

INFO 5707-PROJECT FINAL REPORT

GROUP NUMBER – 13 :

Rohith Singh Thakur (Team Coordinator)
Varshith Bala
Vamshi Kodali

Title of the project: Postal Mail Tracking and Delivery System

Objectives:

- Implement a formal postal delivery tracking system using SQL to manage package logistics and customer data effectively.
- Implement tracking of packages with various delivery statuses including "Pending," "In Transit," "Out for Delivery," "Delivered," and "Failed."
- Log each delivery attempt with accurate timestamps and reasons, providing delivery accountability and traceability.
- Securely store proof of delivery using either customer signatures or image uploads to verify successful handovers.
- Create a normalized relational database schema and represent relationships between entities via a thorough ER diagram.
- Implement feedback capabilities to allow customers to rate their delivery experience and optionally comment for improvement reasons.
- Calculate total shipping fees considering various cost metrics like base fee, package weight, and travelling distance.
- Develop dynamic SQL queries to retrieve valuable information like failed deliveries, packages dispatched for delivery, and billing of customers.
- Ensure that the system supports user requirements like transparency, capture of feedback, secure storage of data, and notification functions.
- Create a platform that can be scaled to support future enhancements like QR code tracking, SMS/email notifications, visual route mapping, and role-based user access.

Scope:

- Design a relational database with structure to facilitate a postal delivery tracking system.
- Create and deploy SQL tables that establish basic entities such as customers, packages, delivery status, routes, and feedback.

- Insert sample data into all the tables to simulate real postal delivery operations.
- Establish relationships between entities using foreign keys to maintain referential integrity.
- Write SQL commands to query information such as failed deliveries, shipping charges, and delivery statuses.
- Use join and aggregation operations to analyze customer activity, package shipping, and cost aggregates.
- Demonstrate the functionality of the system only through SQL operations without UI or application layer.
- Handle only backend data modeling, normalization, and query logic based on project requirements.
- Mock significant business processes (e.g., delivery tracking and cost calculation) on the basis of pre-inserted database records.
- Ensure the design is scalable and can be extended in the future with frontend and real-time considerations.

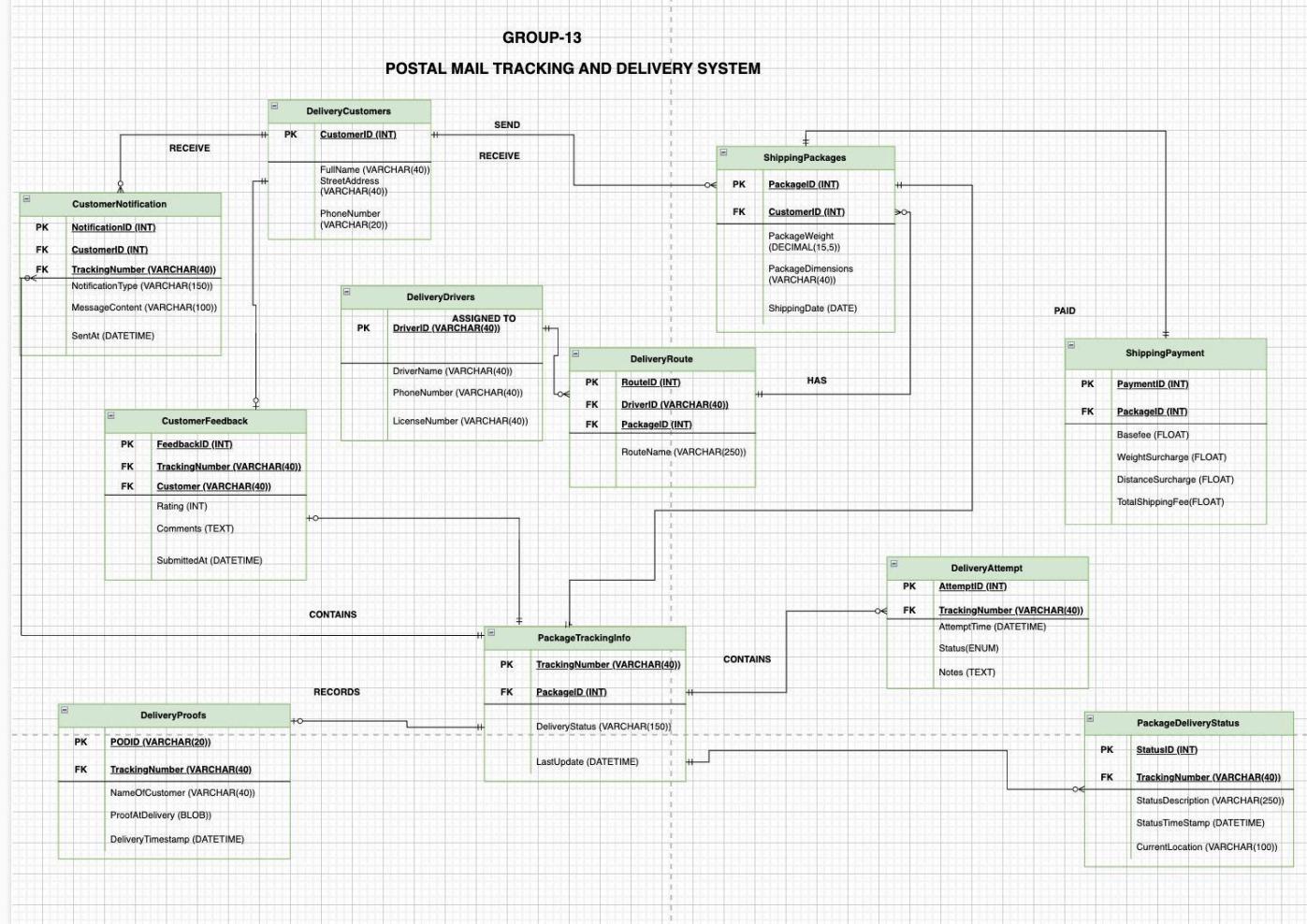
Business Rules:

- The system must log and present package delivery status such as "Out for Delivery," "Delivered," "Delayed," or "Failed."
- Every shipment status update must be time stamped and queryable by SQL queries.
- The system must store proof of delivery in a signature or image with the timestamp of delivery.
- Customers may leave feedback, which may include a rating and an optional comment after delivery.
- The total shipping fee must be calculated on a combination of base cost, package weight, and delivery distance.
- Each tracking number must be unique and must reference consistently across similar tables (e.g., tracking data, delivery status, notifications).
- A package must be allocated to a single customer, but a customer can have multiple packages.
- Each delivery must be assigned to a specific driver and route, and each route can take multiple deliveries.
- Every attempt at delivery must be recorded with the outcome and a reason for success or failure.
- Feedback records must only be entered after delivery has been marked as completed, for data integrity.

User Requirements:

- The system will record and display delivery status updates such as “Out for Delivery,” “Delivered,” or “Delayed.”
- Shipment statuses and their last updated times must be retrievable through SQL queries by administrators.
- The system must securely store proof of delivery, such as a digital signature or delivery photo with a timestamp.
- Customers should be able to leave feedback regarding their delivery experience, including a rating and optional comments.
- The system must calculate and display the total shipping fee, summing up base cost, weight surcharge, and distance surcharge.
- The system ought to have sample record insertion to simulate customer, package, delivery, and payment data.
- The database ought to maintain referential integrity through primary and foreign key constraints.
- The system ought to have SQL grouping and joins to enable complex administrative queries.
- The users ought to be able to produce reports that show delivery statistics, failed attempts, and package distribution.
- The architecture must be future-proofed for scalability to allow for future features such as QR tracking or notification triggers.

Entity Relationship Diagram:



Data Dictionary:

