = X):



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

- Query Editor:** Contains four SQL statements labeled L330, L331, L332, L333, L334, L335, L336, L337, and L338. The last statement (L338) is highlighted.
- Result Grid:** Displays the output of the query:

CourierID	SenderName	ReceiverName	TotalPayments
-----------	------------	--------------	---------------
- Output Window:** Shows the execution log with six entries. The fourth entry (SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID ... Error Code: 1054. Unknown column 'X' in 'where clause' 0 rows(s) returned) contains an error message.

Q21. Retrieve couriers who have received payments totaling more than \$1000 after a certain date (PaymentDate > 'YYYY-MM-DD'):

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator: Query1 x
MANAGEMENT
    • L337 HAVING TotalPayments > 1000;
    • L340 • SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments
        FROM Courier c
        JOIN Payment p ON c.CourierID = p.CourierID
        WHERE p.PaymentDate > '2023-01-30'
        GROUP BY c.CourierID, c.SenderName, c.ReceiverName
        HAVING TotalPayments > 10;
    • L345
    • L346
INSTANCE
    • L342
    • L343
    • L344
    • L345
    • L346
PERFORMANCE
    • Dashboard
    • Performance Rep
    • Performance Sch
Administration
Information
No object selected
Result 156 x
Output:
Action Output
# Time Action
8 16:25:25 SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalAmountReceived
FROM Location l
JOIN Payment p ON l.LocationID = p.LocationID
WHERE p.PaymentDate > '2023-05-30'
GROUP BY l.LocationID, l.LocationName
HAVING TotalAmountReceived > 5000
LIMIT 0, 1000
Duration / Fetch
0.016 sec / 0.000 sec
0.000 sec / 0.000 sec
0.000 sec / 0.000 sec
Object Info S
Result Grid | Filter Rows | Export: | Wrap Cell Content: |
Read Only Context Help Snippets
ENG IN 4:26 PM 12/12/2023

```

Q22. Retrieve locations where the total amount received is more than \$5000 before a certain date (PaymentDate > 'YYYY-MM-DD')

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator: Query1 x
MANAGEMENT
    • L340 • SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments
        FROM Courier c
        JOIN Payment p ON c.CourierID = p.CourierID
        WHERE p.PaymentDate > '2023-01-30'
        GROUP BY c.CourierID, c.SenderName, c.ReceiverName
        HAVING TotalPayments > 10;
    • L348 • SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalAmountReceived
        FROM Location l
    • L349
INSTANCE
    • L345
    • L346
    • L347
PERFORMANCE
    • Dashboard
    • Performance Rep
    • Performance Sch
Administration
Information
No object selected
Result 157 x
Output:
Action Output
# Time Action
9 16:25:35 SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments FROM Courier c JOIN Pay... 4 row(s) returned
10 16:25:47 SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments FROM Courier c JOIN Pay... 4 row(s) returned
11 16:26:39 SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalAmountReceived FROM Location l JOIN Payment p ... 0 row(s) returned
Duration / Fetch
0.000 sec / 0.000 sec
0.000 sec / 0.000 sec
0.000 sec / 0.000 sec
Object Info S
Result Grid | Filter Rows | Export: | Wrap Cell Content: |
Read Only Context Help Snippets
ENG IN 4:26 PM 12/12/2023

```

Task 4: Inner Join, Full Outer Join, Cross Join, Left Outer Join, Right Outer Join

Q23. Retrieve Payments with Courier Information

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The SQL code is:

```
L374 •  SELECT l.LocationID, l.LocationName, COALESCE(SUM(p.Amount), 0) AS TotalPayments
  FROM Location l
  LEFT JOIN Payment p ON l.LocationID = p.LocationID
 GROUP BY l.LocationID, l.LocationName
 HAVING TotalPayments < 40;
L381 •  SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate
  FROM Payment p
```

The result grid displays the following data:

PaymentID	CourierID	SenderName	ReceiverName	Amount	PaymentDate
1	1	SenderA	ReceiverA	35.00	2023-01-30
2	2	SenderB	ReceiverB	55.00	2023-02-25
3	3	SenderC	ReceiverC	60.00	2023-03-30
4	4	SenderD	ReceiverD	35.00	2023-04-30
5	5	SenderE	ReceiverE	40.00	2023-05-30

The status bar at the bottom right shows "5:01 PM 12/12/2023".

Q24. Retrieve Payments with Location Information

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The SQL code is:

```
L377 GROUP BY l.locationID, l.LocationName
L378 HAVING TotalPayments < 40;
L382 •  SELECT p.PaymentID,p.CourierID,l.LocationID,l.LocationName,p.Amount,p.PaymentDate
  FROM Payment p
  LEFT JOIN Location l ON p.LocationID = l.LocationID
L385
```

The result grid displays the following data:

PaymentID	CourierID	LocationID	LocationName	Amount	PaymentDate
1	1	1	LocationA	35.00	2023-01-30
2	2	2	LocationB	55.00	2023-02-25
3	3	3	LocationC	60.00	2023-03-30
4	4	4	LocationD	35.00	2023-04-30
5	5	5	LocationE	40.00	2023-05-30

The status bar at the bottom right shows "5:03 PM 12/12/2023".

Q25. Retrieve Payments with Courier and Location Information

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

- Query Editor:** Contains the SQL query:

```
L378 HAVING TotalPayments < 40;
L382 • SELECT p.PaymentID,p.CourierID,l.LocationID,l.LocationName,p.Amount,p.PaymentDate
      FROM Payment p
      LEFT JOIN Location l ON p.LocationID = l.LocationID;
L386 • SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,l.LocationID,l.LocationName,p.Amount,p.PaymentDate
      FROM Payment p
```
- Result Grid:** Displays the results of the query:

PaymentID	CourierID	SenderName	ReceiverName	LocationID	LocationName	Amount	PaymentDate
1	1	SenderA	ReceiverA	1	LocationA	35.00	2023-01-30
2	2	SenderB	ReceiverB	2	LocationB	55.00	2023-02-25
3	3	SenderC	ReceiverC	3	LocationC	60.00	2023-03-30
4	4	SenderD	ReceiverD	4	LocationD	35.00	2023-04-30
5	5	SenderE	ReceiverE	5	LocationE	40.00	2023-05-30
- Output Panel:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
17	17:01:40	SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate FROM Payment p LEFT JOIN Location l ON p.LocationID = l.LocationID;	5 row(s) returned	0.016 sec / 0.000 sec
18	17:02:57	SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,l.LocationID,l.LocationName,p.Amount,p.PaymentDate FROM Payment p LEFT JOIN Location l ON p.LocationID = l.LocationID;	5 row(s) returned	0.000 sec / 0.000 sec
19	17:04:39	SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,l.LocationID,l.LocationName,p.Amount,p.PaymentDate FROM Payment p	5 row(s) returned	0.000 sec / 0.000 sec

Q26. List all payments with courier details

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

- Query Editor:** Contains the SQL query:

```
L384 LEFT JOIN Location l ON p.LocationID = l.LocationID;
L386 • SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,l.LocationID,l.LocationName,p.Amount,p.PaymentDate
      FROM Payment p
      LEFT JOIN Courier c ON p.CourierID = c.CourierID
      LEFT JOIN Location l ON p.LocationID = l.LocationID;
L392 • SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate
      FROM Payment p
```
- Result Grid:** Displays the results of the query:

PaymentID	CourierID	SenderName	ReceiverName	Amount	PaymentDate
1	1	SenderA	ReceiverA	35.00	2023-01-30
2	2	SenderB	ReceiverB	55.00	2023-02-25
3	3	SenderC	ReceiverC	60.00	2023-03-30
4	4	SenderD	ReceiverD	35.00	2023-04-30
5	5	SenderE	ReceiverE	40.00	2023-05-30
- Output Panel:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	17:06:03	SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate FROM Payment p JOIN Courier c ON p.CourierID = c.CourierID;	5 row(s) returned	0.000 sec / 0.000 sec