S.No	Table Name	Attribute Name	Description	Type	Format	Range	Required	PK/FK	Example
1	DeliveryCustomers	CustomerID	Unique ID for a Customer	INT	99999	100-9999	YES	PK	1408-06-24
		FullName	Customer FullName	VARCHAR(40)	XXXXXX		YES		Grop 13
		StreetAddress	Customer Street Address	VARCHAR(40)	XXXXXX		YES		Union Eagle Post
		PhoneNumber	Customer Phone Number	VARCHAR(40)	XXXXXX		YES		940 999 9999
2	CustomerNotifications	NotificationID	Unique ID for a Notification	INT	99999	100-99999	YES	PK	1200-09-30
		CustomerID	Unique ID of a Customer	INT	99999	100-100000	YES	FK	1200-09-31
		TrackingNumber	Number of a Tracking Number	VARCHAR(20)	XXXXXX	100-10000	YES	FK	UNR719
		NotificationType	Unique Type of a Notification Type	ENUM	XXXXXX		YES		DELAYED
3	ShippingPackages	MessageContent	Customer Message Content	VARCHAR(100)	XXXXXX		YES		CAN LOCATE
		SentAt	Timestamp of a Sent Timestamp	DATE TIME	YYYY-MM-DD		YES		2/2/25
		PackageID	Unique ID of a Package	INT	99999	100-99999	YES	PK	1200-03-55
		CustomerID	Unique ID for a Customer	INT	99999	100-100000	YES	FK	1200-03-56
4	DeliveryDrivers	PackageWeight	Weight of the Package	DECIMAL(15,5)	XXXXXX		YES		1200-03-57
		PackageDimensions	Dimensions of the Package	VARCHAR(40)	XXXXXX		YES		10/15
		ShippingDate	Unique Date for a Shipping Date	DATE	YYYY-MM-DD		YES		3/23/25
		DriverID	Unique ID for a driver	VARCHAR(40)	XXXXXX		YES	PK	1200-09-20
5	DeliveryRoutes	DriverName	DriverName	VARCHAR(40)	XXXXXX		YES		STEVE
		PhoneNumber	Phone Number of Driver	VARCHAR(15)	XXXXXX		YES		99999999
		LicenseNumber	Driver License Number	VARCHAR(10)	XXXXXX		YES		INFAS6
		RouteID	Unique ID for a Route	INT	99999	100-99999	YES	PK	123456
6	PackageTrackingInfo	DriverID	ID for a driver	VARCHAR(40)	XXXXXX		YES	FK	RST123
		PackageID	ID for a Package	INT	99999	100-99999	YES	FK	8888
		RouteName	Unique Name for a Route Name	VARCHAR(250)	XXXXXX		YES		EAGLE
		TypeInCarrier	Unique Number for Tracking Package	VARCHAR(30)	XXXXXX		YES	PK	12/12/25
7	PackageDeliveryStatus	PackageID	ID for a Package	INT	100-99999		YES	FK	99999
		DeliveryStatus	Position of Package	VARCHAR(150)	XXXXXX		YES		In Transit
		LastUpdated	Updated Datetime	YYYY-MM-DD	XXXXXX		YES		3/26/25
		StatusID	Unique ID for a status	INT	99999	1000-99999	YES	PK	2224
8	DeliveryProofs	TrackingNumber	Number for Tracking	VARCHAR(40)	XXXXXX		YES	FK	1ZY678
		NameOfCustomer	Description for a status	VARCHAR(250)	XXXXXX		YES		OUT FOR DELIVERY
		StatusTimeStamp	Unique Stamp for a TimeStamp	DATE TIME	YYYY-MM-DD		YES		3/27/25
		CurrentLocation	Delivery Location	VARCHAR(100)	XXXXXX		YES		EAGLE POST
9	ShippingPayments	DeliveryTimeStamp	Signature or Photo of the Customer	BLOB	XXXXXX		YES		1ZNY67
		TrackingNumber	Proof of Delivery ID	VARCHAR(20)	XXXXXX		YES	PK	1ZNY67
		NameOfCustomer	Number for Tracking	VARCHAR(20)	XXXXXX		YES	FK	STEVE
		ProofOfDelivery	Unique Customer for a NameOfCustomer	VARCHAR(40)	XXXXXX		YES		
10	DeliveryAttempt	DeliveryTimeStamp	Unique Time Stamp at delivery	DATE TIME	YYYY-MM-DD		YES		3/28/25
		PaymentID	Unique ID for a Payment	INT	99999	100-9999999	YES	PK	5526622
		PackageID	Unique ID for a Package	INT	99999	100-9999999	YES	FK	5555555
		BaseFee	Basic Fee for delivery	FLOAT	99999	100-99999	YES		57.89
11	CustomerFeedback	weightSurcharge	Charger for weight of Package	FLOAT	99999	100-99999	YES		6.50
		DistanceSurcharge	Charger for distance travelled	FLOAT	99999	100-99999	YES		2.56
		TotalShippingFee	Total Shipping Fee	INT	99999	100-99999	YES		67.33
		AttemptID	Delivery Attempt ID	VARCHAR(30)	XXXXXX		YES	PK	45756
12	CustomerFeedback	TrackingNumber	Number for Tracking	VARCHAR(30)	XXXXXX		YES	FK	XBBHWQ63
		AttemptTime	Delivery Attempt Time	DATE TIME	YYYY-MM-DD		YES		2/12/24
		Status	Status of the Delivery	ENUM	XXXXXX		YES		FAILED
		Notes	Any notes for improving delivery	TEXT	XXXXXX		YES		Customer did not sign
13	CustomerFeedback	FeedbackID	Unique Feedback ID	INT	9999		YES	PK	5458
		CustomerID	Unique Customer number	INT	9999		YES	FK	45612
		TrackingNumber	Number for Tracking	VARCHAR(30)	XXXXXX		YES	FK	SDFR4521
		Rating	Rating from 1 to 5 from customer	INT	99999		YES		5
14	CustomerFeedback	Comments	Comments provided by customer	TEXT	XXXXXX		YES		Good
		SubmittedAt	Feedback time submitted by customer	DATE TIME	YYYY-MM-DD		YES		12/25/24

Entity Generation and Data Entry

1. DeliveryCustomers :

- **Purpose:** Stores customer information.
- **Fields:** CustomerID, FullName, StreetAddress, PhoneNumber.
- **Entries:** 20 customers inserted.

```
CREATE TABLE DeliveryCustomers (
```

```
    `CustomerID` INT NOT NULL,  
    `FullName` VARCHAR(40) NOT NULL,  
    `StreetAddress` VARCHAR(40) NOT NULL,  
    `PhoneNumber` VARCHAR(40) NOT NULL,  
    PRIMARY KEY (`CustomerID`)
```

```
);
```

Insert Statements:

```
INSERT INTO DeliveryCustomers VALUES (1001, 'Alice Johnson', '1 Elm St', '940-000-1001');  
INSERT INTO DeliveryCustomers VALUES (1002, 'Michael Smith', '2 Elm St', '940-000-1002');  
INSERT INTO DeliveryCustomers VALUES (1003, 'Sophia Martinez', '3 Elm St', '940-000-1003');  
INSERT INTO DeliveryCustomers VALUES (1004, 'David Lee', '4 Elm St', '940-000-1004');  
INSERT INTO DeliveryCustomers VALUES (1005, 'Emily Nguyen', '5 Elm St', '940-000-1005');  
INSERT INTO DeliveryCustomers VALUES (1006, 'Daniel Thompson', '6 Elm St', '940-000-1006');  
INSERT INTO DeliveryCustomers VALUES (1007, 'Olivia Brown', '7 Elm St', '940-000-1007');  
INSERT INTO DeliveryCustomers VALUES (1008, 'James Wilson', '8 Elm St', '940-000-1008');  
INSERT INTO DeliveryCustomers VALUES (1009, 'Ava Patel', '9 Elm St', '940-000-1009');  
INSERT INTO DeliveryCustomers VALUES (1010, 'William Clark', '10 Elm St', '940-000-1010');  
INSERT INTO DeliveryCustomers VALUES (1011, 'Isabella Hernandez', '11 Elm St',  
    '940-000-1011');  
INSERT INTO DeliveryCustomers VALUES (1012, 'Benjamin White', '12 Elm St', '940-000-1012');
```

```

INSERT INTO DeliveryCustomers VALUES (1013, 'Mia Robinson', '13 Elm St', '940-000-1013');

INSERT INTO DeliveryCustomers VALUES (1014, 'Lucas King', '14 Elm St', '940-000-1014');

INSERT INTO DeliveryCustomers VALUES (1015, 'Charlotte Wright', '15 Elm St', '940-000-1015');

INSERT INTO DeliveryCustomers VALUES (1016, 'Henry Baker', '16 Elm St', '940-000-1016');

INSERT INTO DeliveryCustomers VALUES (1017, 'Amelia Scott', '17 Elm St', '940-000-1017');

INSERT INTO DeliveryCustomers VALUES (1018, 'Elijah Turner', '18 Elm St', '940-000-1018');

INSERT INTO DeliveryCustomers VALUES (1019, 'Grace Adams', '19 Elm St', '940-000-1019');

INSERT INTO DeliveryCustomers VALUES (1020, 'Logan Evans', '20 Elm St', '940-000-1020');

```

Query:

Now we need to display the information which was inserted into the table.

SELECT * FROM DeliveryCustomers

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

- Navigator:** Shows the schema `info_5707_project` containing tables, views, stored procedures, and functions.
- SQL Editor:** Contains the following SQL code:

```

18 INSERT INTO DeliveryCustomers VALUES (1018, 'Elijah Turner', '18 Elm St', '940-000-1018');
19 INSERT INTO DeliveryCustomers VALUES (1019, 'Grace Adams', '19 Elm St', '940-000-1019');
20 INSERT INTO DeliveryCustomers VALUES (1020, 'Logan Evans', '20 Elm St', '940-000-1020');
21
22 • SELECT * FROM DeliveryCustomers

```
- Result Grid:** Displays the results of the `SELECT * FROM DeliveryCustomers` query, showing 20 rows of data with columns: CustomerID, FullName, StreetAddress, and PhoneNumber.
- Output:** Shows the execution details:
 - Action Output: 61 rows affected (Time: 22:44:42, Action: INSERT INTO DeliveryCustomers VALUES (1020, 'Logan Evans', '20 Elm St', '940-000-1020'))
 - Message: 1 row(s) affected (Duration / Fetch: 0.000 sec / 0.000 rows / 0.000 bytes)
- System:** Shows the system status with a temperature of 16°C, a clear sky icon, and the date/time 4/20/2025 10:46 PM.