Q27. Total payments received for each courier

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

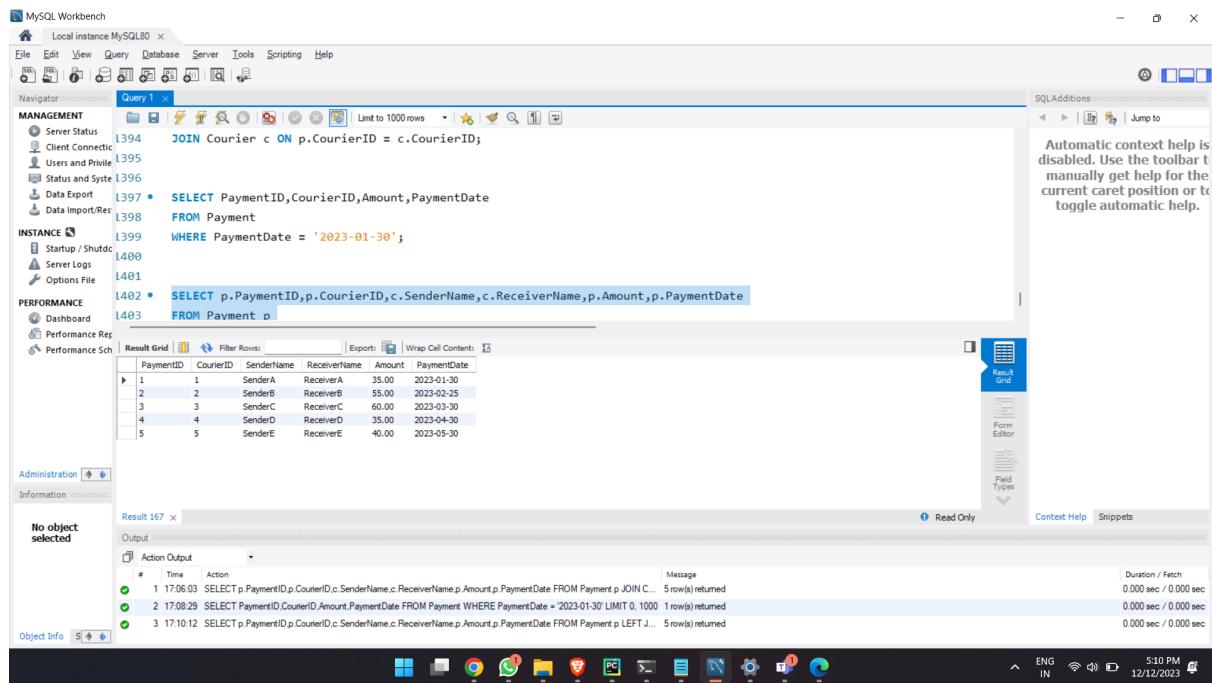
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL icons.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- Query Editor:** Shows two queries:
 - L384: LEFT JOIN Location l ON p.LocationID = l.LocationID;
 - L386: SELECT p.PaymentID, p.CourierID, c.SenderName, c.ReceiverName, l.LocationID, l.LocationName, p.Amount, p.PaymentDate FROM Payment p LEFT JOIN Courier c ON p.CourierID = c.CourierID LEFT JOIN Location l ON p.LocationID = l.LocationID;
 - L388: LEFT JOIN Courier c ON p.CourierID = c.CourierID
 - L391: SELECT p.PaymentID, p.CourierID, c.SenderName, c.ReceiverName, p.Amount, p.PaymentDate FROM Payment p
- Result Grid:** Displays the results of the final query L391, showing 5 rows of payment data.
- Output Window:** Shows the execution log:
 - Action Output: # Time Action
 - 1 17:06:03 SELECT p.PaymentID, p.CourierID, c.SenderName, c.ReceiverName, p.Amount, p.PaymentDate FROM Payment p JOIN Courier c ON p.CourierID = c.CourierID; 5 row(s) returned
- System Bar:** Shows the date and time (12/12/2023, 5:07 PM), system status (ENG IN), and taskbar icons.

Q28. List payments made on a specific date

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL icons.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- Query Editor:** Shows two queries:
 - L392: SELECT p.PaymentID, p.CourierID, c.SenderName, c.ReceiverName, p.Amount, p.PaymentDate FROM Payment p JOIN Courier c ON p.CourierID = c.CourierID;
 - L397: SELECT PaymentID, CourierID, Amount, PaymentDate FROM Payment WHERE PaymentDate = '2023-01-30';
- Result Grid:** Displays the results of the second query L397, showing 1 row of payment data.
- Output Window:** Shows the execution log:
 - Action Output: # Time Action
 - 1 17:06:03 SELECT p.PaymentID, p.CourierID, c.SenderName, c.ReceiverName, p.Amount, p.PaymentDate FROM Payment p JOIN Courier c ON p.CourierID = c.CourierID; 5 row(s) returned
 - 2 17:08:29 SELECT PaymentID, CourierID, Amount, PaymentDate FROM Payment WHERE PaymentDate = '2023-01-30' LIMIT 0, 1000 1 row(s) returned
- System Bar:** Shows the date and time (12/12/2023, 5:08 PM), system status (ENG IN), and taskbar icons.

Q29. Get Courier Information for Each Payment



The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

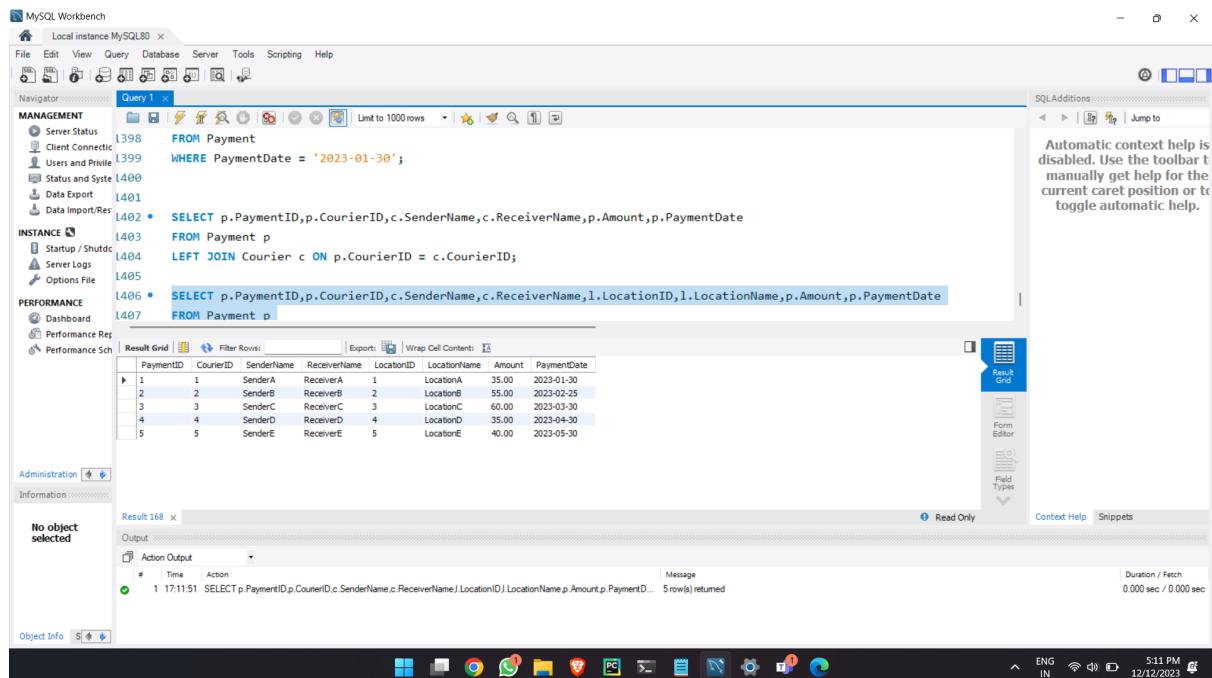
```
L394  JOIN Courier c ON p.CourierID = c.CourierID;
L397 •  SELECT PaymentID,CourierID,Amount,PaymentDate
      FROM Payment
      WHERE PaymentDate = '2023-01-30';
L482 •  SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate
      FROM Payment p
      
```

The result grid displays the following data:

PaymentID	CourierID	SenderName	ReceiverName	Amount	PaymentDate
1	1	SenderA	ReceiverA	35.00	2023-01-30
2	2	SenderB	ReceiverB	55.00	2023-02-25
3	3	SenderC	ReceiverC	60.00	2023-03-30
4	4	SenderD	ReceiverD	35.00	2023-04-30
5	5	SenderE	ReceiverE	40.00	2023-05-30

The status bar at the bottom right shows the time as 5:10 PM and the date as 12/12/2023.

Q30. Get Payment Details with Location



The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
L398  FROM Payment
L399  WHERE PaymentDate = '2023-01-30';
L482 •  SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate
      FROM Payment p
      LEFT JOIN Courier c ON p.CourierID = c.CourierID;
L486 •  SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,l.LocationID,l.LocationName,p.Amount,p.PaymentDate
      FROM Payment p
      
```

The result grid displays the following data:

PaymentID	CourierID	SenderName	ReceiverName	LocationID	LocationName	Amount	PaymentDate
1	1	SenderA	ReceiverA	1	LocationA	35.00	2023-01-30
2	2	SenderB	ReceiverB	2	LocationB	55.00	2023-02-25
3	3	SenderC	ReceiverC	3	LocationC	60.00	2023-03-30
4	4	SenderD	ReceiverD	4	LocationD	35.00	2023-04-30
5	5	SenderE	ReceiverE	5	LocationE	40.00	2023-05-30

The status bar at the bottom right shows the time as 5:11 PM and the date as 12/12/2023.

Q31. Calculating Total Payments for Each Courier

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
L400 • SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate
L401 FROM Payment p
L402 LEFT JOIN Courier c ON p.CourierID = c.CourierID;
L403
L404
L405 • SELECT c.CourierID,c.SenderName,c.ReceiverName, SUM(p.Amount) AS TotalPayments
L406
L407
L408
L409 FROM Courier c
```

The results grid displays the following data:

CourierID	SenderName	ReceiverName	TotalPayments
1	SenderA	ReceiverA	35.00
2	SenderB	ReceiverB	55.00
3	SenderC	ReceiverC	60.00
4	SenderD	ReceiverD	35.00
5	SenderE	ReceiverE	40.00

The status bar at the bottom right indicates the date and time as 12/12/2023 5:13 PM.