2. Customer Notifications

- **Purpose:** Tracks notifications sent to customers about their packages.
- **Fields:** NotificationID, CustomerID, TrackingNumber, NotificationType, MessageContent, SentAt.
- **Entries:** 20 notifications inserted.

```

CREATE TABLE CustomerNotifications (
    `NotificationID` INT NOT NULL,
    `CustomerID` INT NOT NULL,
    `TrackingNumber` VARCHAR(20) NOT NULL,
    `NotificationType` ENUM('TO BE DELIVERED SOON', 'DELAYED', 'OUT FOR
DELIVERY', 'DELIVERED') NOT NULL,
    `MessageContent` VARCHAR(100) NOT NULL,
    `SentAt` DATETIME NOT NULL,
    PRIMARY KEY (`NotificationID`)
);

```

Insert Statements :

```
INSERT INTO CustomerNotifications VALUES (3001, 1001, 'TRK0001', 'DELIVERED', 'Message
Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3002, 1002, 'TRK0002', 'DELIVERED', 'Message
Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3003, 1003, 'TRK0003', 'OUT FOR DELIVERY',
'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3004, 1004, 'TRK0004', 'OUT FOR DELIVERY',
'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3005, 1005, 'TRK0005', 'TO BE DELIVERED
SOON', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3006, 1006, 'TRK0006', 'DELIVERED', 'Message
Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3007, 1007, 'TRK0007', 'TO BE DELIVERED
SOON', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3008, 1008, 'TRK0008', 'TO BE DELIVERED
SOON', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3009, 1009, 'TRK0009', 'DELIVERED', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3010, 1010, 'TRK0010', 'DELIVERED', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3011, 1011, 'TRK0011', 'DELIVERED', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3012, 1012, 'TRK0012', 'TO BE DELIVERED SOON', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3013, 1013, 'TRK0013', 'TO BE DELIVERED SOON', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3014, 1014, 'TRK0014', 'DELIVERED', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3015, 1015, 'TRK0015', 'TO BE DELIVERED SOON', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3016, 1016, 'TRK0016', 'DELIVERED', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3017, 1017, 'TRK0017', 'OUT FOR DELIVERY', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3018, 1018, 'TRK0018', 'TO BE DELIVERED SOON', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3019, 1019, 'TRK0019', 'DELAYED', 'Message Text', '2025-03-23 10:00:00');
```

```
INSERT INTO CustomerNotifications VALUES (3020, 1020, 'TRK0020', 'DELAYED', 'Message Text', '2025-03-23 10:00:00');
```

Query:

Now we need to display the information which was inserted into the table.

```
SELECT * FROM CustomerNotifications
```

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, info_5707_project.
- Navigator:** Shows the schema structure for info_5707_project, including tables, views, stored procedures, and functions.
- SQL Editor:** Contains the SQL code for inserting 13 rows into the CustomerNotifications table and a SELECT statement to retrieve all rows.
- Result Grid:** Displays the 13 rows inserted, with columns: NotificationID, CustomerID, TrackingNumber, ReleaseType, MessageText, and LastUpdated.
- Output:** Shows the execution log with 8 entries, each detailing an INSERT operation and a SELECT operation.
- Object Info:** Shows the status of the table.
- Session:** Shows the current session details.
- System:** Shows system status like CPU usage (16°C), memory (Clear), and network.

NotificationID	CustomerID	TrackingNumber	ReleaseType	MessageText	LastUpdated
3001	1001	TRK0001	DELIVERED	Message Text	2025-03-23 10:00:00
3002	1002	TRK0002	DELIVERED	Message Text	2025-03-23 10:00:00
3003	1003	TRK0003	OUT FOR DELIVERY	Message Text	2025-03-23 10:00:00
3004	1004	TRK0004	TO BE DELIVERED	Message Text	2025-03-23 10:00:00
3005	1005	TRK0005	TO BE DELIVERED SOON	Message Text	2025-03-23 10:00:00
3006	1006	TRK0006	DELIVERED	Message Text	2025-03-23 10:00:00
3007	1007	TRK0007	TO BE DELIVERED SOON	Message Text	2025-03-23 10:00:00
3008	1008	TRK0008	TO BE DELIVERED SOON	Message Text	2025-03-23 10:00:00
3009	1009	TRK0009	DELIVERED	Message Text	2025-03-23 10:00:00
3010	1010	TRK0010	DELIVERED	Message Text	2025-03-23 10:00:00
3011	1011	TRK0011	DELIVERED	Message Text	2025-03-23 10:00:00
3012	1012	TRK0012	TO BE DELIVERED SOON	Message Text	2025-03-23 10:00:00

3. ShippingPackage

- **Purpose:** Stores package details sent by customers.
- **Fields:** PackageID, CustomerID, Packageweight, PackageDimensions, ShippingDate.
- **Entries:** 20 packages inserted.

```
CREATE TABLE ShippingPackage (
    `PackageID` INT NOT NULL,
    `CustomerID` INT NOT NULL,
    `Packageweight` DECIMAL(15,5) NOT NULL,
    `PackageDimensions` VARCHAR(40) NOT NULL,
    `ShippingDate` DATE NOT NULL,
    PRIMARY KEY (`PackageID`)
);
```

Insert statements :

```
INSERT INTO ShippingPackage VALUES (2001, 1001, 48.69582, '29x15x24', '2025-03-27');
INSERT INTO ShippingPackage VALUES (2002, 1002, 45.10995, '22x13x10', '2025-03-30');
INSERT INTO ShippingPackage VALUES (2003, 1003, 41.15647, '22x23x29', '2025-03-23');
INSERT INTO ShippingPackage VALUES (2004, 1004, 11.13834, '30x13x15', '2025-03-28');
INSERT INTO ShippingPackage VALUES (2005, 1005, 48.16558, '21x14x22', '2025-03-27');
INSERT INTO ShippingPackage VALUES (2006, 1006, 42.95698, '20x21x11', '2025-03-21');
INSERT INTO ShippingPackage VALUES (2007, 1007, 38.23529, '24x20x10', '2025-03-29');
INSERT INTO ShippingPackage VALUES (2008, 1008, 33.19635, '21x23x23', '2025-03-29');
INSERT INTO ShippingPackage VALUES (2009, 1009, 34.19965, '24x18x16', '2025-03-27');
INSERT INTO ShippingPackage VALUES (2010, 1010, 25.85617, '26x22x23', '2025-03-26');
INSERT INTO ShippingPackage VALUES (2011, 1011, 14.13477, '17x29x18', '2025-03-30');
INSERT INTO ShippingPackage VALUES (2012, 1012, 8.35196, '30x12x30', '2025-03-21');
INSERT INTO ShippingPackage VALUES (2013, 1013, 31.35637, '25x14x29', '2025-03-29');
```

```

INSERT INTO ShippingPackage VALUES (2014, 1014, 13.19882, '13x13x18', '2025-03-21');

INSERT INTO ShippingPackage VALUES (2015, 1015, 2.26953, '10x13x17', '2025-03-23');

INSERT INTO ShippingPackage VALUES (2016, 1016, 14.26424, '28x22x29', '2025-03-27');

INSERT INTO ShippingPackage VALUES (2017, 1017, 17.60792, '23x18x10', '2025-03-23');

INSERT INTO ShippingPackage VALUES (2018, 1018, 13.67461, '30x30x24', '2025-03-21');

INSERT INTO ShippingPackage VALUES (2019, 1019, 29.31636, '13x13x19', '2025-03-20');

INSERT INTO ShippingPackage VALUES (2020, 1020, 17.9572, '24x13x20', '2025-03-28');

```

Query:

Now we need to display the information which was inserted into the table.

```
SELECT * FROM ShippingPackage
```

PackageID	CustomerID	PackageWeight	PackageDimensions	ShippingDate
2001	1001	48.69582	29x15x24	2025-03-27
2002	1002	45.10995	22x13x10	2025-03-30
2003	1003	41.15647	22x23x29	2025-03-23
2004	1004	11.13834	30x13x15	2025-03-28
2005	1005	48.16558	21x14x22	2025-03-27
2006	1006	42.95698	20x21x11	2025-03-21
2007	1007	38.23529	24x20x10	2025-03-29
2008	1008	33.19635	21x23x23	2025-03-29
2009	1009	34.19965	24x18x16	2025-03-27
2010	1010	25.85617	26x22x23	2025-03-26
2011	1011	14.13477	17x29x18	2025-03-30
2012	1012	8.35196	30x12x8	2025-03-21
2013	1013	31.35637	25x14x29	2025-03-29
2014	1014	13.19882	13x13x18	2025-03-21

4. ShippingPayment

- Purpose:** Stores payment details including fee breakdowns.
- Fields:** PaymentID, PackageID, BaseFee, WeightSurcharge, DistanceSurcharge, TotalShippingFee.
- Entries:** 20 payments inserted.

```
CREATE TABLE ShippingPayment (
    `PaymentID` INT NOT NULL,
    `PackageID` INT NOT NULL,
    `BaseFee` FLOAT NOT NULL,
    `weightSurcharge` FLOAT NOT NULL,
    `DistanceSurcharge` FLOAT NOT NULL,
    `TotalShippingFee` FLOAT NOT NULL,
    PRIMARY KEY (`PaymentID`)
);
```