Q32. List Payments Within a Date Range

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
L401 • SELECT p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate
L402 FROM Payment p
L403 LEFT JOIN Courier c ON p.CourierID = c.CourierID;
L404
L405
L406
L407
L408
L409 • SELECT PaymentID,CourierID,Amount,PaymentDate
L410 FROM Payment
```

The results grid displays the following data:

PaymentID	CourierID	Amount	PaymentDate
1	1	35.00	2023-01-30

The status bar at the bottom right indicates the date and time as 12/12/2023 5:16 PM.

Q33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side

Q34. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side

Q35. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side

Q36. List all users and all courier services, showing all possible combinations.

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
L409 • SELECT PaymentID, CourierID, Amount, PaymentDate
  FROM Payment
 WHERE PaymentDate BETWEEN '2023-01-01' AND '2023-02-01';
L410
L411
L412
L413
L414 • SELECT
      u.UserID,
      u.Name AS UserName,
      u.Email,
      u.ContactNumber.
L415
L416
L417
L418
```

The results grid displays the following data:

User ID	Username	Email	Contact Number	Address	Service ID	Service Name	Cost
2	Jane Smith	jane.smith@email.com	9876543210	456 Oak Avenue	2	Express Delivery	20.00
1	John Doe	john.doe@email.com	1234567890	123 Main Street	2	Express Delivery	20.00
5	Michael Davis	michael.d@email.com	7789899999	202 Elm Street	3	Same-Day Delivery	30.00
4	Sarah Miller	sarah.m@email.com	4445566666	101 Maple Drive	3	Same-Day Delivery	30.00
3	Robert Johnson	robert.j@email.com	5551112233	789 Pine Lane	3	Same-Day Delivery	30.00
2	Jane Smith	jane.smith@email.com	9876543210	456 Oak Avenue	3	Same-Day Delivery	30.00
1	John Doe	john.doe@email.com	1234567890	123 Main Street	3	Same-Day Delivery	30.00
5	Michael Davis	michael.d@email.com	7789899999	202 Elm Street	4	Oversight Delivery	25.00
4	Sarah Miller	sarah.m@email.com	4445566666	101 Maple Drive	4	Oversight Delivery	25.00
3	Robert Johnson	robert.j@email.com	5551112233	789 Pine Lane	4	Oversight Delivery	25.00

The execution history at the bottom shows:

Action	Time	Action	Message	Duration / Fetch
6	17:28:21	SELECT e.EmployeeID, e.Name AS EmployeeName, e.Email, e.ContactNumber, e.Role, e.Salary, p...	Error Code: 1054. Unknown column 'p.EmployeeID' in 'on clause'	0.000 sec
7	17:29:57	SELECT c.CourierID, c.SenderName, c.ReceiverName, c.Weight, c.Status, c.TrackingNumber, c.D...	Error Code: 1054. Unknown column 'c.ServiceID' in 'on clause'	0.046 sec
8	17:32:05	SELECT u.UserID, u.Name AS UserName, u.Email, u.ContactNumber, u.Address, cs.ServiceID, cs...	25 row(s) returned	0.062 sec / 0.000 sec

Q37. List all employees and all locations, showing all possible combinations:

The screenshot shows the MySQL Workbench interface with a query results window open. The query executed was:

```
CROSS JOIN
CourierServices cs;
```

The results grid displays data from two tables joined together. The columns include EmployeeID, EmployeeName, Email, ContactNumber, Role, Salary, LocationID, LocationName, Address, and various IDs from the CourierServices table. The data shows multiple rows for each employee, indicating their roles and locations.

EmployeeID	EmployeeName	Email	ContactNumber	Role	Salary	LocationID	LocationName	Address
1	EmployeeA	employeeA@email.com	5556667777	Courier Handler	32000.00	1	LocationA	789 Main Street
2	EmployeeB	employeeB@email.com	4445556666	Customer Service	28000.00	1	LocationA	789 Main Street
3	EmployeeC	employeeC@email.com	3334445555	Manager	50000.00	1	LocationA	789 Main Street
4	EmployeeD	employeeD@email.com	2223334444	Delivery Driver	35000.00	1	LocationA	789 Main Street
5	EmployeeE	employeeE@email.com	1112223333	Courier Handler	30000.00	1	LocationA	789 Main Street
6	EmployeeF	employeeF@email.com	5556667777	Courier Handler	32000.00	2	LocationB	456 Oak Avenue
7	EmployeeG	employeeG@email.com	4445556666	Customer Service	28000.00	2	LocationB	456 Oak Avenue
8	EmployeeH	employeeH@email.com	3334445555	Manager	50000.00	2	LocationB	456 Oak Avenue
9	EmployeeI	employeeI@email.com	2223334444	Delivery Driver	35000.00	2	LocationB	456 Oak Avenue
10	EmployeeJ	employeeJ@email.com	1112223333	Courier Handler	30000.00	2	LocationB	456 Oak Avenue
11	EmployeeK	employeeK@email.com	5556667777	Courier Handler	32000.00	3	LocationC	101 Pine Lane
12	EmployeeL	employeeL@email.com	4445556666	Customer Service	28000.00	3	LocationC	101 Pine Lane
13	EmployeeM	employeeM@email.com	3334445555	Manager	50000.00	3	LocationC	101 Pine Lane
14	EmployeeN	employeeN@email.com	2223334444	Delivery Driver	35000.00	3	LocationC	101 Pine Lane
15	EmployeeO	employeeO@email.com	1112223333	Courier Handler	30000.00	3	LocationC	101 Pine Lane
16	EmployeeP	employeeP@email.com	5556667777	Courier Handler	32000.00	4	LocationD	202 Elm Street
17	EmployeeQ	employeeQ@email.com	4445556666	Customer Service	28000.00	4	LocationD	202 Elm Street
18	EmployeeR	employeeR@email.com	3334445555	Manager	50000.00	4	LocationD	202 Elm Street

Q38. Retrieve a list of couriers and their corresponding sender information (if available)

The screenshot shows the MySQL Workbench interface with a query results window open. The query executed was:

```
L431 • SELECT
```

The results grid displays data from the CourierServices table, specifically the SenderID and ReceiverID columns. The data shows multiple rows for each courier, indicating their sender and receiver information.

CourierID	SenderId	ReceiverID	ReceiverName	SenderUserID	SenderId	SenderName	SenderEmail	SenderContactNumber	SenderAddress
1	SenderId A	Receiver A	REDACTED	SenderId A	SenderId B	Receiver B	REDACTED	REDACTED	REDACTED
2	SenderId B	Receiver B	REDACTED	SenderId B	SenderId C	Receiver C	REDACTED	REDACTED	REDACTED
3	SenderId C	Receiver C	REDACTED	SenderId C	SenderId D	Receiver D	REDACTED	REDACTED	REDACTED
4	SenderId D	Receiver D	REDACTED	SenderId D	SenderId E	Receiver E	REDACTED	REDACTED	REDACTED
5	SenderId E	Receiver E	REDACTED	SenderId E	SenderId F	Receiver F	REDACTED	REDACTED	REDACTED

Q39. Retrieve a list of couriers and their corresponding receiver information (if available):

MySQL Workbench - Local instance MySQL80

Query 1:

```
1433 LEFT JOIN Users u ON c.SenderName = u.Name;
1434
1435
1436
```

Result Grid:

CourierID	SenderName	ReceiverName	SenderId	SenderName	SenderEmail	SenderContactNumber	SenderAddress
1	SenderA	ReceiverA	NULL	NULL	NULL	NULL	NULL
2	SenderB	ReceiverB	NULL	NULL	NULL	NULL	NULL
3	SenderC	ReceiverC	NULL	NULL	NULL	NULL	NULL
4	SenderD	ReceiverD	NULL	NULL	NULL	NULL	NULL
5	SenderE	ReceiverE	NULL	NULL	NULL	NULL	NULL

Output:

```
1 17:41:30 SELECT c.CourierID,c.SenderName,c.ReceiverName,u.UserID AS SenderUserID,u.Name AS SenderName,u.Email AS ... 5 row(s) returned
```

Q40. Retrieve a list of couriers along with the courier service details (if available):

Q41. Retrieve a list of employees and the number of couriers assigned to each employee:

Q42. Retrieve a list of locations and the total payment amount received at each location:

MySQL Workbench - Local instance MySQL80

Query 1:

```
1433 LEFT JOIN Users u ON c.SenderName = u.Name;
1434
1435
1436 • SELECT
```

Result Grid:

LocationID	LocationName	TotalPaymentAmount
3	LocationC	60.00
2	LocationB	55.00
5	LocationE	40.00
1	LocationA	35.00
4	LocationD	35.00

Output:

```
1 17:50:29 SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPaymentAmount FROM Location l LEFT JO... 5 row(s) returned
```

Q43. Retrieve all couriers sent by the same sender (based on SenderName).

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:L450 • SELECT SenderName,COUNT(*) AS NumberOfCouriers, GROUP_CONCAT(CourierID ORDER BY CourierID) AS CourierIDs
FROM Courier
GROUP BY SenderName HAVING COUNT(*) > 1;

```
L451  
L452  
L453
```

The results grid shows the following data:

SenderName	NumberOfCouriers	CourierIDs
SenderA	3	CourierID1, CourierID2, CourierID3

On the right side of the interface, there is a vertical toolbar titled "SQLAdditions" with various icons for managing queries.

Q44. List all employees who share the same role.

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:L449
L450 • SELECT Role, GROUP_CONCAT(Name ORDER BY EmployeeID) AS Employees
FROM Employee
GROUP BY Role HAVING COUNT(*) > 1;

```
L451  
L452  
L453  
L454  
L455 •  
L456  
L457  
L458
```

The results grid shows the following data:

Role	Employees
RoleA	EmployeeA,EmployeeB,EmployeeC

On the right side of the interface, there is a vertical toolbar titled "SQLAdditions" with various icons for managing queries.