Insert Statements :

```
INSERT INTO ShippingPayment VALUES (4001, 2001, 73.5, 26.7, 45.3, 145.5);
INSERT INTO ShippingPayment VALUES (4002, 2002, 88.1, 38.6, 49.2, 175.9);
INSERT INTO ShippingPayment VALUES (4003, 2003, 66.3, 19.5, 34.8, 120.6);
INSERT INTO ShippingPayment VALUES (4004, 2004, 101.2, 31.9, 48.7, 181.8);
INSERT INTO ShippingPayment VALUES (4005, 2005, 52.6, 20.3, 33.9, 106.8);
INSERT INTO ShippingPayment VALUES (4006, 2006, 95.4, 42.2, 55.6, 193.2);
INSERT INTO ShippingPayment VALUES (4007, 2007, 85.0, 37.3, 41.6, 163.9);
INSERT INTO ShippingPayment VALUES (4008, 2008, 69.9, 23.8, 36.0, 129.7);
INSERT INTO ShippingPayment VALUES (4009, 2009, 91.7, 39.1, 44.4, 175.2);
INSERT INTO ShippingPayment VALUES (4010, 2010, 77.6, 25.4, 29.3, 132.3);
INSERT INTO ShippingPayment VALUES (4011, 2011, 62.8, 18.6, 27.5, 108.9);
INSERT INTO ShippingPayment VALUES (4012, 2012, 48.9, 14.5, 21.8, 85.2);
INSERT INTO ShippingPayment VALUES (4013, 2013, 132.4, 41.7, 19.5, 193.6);
INSERT INTO ShippingPayment VALUES (4014, 2014, 102.1, 36.4, 39.8, 178.3);
INSERT INTO ShippingPayment VALUES (4015, 2015, 56.7, 19.8, 32.1, 108.6);
INSERT INTO ShippingPayment VALUES (4016, 2016, 118.6, 48.3, 26.0, 192.9);
INSERT INTO ShippingPayment VALUES (4017, 2017, 71.2, 21.1, 28.4, 120.7);
```

```
INSERT INTO ShippingPayment VALUES (4018, 2018, 84.3, 39.2, 51.0, 174.5);
```

```
INSERT INTO ShippingPayment VALUES (4019, 2019, 67.9, 24.5, 27.6, 120.0);
```

```
INSERT INTO ShippingPayment VALUES (4020, 2020, 90.8, 34.1, 39.0, 163.9);
```

Query:

Now we need to display the information which was inserted into the table.

```
SELECT * FROM ShippingPayment
```

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

- Project:** Local instance MySQL80
- Schema:** info_5707_project
- Table:** ShippingPayment
- Query Results:** The results of the SELECT query are displayed in a grid:

PaymentID	PackageID	BaseFee	weightSurcharge	DistanceSurcharge	TotalShippingFee
4001	2001	73.5	26.7	45.3	145.5
4002	2002	88.1	38.6	49.2	175.9
4003	2003	66.3	19.5	34.8	120.6
4004	2004	101.2	31.9	48.7	181.8
4005	2005	52.6	20.3	33.9	106.8
4006	2006	95.4	42.2	55.6	193.2
4007	2007	85	37.3	41.6	163.9
4008	2008	69.9	23.8	36	129.7
4009	2009	91.7	39.1	44.4	175.2
4010	2010	77.6	25.4	29.3	132.3
4011	2011	62.8	18.6	27.5	108.9
4012	2012	48.9	14.5	21.8	85.2
4013	2013	132.4	41.7	19.5	193.6
4014	2014	102.1	36.4	39.8	178.3

- Output:** The Action Output pane shows the log of actions taken:

Action	Time	Message	Duration / Fetch
121	22:53:30	INSERT INTO ShippingPayment VALUES (4017, 2017, 71.2, 21.1, 28.4, 120.7)	1 row(s) affected 0.000 sec
122	22:53:30	INSERT INTO ShippingPayment VALUES (4018, 2018, 84.3, 39.2, 51.0, 174.5)	1 row(s) affected 0.000 sec
123	22:53:30	INSERT INTO ShippingPayment VALUES (4019, 2019, 67.9, 24.5, 27.6, 120.0)	1 row(s) affected 0.016 sec
124	22:53:30	INSERT INTO ShippingPayment VALUES (4020, 2020, 90.8, 34.1, 39.0, 163.9)	1 row(s) affected 0.000 sec
125	22:53:52	SELECT * FROM ShippingPayment LIMIT 0, 1000	20 row(s) returned 0.000 sec / 0.000 sec

5. DeliveryDriver

- Purpose:** Contains details of delivery drivers.
- Fields:** DriverID, DriverName, PhoneNumber, LicenseNumber.
- Entries:** 20 drivers inserted.

```
CREATE TABLE DeliveryDriver (
```

```
    `DriverID` VARCHAR(40) NOT NULL,
```

```
    `DriverName` VARCHAR(40) NOT NULL,
```

```
    `PhoneNumber` VARCHAR(15) NOT NULL,
```

```
    `LicenseNumber` VARCHAR(10) NOT NULL,
```

PRIMARY KEY (`DriverID`)

);

Insert Statements :

```
INSERT INTO DeliveryDriver VALUES ('DRV001', 'Jason Reed', '999-000-1001', 'LIC0000001');
INSERT INTO DeliveryDriver VALUES ('DRV002', 'Natalie Brooks', '999-000-1002', 'LIC0000002');
INSERT INTO DeliveryDriver VALUES ('DRV003', 'Chris Parker', '999-000-1003', 'LIC0000003');
INSERT INTO DeliveryDriver VALUES ('DRV004', 'Samantha Hayes', '999-000-1004', 'LIC0000004');
INSERT INTO DeliveryDriver VALUES ('DRV005', 'Brandon Murphy', '999-000-1005', 'LIC0000005');
INSERT INTO DeliveryDriver VALUES ('DRV006', 'Rachel Foster', '999-000-1006', 'LIC0000006');
INSERT INTO DeliveryDriver VALUES ('DRV007', 'Anthony Rivera', '999-000-1007', 'LIC0000007');
INSERT INTO DeliveryDriver VALUES ('DRV008', 'Julia Simmons', '999-000-1008', 'LIC0000008');
INSERT INTO DeliveryDriver VALUES ('DRV009', 'Kevin Price', '999-000-1009', 'LIC0000009');
INSERT INTO DeliveryDriver VALUES ('DRV010', 'Laura Bell', '999-000-1010', 'LIC0000010');
INSERT INTO DeliveryDriver VALUES ('DRV011', 'Dylan Carter', '999-000-1011', 'LIC0000011');
INSERT INTO DeliveryDriver VALUES ('DRV012', 'Ashley Barnes', '999-000-1012', 'LIC0000012');
INSERT INTO DeliveryDriver VALUES ('DRV013', 'Zachary Ross', '999-000-1013', 'LIC0000013');
INSERT INTO DeliveryDriver VALUES ('DRV014', 'Victoria Powell', '999-000-1014', 'LIC0000014');
INSERT INTO DeliveryDriver VALUES ('DRV015', 'Tyler Morgan', '999-000-1015', 'LIC0000015');
INSERT INTO DeliveryDriver VALUES ('DRV016', 'Brianna Hughes', '999-000-1016', 'LIC0000016');
INSERT INTO DeliveryDriver VALUES ('DRV017', 'Eric Long', '999-000-1017', 'LIC0000017');
INSERT INTO DeliveryDriver VALUES ('DRV018', 'Nicole Peterson', '999-000-1018', 'LIC0000018');
INSERT INTO DeliveryDriver VALUES ('DRV019', 'Kyle Ward', '999-000-1019', 'LIC0000019');
INSERT INTO DeliveryDriver VALUES ('DRV020', 'Chloe Sanders', '999-000-1020', 'LIC0000020');
```

Query:

Now we need to display the information which was inserted into the table.

SELECT * FROM DeliveryDriver

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

- Navigator:** Shows the schema `info_5707_project` containing tables, views, stored procedures, and functions.
- SQL Editor:** Displays the following SQL code:

```
18 • INSERT INTO DeliveryDriver VALUES ('DRV018', 'Nicole Peterson', '999-000-1018', 'LIC00000018');
19 • INSERT INTO DeliveryDriver VALUES ('DRV019', 'Kyle Ward', '999-000-1019', 'LIC00000019');
20 • INSERT INTO DeliveryDriver VALUES ('DRV020', 'Chloe Sanders', '999-000-1020', 'LIC00000020');
21
22
23 • SELECT * FROM DeliveryDriver
```
- Result Grid:** Shows the data from the `DeliveryDriver` table:

DriverID	DriverName	PhoneNumber	LiceneNumber
DRV001	Jason Reed	999-000-1001	LIC00000001
DRV002	Natalie Brooks	999-000-1002	LIC00000002
DRV003	Chris Parker	999-000-1003	LIC00000003
DRV004	Samantha Hayes	999-000-1004	LIC00000004
DRV005	Brandon Murphy	999-000-1005	LIC00000005
DRV006	Rachel Foster	999-000-1006	LIC00000006
DRV007	Anthony Rivera	999-000-1007	LIC00000007
DRV008	Julia Simmons	999-000-1008	LIC00000008
DRV009	Kevin Price	999-000-1009	LIC00000009
DRV010	Laura Bell	999-000-1010	LIC00000010
DRV011	Dylan Carter	999-000-1011	LIC00000011
DRV012	Ashley Barnes	999-000-1012	LIC00000012
DRV013	Zachary Ross	999-000-1013	LIC00000013
DRV014	Victoria Powell	999-000-1014	LIC00000014
- Output:** Shows the execution history with the following details:

#	Time	Action	Message	Duration / Fetch
142	22:56:26	INSERT INTO DeliveryDriver VALUES (DRV017, 'Eric Long', '999-000-1017', 'LIC00000017)	1 row(s) affected	0.032 sec
143	22:56:26	INSERT INTO DeliveryDriver VALUES (DRV018, 'Nicole Peterson', '999-000-1018', 'LIC00000018)	1 row(s) affected	0.000 sec
144	22:56:26	INSERT INTO DeliveryDriver VALUES (DRV019, 'Kyle Ward', '999-000-1019', 'LIC00000019)	1 row(s) affected	0.000 sec
145	22:56:26	INSERT INTO DeliveryDriver VALUES (DRV020, 'Chloe Sanders', '999-000-1020', 'LIC00000020)	1 row(s) affected	0.000 sec
146	22:56:48	SELECT * FROM DeliveryDriver LIMIT 0, 1000	20 row(s) returned	0.000 sec / 0.000 sec

6. DeliveryRoute

- Purpose:** Assigns packages to routes and drivers.
- Fields:** RouteID, DriverID, PackageID, RouteName.
- Entries:** 20 delivery routes inserted.

CREATE TABLE DeliveryRoute (

`RouteID` INT NOT NULL,

`DriverID` VARCHAR(40) NOT NULL,

`PackageID` INT NOT NULL,

`RouteName` VARCHAR(250) NOT NULL,

PRIMARY KEY (`RouteID`)

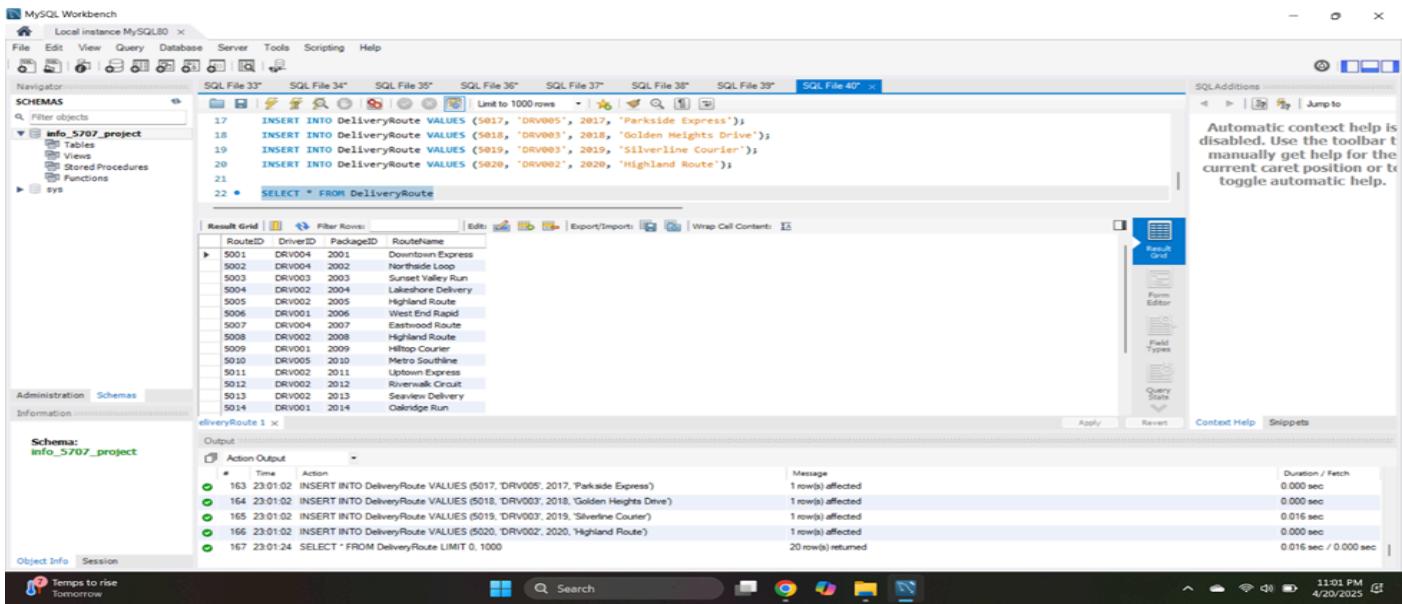
);

Insert Statements :

```
INSERT INTO DeliveryRoute VALUES (5001, 'DRV004', 2001, 'Downtown Express');  
INSERT INTO DeliveryRoute VALUES (5002, 'DRV004', 2002, 'Northside Loop');  
INSERT INTO DeliveryRoute VALUES (5003, 'DRV003', 2003, 'Sunset Valley Run');  
INSERT INTO DeliveryRoute VALUES (5004, 'DRV002', 2004, 'Lakeshore Delivery');  
INSERT INTO DeliveryRoute VALUES (5005, 'DRV002', 2005, 'Highland Route');  
INSERT INTO DeliveryRoute VALUES (5006, 'DRV001', 2006, 'West End Rapid');  
INSERT INTO DeliveryRoute VALUES (5007, 'DRV004', 2007, 'Eastwood Route');  
INSERT INTO DeliveryRoute VALUES (5008, 'DRV002', 2008, 'Highland Route');  
INSERT INTO DeliveryRoute VALUES (5009, 'DRV001', 2009, 'Hilltop Courier');  
INSERT INTO DeliveryRoute VALUES (5010, 'DRV005', 2010, 'Metro Southline');  
INSERT INTO DeliveryRoute VALUES (5011, 'DRV002', 2011, 'Uptown Express');  
INSERT INTO DeliveryRoute VALUES (5012, 'DRV002', 2012, 'Riverwalk Circuit');  
INSERT INTO DeliveryRoute VALUES (5013, 'DRV002', 2013, 'Seaview Delivery');  
INSERT INTO DeliveryRoute VALUES (5014, 'DRV001', 2014, 'Oakridge Run');  
INSERT INTO DeliveryRoute VALUES (5015, 'DRV003', 2015, 'Central Loop');  
INSERT INTO DeliveryRoute VALUES (5016, 'DRV004', 2016, 'Old Town Trail');  
INSERT INTO DeliveryRoute VALUES (5017, 'DRV005', 2017, 'Parkside Express');  
INSERT INTO DeliveryRoute VALUES (5018, 'DRV003', 2018, 'Golden Heights Drive');  
INSERT INTO DeliveryRoute VALUES (5019, 'DRV003', 2019, 'Silverline Courier');  
INSERT INTO DeliveryRoute VALUES (5020, 'DRV002', 2020, 'Highland Route');
```

Query:

Now we need to display the information which was inserted into the table.
SELECT * FROM DeliveryRoute



7. PackageTrackingInformation

- Purpose:** Tracks delivery status of each package.
- Fields:** TrackingNumber, PackageID, DeliveryStatus, LastUpdated.
- Entries:** 20 tracking records inserted

CREATE TABLE PackageTrackingInformation (

 `TrackingNumber` VARCHAR(30) NOT NULL,

 `PackageID` INT NOT NULL,

 `DeliveryStatus` ENUM('Pending', 'In Transit', 'Out for Delivery', 'Delivered', 'Failed') NOT NULL,

 `LastUpdated` DATE NOT NULL,

 PRIMARY KEY (`TrackingNumber`)

);

Insert Statements :

```
INSERT INTO PackageTrackingInformation VALUES ('TRK0001', 2001, 'Failed', '2025-03-24');
```

```
INSERT INTO PackageTrackingInformation VALUES ('TRK0002', 2002, 'Out for Delivery', '2025-03-24');
```

```
INSERT INTO PackageTrackingInformation VALUES ('TRK0003', 2003, 'Failed', '2025-03-24');
```

```
INSERT INTO PackageTrackingInformation VALUES ('TRK0004', 2004, 'Out for Delivery', '2025-03-24');
```

```
INSERT INTO PackageTrackingInformation VALUES ('TRK0005', 2005, 'In Transit', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0006', 2006, 'Failed', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0007', 2007, 'In Transit', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0008', 2008, 'Out for Delivery', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0009', 2009, 'Pending', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0010', 2010, 'Pending', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0011', 2011, 'Pending', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0012', 2012, 'Failed', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0013', 2013, 'Out for Delivery', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0014', 2014, 'Pending', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0015', 2015, 'In Transit', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0016', 2016, 'Delivered', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0017', 2017, 'In Transit', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0018', 2018, 'Out for Delivery', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0019', 2019, 'Pending', '2025-03-24');

INSERT INTO PackageTrackingInformation VALUES ('TRK0020', 2020, 'Failed', '2025-03-24');
```

Query:

Now we need to display the information which was inserted into the table.

```
SELECT * FROM PackageTrackingInformation
```

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, info_5707_project (selected).
- SQL Editor:** SQL File 41* (active), containing the following SQL code:


```

17 • INSERT INTO PackageTrackingInformation VALUES ('TRK0017', 2017, 'In Transit', '2025-03-24');
18 • INSERT INTO PackageTrackingInformation VALUES ('TRK0018', 2018, 'Out for Delivery', '2025-03-24');
19 • INSERT INTO PackageTrackingInformation VALUES ('TRK0019', 2019, 'Pending', '2025-03-24');
20 • INSERT INTO PackageTrackingInformation VALUES ('TRK0020', 2020, 'Failed', '2025-03-24');
21
22 • SELECT * FROM PackageTrackingInformation
      
```
- Result Grid:** Shows a table with columns: TrackingNumber, PackageID, DeliveryStatus, LastUpdated. The data consists of 15 rows from TRK0001 to TRK0015, with the last five rows being inserts from the previous SQL.
- Output:** Action Output pane showing the execution log with 188 entries, all successful, with execution times ranging from 0.000 sec to 0.015 sec.
- System Bar:** Shows the schema (info_5707_project), session information (15°C Clear), and system status (11:04 PM, 4/20/2025).

8. PackageDeliveryStatus

- Purpose:** Records delivery status updates with timestamp and location.
- Fields:** StatusID, TrackingNumber, StatusDescription, StatusTimeStamp, CurrentLocation.
- Entries:** 20 status records inserted.

```

CREATE TABLE PackageDeliveryStatus (
    `StatusID` INT NOT NULL,
    `TrackingNumber` VARCHAR(40) NOT NULL,
    `StatusDescription` VARCHAR(250) NOT NULL,
    `StatusTimeStamp` DATETIME NOT NULL,
    `CurrentLocation` VARCHAR(100) NOT NULL,
    PRIMARY KEY (`StatusID`)
);
      
```

Insert Statements :

INSERT INTO PackageDeliveryStatus VALUES (6001, 'TRK0001', 'Processing', '2025-03-24 12:00:00', 'Dallas Distribution Center');

INSERT INTO PackageDeliveryStatus VALUES (6002, 'TRK0002', 'In Transit', '2025-03-24 12:00:00', 'Austin Hub');

INSERT INTO PackageDeliveryStatus VALUES (6003, 'TRK0003', 'Out for Delivery', '2025-03-24 12:00:00', 'San Antonio Depot');

INSERT INTO PackageDeliveryStatus VALUES (6004, 'TRK0004', 'Delivered', '2025-03-24 12:00:00', 'Houston Sorting Facility');

INSERT INTO PackageDeliveryStatus VALUES (6005, 'TRK0005', 'Attempted Delivery', '2025-03-24 12:00:00', 'El Paso Terminal');

INSERT INTO PackageDeliveryStatus VALUES (6006, 'TRK0006', 'Failed Delivery', '2025-03-24 12:00:00', 'Fort Worth Transit');

INSERT INTO PackageDeliveryStatus VALUES (6007, 'TRK0007', 'Delayed', '2025-03-24 12:00:00', 'Plano Dispatch Center');

INSERT INTO PackageDeliveryStatus VALUES (6008, 'TRK0008', 'Returned to Sender', '2025-03-24 12:00:00', 'Garland Transfer Station');

INSERT INTO PackageDeliveryStatus VALUES (6009, 'TRK0009', 'Processing', '2025-03-24 12:00:00', 'Irving Crossdock');

INSERT INTO PackageDeliveryStatus VALUES (6010, 'TRK0010', 'In Transit', '2025-03-24 12:00:00', 'Frisco Sorting Point');

INSERT INTO PackageDeliveryStatus VALUES (6011, 'TRK0011', 'Out for Delivery', '2025-03-24 12:00:00', 'McKinney Warehouse');

INSERT INTO PackageDeliveryStatus VALUES (6012, 'TRK0012', 'Delivered', '2025-03-24 12:00:00', 'Lubbock Drop-off');

INSERT INTO PackageDeliveryStatus VALUES (6013, 'TRK0013', 'Attempted Delivery', '2025-03-24 12:00:00', 'Amarillo Regional');

INSERT INTO PackageDeliveryStatus VALUES (6014, 'TRK0014', 'Failed Delivery', '2025-03-24 12:00:00', 'Waco Substation');

INSERT INTO PackageDeliveryStatus VALUES (6015, 'TRK0015', 'Delayed', '2025-03-24 12:00:00', 'Tyler Freight Hub');

INSERT INTO PackageDeliveryStatus VALUES (6016, 'TRK0016', 'Returned to Sender', '2025-03-24 12:00:00', 'Denton Logistics Base');

INSERT INTO PackageDeliveryStatus VALUES (6017, 'TRK0017', 'Processing', '2025-03-24 12:00:00', 'Richardson Delivery Point');

```
INSERT INTO PackageDeliveryStatus VALUES (6018, 'TRK0018', 'In Transit', '2025-03-24 12:00:00', 'Midland Transit Station');
```

```
INSERT INTO PackageDeliveryStatus VALUES (6019, 'TRK0019', 'Out for Delivery', '2025-03-24 12:00:00', 'Beaumont Express Hub');
```

```
INSERT INTO PackageDeliveryStatus VALUES (6020, 'TRK0020', 'Delivered', '2025-03-24 12:00:00', 'Killeen Truck Stop');
```

Query:

Now we need to display the information which was inserted into the table.

```
SELECT * FROM PackageDeliveryStatus
```

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

- Navigator:** Shows the schema `info_5707_project` containing tables, views, stored procedures, and functions.
- SQL Editor:** Displays the executed SQL statements:

```
17 • INSERT INTO PackageDeliveryStatus VALUES (6017, 'TRK0017', 'Processing', '2025-03-24 12:00:00', 'Richardson Delivery Point');
18 • INSERT INTO PackageDeliveryStatus VALUES (6018, 'TRK0018', 'In Transit', '2025-03-24 12:00:00', 'Midland Transit Station');
19 • INSERT INTO PackageDeliveryStatus VALUES (6019, 'TRK0019', 'Out for Delivery', '2025-03-24 12:00:00', 'Beaumont Express Hub');
20 • INSERT INTO PackageDeliveryStatus VALUES (6020, 'TRK0020', 'Delivered', '2025-03-24 12:00:00', 'Killeen Truck Stop');
21
22 • SELECT * FROM PackageDeliveryStatus
```
- Result Grid:** Shows the results of the `SELECT` query:

StatusID	TrackingNumber	StatusDescription	StatusTimeStamp	CurrentLocation
6001	TRK0001	Processing	2025-03-24 12:00:00	Dallas Distribution Center
6002	TRK0002	In Transit	2025-03-24 12:00:00	Austin Hub
6003	TRK0003	Out for Delivery	2025-03-24 12:00:00	San Antonio Depot
6004	TRK0004	Delivered	2025-03-24 12:00:00	Houston Sorting Facility
6005	TRK0005	Attempted Delivery	2025-03-24 12:00:00	El Paso Terminal
6006	TRK0006	Failed Delivery	2025-03-24 12:00:00	Fort Worth Transit
6007	TRK0007	Delayed	2025-03-24 12:00:00	Plano Dispatch Center
6008	TRK0008	Returned to Sender	2025-03-24 12:00:00	Garland Transfer Station
6009	TRK0009	Processing	2025-03-24 12:00:00	Irving Crossdock
6010	TRK0010	In Transit	2025-03-24 12:00:00	Frisco Sorting Point
6011	TRK0011	Out for Delivery	2025-03-24 12:00:00	McKinney Warehouse
6012	TRK0012	Delivered	2025-03-24 12:00:00	Lubbock Drop-off
6013	TRK0013	Attempted Delivery	2025-03-24 12:00:00	Amarillo Regional
6014	TRK0014	Failed Delivery	2025-03-24 12:00:00	Waco Substation
- Output:** Shows the execution log with 20 rows affected by the previous statements.
- System Bar:** Includes icons for weather (15°C Clear), search, and system status (11:08 PM, 4/20/2025).