Q45. Retrieve all payments made for couriers sent from the same location.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
L456     FROM Employee
L457     GROUP BY Role HAVING COUNT(*) > 1;
L458
L459 •  SELECT l.LocationName,p.PaymentID,c.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate
L460     FROM Payment p
L461     JOIN Courier c ON p.CourierID = c.CourierID
L462     JOIN Location l ON c.SenderAddress = l.Address
L463
L464     ORDER BY l.LocationName, p.PaymentDate;
```

The results grid shows columns: LocationName, PaymentID, CourierID, SenderName, ReceiverName, Amount, and PaymentDate. The output pane shows the execution log with three rows:

#	Time	Action	Message	Duration / Fetch
5	17:54:58	SELECT Role, GROUP_CONCAT(Name ORDER BY EmployeeID) AS Employees FROM Employee GROUP BY Role ...	1 row(s) returned	0.000 sec / 0.000 sec
6	17:56:21	SELECT l.LocationName, p.PaymentID, c.CourierID, c.SenderName, c.ReceiverName, p.Amount, p... 0 row(s) returned		0.000 sec / 0.000 sec
7	18:02:17	SELECT l.LocationName,p.PaymentID,c.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate FROM ... 0 row(s) returned		0.000 sec / 0.000 sec

Q46. Retrieve all couriers sent from the same location (based on SenderAddress).

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
L460     FROM Payment p
L461     JOIN Courier c ON p.CourierID = c.CourierID
L462     JOIN Location l ON c.SenderAddress = l.Address
L463
L464     ORDER BY l.LocationName, p.PaymentDate;
```

The results grid shows columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliveryDate, and SenderLocation. The output pane shows the execution log with three rows:

#	Time	Action	Message	Duration / Fetch
7	18:02:17	SELECT l.LocationName,p.PaymentID,c.CourierID,c.SenderName,c.ReceiverName,p.Amount,p.PaymentDate FROM ... 0 row(s) returned		0.000 sec / 0.000 sec
8	18:04:12	SELECT c1.CourierID, c1.SenderName, c1.SenderAddress, c1.ReceiverName, c1.ReceiverAddress, c1.Weight, c1.Status, c1.TrackingNu... 0 row(s) returned		0.000 sec / 0.000 sec
9	18:05:00	SELECT c1.CourierID,c1.SenderName,c1.SenderAddress,c1.ReceiverName,c1.ReceiverAddress,c1.Weight,c1.Status,c1.TrackingNu... 0 row(s) returned		0.000 sec / 0.000 sec

Q47. List employees and the number of couriers they have delivered:

Q48. Find couriers that were paid an amount greater than the cost of their respective courier services

Task 5 Scope: Inner Queries, Non Equi Joins, Equi joins, Exist, Any, All

Q49. Find couriers that have a weight greater than the average weight of all couriers

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

- Query Editor:** Contains two queries:
 - L460: `FROM Payment p`
 - L461: `JOIN Courier c ON p.CourierID = c.CourierID`
 - L462: `JOIN Location l ON c.SenderAddress = l.Address`
 - L463: `ORDER BY l.LocationName, p.PaymentDate;`
 - L466: `SELECT * FROM Courier WHERE Weight > (SELECT AVG(Weight) FROM Courier);`
 - L467: A result grid showing courier data with columns: CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TradingNumber, DeliveryDate.
- Output Window:** Shows the execution log with two rows:
 - Row 1: SELECT c.CourierID, c.SenderName, c.ReceiverName, p.Amount AS PaymentAmount, cs.Cost AS CourierCost. Duration: 0.000 sec.
 - Row 2: SELECT * FROM Courier WHERE Weight > (SELECT AVG(Weight) FROM Courier) LIMIT 0, 1000. Duration: 0.000 sec / 0.000 sec.
- System Bar:** Displays the date and time as 12/12/2023 6:10 PM.

Q50. Find the names of all employees who have a salary greater than the average salary:

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

- Query Editor:** Contains two queries:
 - L460: `FROM Payment p`
 - L461: `JOIN Courier c ON p.CourierID = c.CourierID`
 - L462: `JOIN Location l ON c.SenderAddress = l.Address`
 - L463: `ORDER BY l.LocationName, p.PaymentDate;`
 - L466: `SELECT * FROM Courier WHERE Weight > (SELECT AVG(Weight) FROM Courier);`
 - L468: `SELECT Name FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee);`
 - L469: A result grid showing employee names with a single row: EmployeeC.
- Output Window:** Shows the execution log with three rows:
 - Row 1: SELECT c.CourierID, c.SenderName, c.ReceiverName, p.Amount AS PaymentAmount, cs.Cost AS CourierCost. Duration: 0.000 sec.
 - Row 2: SELECT * FROM Courier WHERE Weight > (SELECT AVG(Weight) FROM Courier) LIMIT 0, 1000. Duration: 0.000 sec / 0.000 sec.
 - Row 3: SELECT Name FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee) LIMIT 0, 1000. Duration: 0.047 sec / 0.000 sec.
- System Bar:** Displays the date and time as 12/12/2023 6:11 PM.

Q51. Find the total cost of all courier services where the cost is less than the maximum cost

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

Query Editor:

```
L462 JOIN Location l ON c.SenderAddress = l.Address
L463 ORDER BY l.LocationName, p.PaymentDate;
L466 • SELECT * FROM Courier WHERE Weight > (SELECT AVG(Weight) FROM Courier);
L467
L468 • SELECT Name FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee);
L469
L470 • SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);
```

Results Grid:

TotalCost
85.00

Information Panel:

No object selected

Action Output:

#	Time	Action	Message	Duration / Fetch
1	18:08:36	SELECT c.CourierID, c.SenderName, c.ReceiverName, p.Amount AS PaymentAmount, cs.Cost AS CourierCost	Error Code: 1054. Unknown column 'c.ServiceID' in 'on clause'	0.000 sec
2	18:09:53	SELECT * FROM Courier WHERE Weight > (SELECT AVG(Weight) FROM Courier) LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
3	18:11:51	SELECT Name FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee) LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec
4	18:12:51	SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices) ...	1 row(s) returned	0.000 sec / 0.000 sec

Q52. Find all couriers that have been paid for

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

Query Editor:

```
L465
L466 • SELECT * FROM Courier WHERE Weight > (SELECT AVG(Weight) FROM Courier);
L467
L468 • SELECT Name FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee);
L469
L470 • SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);
L471
L472
L473 • SELECT c.* FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID;
```

Results Grid:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	SenderA	123 Sender St	ReceiverA	456 Receiver St	2.50	In Transit	TN12345	2023-01-15
2	SenderB	789 Sender St	ReceiverB	101 Receiver St	3.00	Delivered	TN67890	2023-02-20
3	SenderC	456 Sender St	ReceiverC	202 Receiver St	1.80	Pending	TN34567	
4	SenderD	101 Sender St	ReceiverD	303 Receiver St	2.20	In Transit	TN23456	2023-04-10
5	SenderE	202 Sender St	ReceiverE	404 Receiver St	4.50	Delivered	TN78901	2023-05-18

Information Panel:

No object selected

Action Output:

#	Time	Action	Message	Duration / Fetch
2	18:09:53	SELECT * FROM Courier WHERE Weight > (SELECT AVG(Weight) FROM Courier) LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
3	18:11:51	SELECT Name FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee) LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec
4	18:12:51	SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices) ...	1 row(s) returned	0.000 sec / 0.000 sec
5	18:15:06	SELECT c.* FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Q53. Find the locations where the maximum payment amount was made

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

- Query Editor:** Contains the following SQL code:


```
L468 •  SELECT Name FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee);
L469 •  SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);
L470 •  SELECT c.* FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID;
L471
L472
L473 •  SELECT c.* FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID;
L474
L475 •  SELECT 1.* , MAX(p.Amount) AS MaxPaymentAmount;
```
- Result Grid:** Displays the results of the final query, showing five rows of data:

LocationID	LocationName	Address	MaxPaymentAmount
1	LocationA	789 Main Street	35.00
2	LocationB	456 Oak Avenue	55.00
3	LocationC	101 Pine Lane	60.00
4	LocationD	202 Elm Street	35.00
5	LocationE	303 Cedar Road	40.00
- Information Panel:** Shows the following output:

Action	Time	Action	Message	Duration / Fetch
3	18:11:51	SELECT Name FROM Employee WHERE Salary > (SELECT AVG(Salary) FROM Employee) LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec
4	18:12:51	SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices)	1 row(s) returned	0.000 sec / 0.000 sec
5	18:15:06	SELECT c.* FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
6	18:16:26	SELECT 1.* , MAX(p.Amount) AS MaxPaymentAmount FROM Location l JOIN Payment p ON l.LocationID = p.LocationID	5 row(s) returned	0.000 sec / 0.000 sec

Q54. Find all couriers whose weight is greater than the weight of all couriers sent by a specific sender (e.g., 'SenderName'):

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

- Query Editor:** Contains the following SQL code:


```
L479
L480 •  SELECT c.*
FROM Courier c
WHERE c.Weight > (
    SELECT MAX(c2.Weight)
    FROM Courier c2
    WHERE c2.SenderName = 'SenderName'
);
```
- Result Grid:** Displays the results of the query, showing one row of data:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1000	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
- Information Panel:** Shows the following output:

Action	Time	Action	Message	Duration / Fetch
4	18:12:51	SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices)	1 row(s) returned	0.000 sec / 0.000 sec
5	18:15:06	SELECT c.* FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
6	18:16:26	SELECT 1.* , MAX(p.Amount) AS MaxPaymentAmount FROM Location l JOIN Payment p ON l.LocationID = p.LocationID	5 row(s) returned	0.000 sec / 0.000 sec
7	18:17:21	SELECT c.* FROM Courier c WHERE c.Weight > (SELECT MAX(c2.Weight) FROM Courier c2 WHERE c2.S...	0 row(s) returned	0.000 sec / 0.000 sec