9. DeliveryProof

- Purpose:** Stores proof of delivery (signatures or photos).
- Fields:** PODID, TrackingNumber, NameOfCustomer, ProofAtDelivery, DeliveryTimeStamp.
- Entries:** 20 proof records inserted.

```
CREATE TABLE DeliveryProof (
```

```
`PODID` VARCHAR(20) NOT NULL,  
`TrackingNumber` VARCHAR(20) NOT NULL,  
`NameOfCustomer` VARCHAR(40) NOT NULL,  
`ProofAtDelivery` BLOB NOT NULL,  
`DeliveryTimeStamp` DATETIME NOT NULL,  
PRIMARY KEY (`PODID`)  
);
```

Insert Statements :

```
INSERT INTO DeliveryProof VALUES ('POD0001', 'TRK0001', 'Alice Johnson',  
'signature_alice_johnson.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0002', 'TRK0002', 'Michael Smith',  
'photo_michael_smith.jpg', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0003', 'TRK0003', 'Sophia Martinez',  
'signature_sophia_martinez.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0004', 'TRK0004', 'David Lee', 'photo_david_lee.jpg',  
'2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0005', 'TRK0005', 'Emily Nguyen',  
'signature_emily_nguyen.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0006', 'TRK0006', 'Daniel Thompson',  
'signature_daniel_thompson.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0007', 'TRK0007', 'Olivia Brown',  
'photo.olivia_brown.jpg', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0008', 'TRK0008', 'James Wilson',  
'signature_james_wilson.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0009', 'TRK0009', 'Ava Patel', 'photo_ava_patel.jpg',  
'2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0010', 'TRK0010', 'William Clark',  
'signature_william_clark.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0011', 'TRK0011', 'Isabella Hernandez',  
'signature_isabella_hernandez.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0012', 'TRK0012', 'Benjamin White',  
'photo_benjamin_white.jpg', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0013', 'TRK0013', 'Mia Robinson',  
'signature_mia_robinson.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0014', 'TRK0014', 'Lucas King',  
'photo_lucas_king.jpg', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0015', 'TRK0015', 'Charlotte Wright',  
'signature_charlotte_wright.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0016', 'TRK0016', 'Henry Baker',  
'signature_henry_baker.png', '2025-03-25 13:00:00');
```

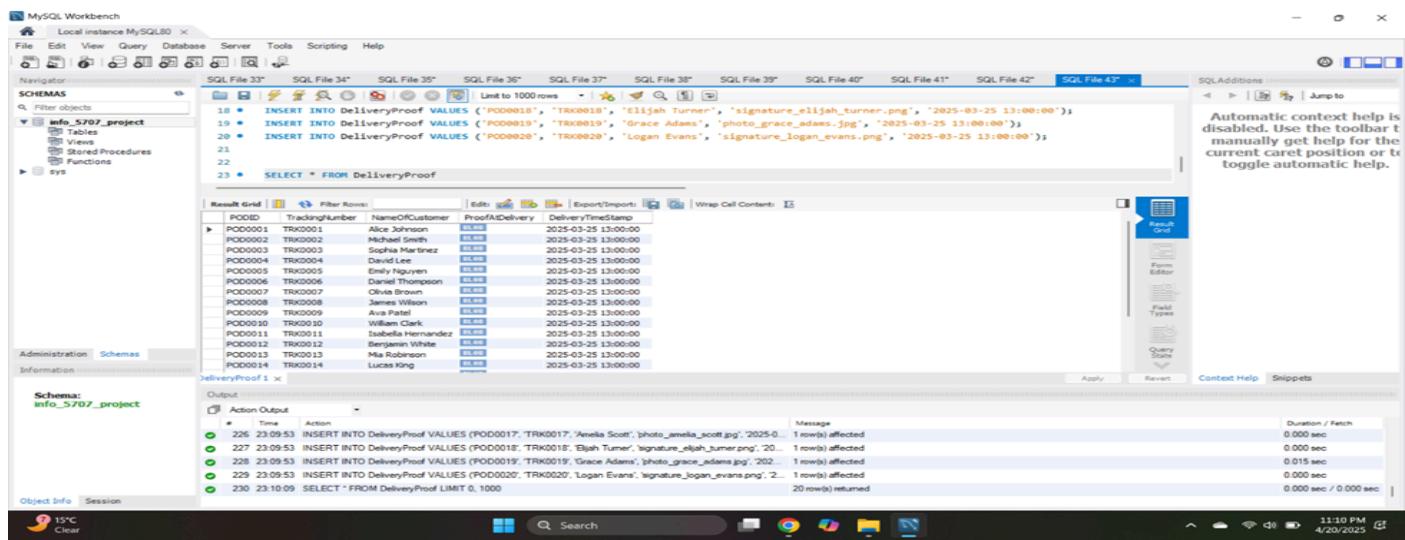
```
INSERT INTO DeliveryProof VALUES ('POD0017', 'TRK0017', 'Amelia Scott',  
'photo_amelia_scott.jpg', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0018', 'TRK0018', 'Elijah Turner',  
'signature_elijah_turner.png', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0019', 'TRK0019', 'Grace Adams',  
'photo_grace_adams.jpg', '2025-03-25 13:00:00');
```

```
INSERT INTO DeliveryProof VALUES ('POD0020', 'TRK0020', 'Logan Evans',  
'signature_logan_evans.png', '2025-03-25 13:00:00');
```

Query: Now we need to display the information which was inserted into the table.
SELECT * FROM DeliveryProof



The screenshot shows the MySQL Workbench interface with the 'DeliveryProof' table selected. The table has columns: POID, TrackingNumber, NameOfCustomer, ProofOfDelivery, and DeliveryTimeStamp. The data consists of 20 rows, each representing one of the customers listed in the previous INSERT statements. The 'Actions' pane at the bottom shows the SQL queries used to insert the data.

POID	TrackingNumber	NameOfCustomer	ProofOfDelivery	DeliveryTimeStamp
POD0001	TRK0001	Alice Johnson	image	2025-03-25 13:00:00
POD0002	TRK0002	Michael Smith	image	2025-03-25 13:00:00
POD0003	TRK0003	Sophia White	image	2025-03-25 13:00:00
POD0004	TRK0004	David Lee	image	2025-03-25 13:00:00
POD0005	TRK0005	Emily Nguyen	image	2025-03-25 13:00:00
POD0006	TRK0006	Daniel Thompson	image	2025-03-25 13:00:00
POD0007	TRK0007	Olivia Kim	image	2025-03-25 13:00:00
POD0008	TRK0008	James Wilson	image	2025-03-25 13:00:00
POD0009	TRK0009	Ava Patel	image	2025-03-25 13:00:00
POD0010	TRK0010	William Parker	image	2025-03-25 13:00:00
POD0011	TRK0011	Isabella Hernandez	image	2025-03-25 13:00:00
POD0012	TRK0012	Benjamin White	image	2025-03-25 13:00:00
POD0013	TRK0013	Mia Robinson	image	2025-03-25 13:00:00
POD0014	TRK0014	Lucas King	image	2025-03-25 13:00:00

10. DeliveryAttempt

- Purpose:** Logs each delivery attempt and outcome.
- Fields:** AttemptID, TrackingNumber, AttemptTime, Status, Notes.
- Entries:** 20 attempts inserted.

```
CREATE TABLE DeliveryAttempt (
```

```
    AttemptID INT NOT NULL,
```

```

TrackingNumber VARCHAR(30) NOT NULL,
AttemptTime DATETIME NOT NULL,
Status ENUM('Successful', 'Failed') NOT NULL,
Notes TEXT,
PRIMARY KEY (AttemptID),
FOREIGN KEY (TrackingNumber) REFERENCES
PackageTrackingInformation(TrackingNumber)
);

```

Insert Statements :

```

INSERT INTO DeliveryAttempt VALUES
(7001, 'TRK0001', '2025-03-24 09:00:00', 'Failed', 'Customer not available at address'),
(7002, 'TRK0001', '2025-03-25 10:30:00', 'Successful', 'Delivered to neighbor'),
(7003, 'TRK0003', '2025-03-24 08:45:00', 'Failed', 'Address not found'),
(7004, 'TRK0003', '2025-03-26 11:15:00', 'Failed', 'Incorrect address again'),
(7005, 'TRK0005', '2025-03-24 13:10:00', 'Failed', 'No safe place to leave package'),
(7006, 'TRK0012', '2025-03-24 09:50:00', 'Failed', 'Security refused entry'),
(7007, 'TRK0012', '2025-03-25 14:00:00', 'Successful', 'Delivered to front desk'),
(7008, 'TRK0020', '2025-03-24 12:20:00', 'Failed', 'Weather conditions delayed delivery'),
(7009, 'TRK0020', '2025-03-25 16:10:00', 'Successful', 'Left at reception'),
(7010, 'TRK0004', '2025-03-23 10:00:00', 'Successful', 'Delivered to customer directly'),
(7011, 'TRK0006', '2025-03-24 11:20:00', 'Failed', 'Package damaged during transit'),
(7012, 'TRK0015', '2025-03-25 13:15:00', 'Failed', 'Gate locked, access denied'),
(7013, 'TRK0015', '2025-03-26 10:10:00', 'Successful', 'Left with building manager'),
(7014, 'TRK0007', '2025-03-24 10:45:00', 'Failed', 'Delay at sorting center'),
(7015, 'TRK0010', '2025-03-24 08:30:00', 'Successful', 'Package delivered before noon'),
(7016, 'TRK0017', '2025-03-24 13:45:00', 'Failed', 'Customer requested reschedule'),
(7017, 'TRK0017', '2025-03-25 15:00:00', 'Successful', 'Delivered on second attempt'),
(7018, 'TRK0019', '2025-03-24 14:20:00', 'Failed', 'Receiver not available'),

```

(7019, 'TRK0019', '2025-03-26 09:40:00', 'Successful', 'Delivered at back door'),

(7020, 'TRK0002', '2025-03-24 10:30:00', 'Successful', 'Left in secure package box');

Query:

Now we need to display the information which was inserted into the table.

SELECT * FROM DeliveryAttempt

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

- Navigator:** Shows the schema `info_5707_project` with objects: Tables, Views, Stored Procedures, Functions, and sys.
- SQL Editor:** Contains the SQL query: `SELECT * FROM DeliveryAttempt;`. Below it, the results are displayed in a grid.
- Result Grid:** Displays the data from the DeliveryAttempt table. The columns are `AttemptID`, `TrackingNumber`, `AttemptTime`, `Status`, and `Notes`. The data includes rows for attempts 7001 through 7010, along with two additional rows (7015 and 7016) shown in the SQL editor.
- Action Output:** Shows the log of actions taken during the session, including insertions into the `DeliveryProof` and `DeliveryAttempt` tables, and SELECT statements.
- System Bar:** Includes icons for weather (15°C Clear), search, and system status (11:19 PM, 4/20/2025).

11. CustomerFeedback

- Purpose:** Stores ratings and comments from customers after delivery.
- Fields:** FeedbackID, CustomerID, TrackingNumber, Rating, Comments, SubmittedAt.
- Entries:** 20 feedback records inserted.

CREATE TABLE CustomerFeedback (

FeedbackID INT NOT NULL,

```

CustomerID INT NOT NULL,
TrackingNumber VARCHAR(30) NOT NULL,
Rating INT CHECK (Rating BETWEEN 1 AND 5),
Comments TEXT,
SubmittedAt DATETIME NOT NULL,
PRIMARY KEY (FeedbackID),
FOREIGN KEY (CustomerID) REFERENCES
DeliveryCustomers(CustomerID),
FOREIGN KEY (TrackingNumber) REFERENCES
PackageTrackingInformation(TrackingNumber)
);

```

Insert Statements :

```

INSERT INTO CustomerFeedback VALUES
(8001, 1001, 'TRK0001', 4, 'Package arrived in good condition.', '2025-03-26 14:00:00'),
(8002, 1002, 'TRK0002', 5, 'Very fast delivery. Great service!', '2025-03-26 15:30:00'),
(8003, 1003, 'TRK0003', 2, 'Faced multiple failed delivery attempts.', '2025-03-27 10:45:00'),
(8004, 1004, 'TRK0004', 4, 'Driver was polite and prompt.', '2025-03-26 16:10:00'),
(8005, 1005, 'TRK0005', 3, 'Took longer than expected.', '2025-03-27 12:20:00'),
(8006, 1006, 'TRK0006', 1, 'Package was damaged and delivery failed.', '2025-03-27 09:00:00'),
(8007, 1007, 'TRK0007', 3, 'Still in transit, updates needed.', '2025-03-26 18:30:00'),
(8008, 1008, 'TRK0008', 4, 'Received with slight delay.', '2025-03-27 14:40:00'),
(8009, 1009, 'TRK0009', 2, 'Pending too long, not happy.', '2025-03-28 08:20:00'),
(8010, 1010, 'TRK0010', 4, 'Smooth and timely delivery.', '2025-03-26 13:05:00'),
(8011, 1011, 'TRK0011', 5, 'Very professional and fast.', '2025-03-27 11:00:00'),
(8012, 1012, 'TRK0012', 2, 'Faced building access issues.', '2025-03-28 15:15:00'),
(8013, 1013, 'TRK0013', 4, 'Minor issues but got the package.', '2025-03-28 10:30:00'),
(8014, 1014, 'TRK0014', 3, 'Failed delivery, needed reschedule.', '2025-03-27 17:25:00'),
(8015, 1015, 'TRK0015', 4, 'Delivery attempt successful after retry.', '2025-03-28 12:00:00'),

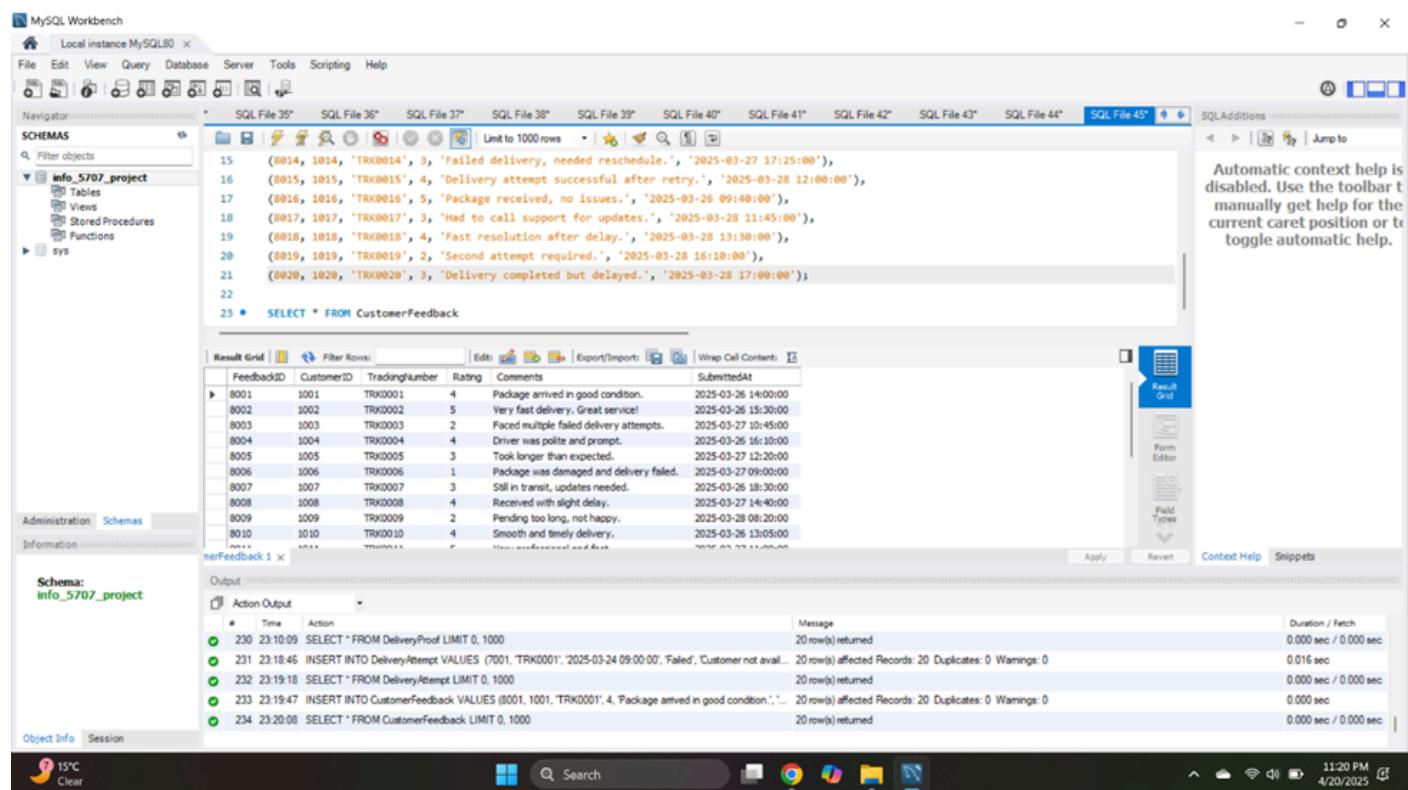
```

(8016, 1016, 'TRK0016', 5, 'Package received, no issues.', '2025-03-26 09:40:00'),
 (8017, 1017, 'TRK0017', 3, 'Had to call support for updates.', '2025-03-28 11:45:00'),
 (8018, 1018, 'TRK0018', 4, 'Fast resolution after delay.', '2025-03-28 13:30:00'),
 (8019, 1019, 'TRK0019', 2, 'Second attempt required.', '2025-03-28 16:10:00'),
 (8020, 1020, 'TRK0020', 3, 'Delivery completed but delayed.', '2025-03-28 17:00:00');

Query:

Now we need to display the information which was inserted into the table.

SELECT * FROM CustomerFeedback



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

- MySQL Workbench - Local instance MySQL80**: The title bar.
- File Edit View Query Database Server Tools Scripting Help**: The menu bar.
- Navigator**: The left sidebar showing the schema tree for the **info_5707_project** database, including Schemas, Tables, Views, Stored Procedures, and Functions.
- SQL Editor**: The main workspace displaying the query:


```
15  (8014, 1014, 'TRK0014', 3, 'Failed delivery, needed reschedule.', '2025-03-27 17:25:00'),
16  (8015, 1015, 'TRK0015', 4, 'Delivery attempt successful after retry.', '2025-03-28 12:00:00'),
17  (8016, 1016, 'TRK0016', 5, 'Package received, no issues.', '2025-03-26 09:40:00'),
18  (8017, 1017, 'TRK0017', 3, 'Had to call support for updates.', '2025-03-28 11:45:00'),
19  (8018, 1018, 'TRK0018', 4, 'Fast resolution after delay.', '2025-03-28 13:30:00'),
20  (8019, 1019, 'TRK0019', 2, 'Second attempt required.', '2025-03-28 16:10:00'),
21  (8020, 1020, 'TRK0020', 3, 'Delivery completed but delayed.', '2025-03-28 17:00:00');

23 • SELECT * FROM CustomerFeedback
```
- Result Grid**: The results of the query, showing 21 rows of data from the CustomerFeedback table.
- Action Output**: A log of SQL actions and their execution details, including time, action, message, and duration.
- Object Info** and **Session**: The bottom-left pane showing schema and session information.
- System tray**: The bottom-right pane showing system status (15°C, Clear, 11:20 PM, 4/20/2025).

Query 1 : Failed Deliveries

SELECT c.FullName, t.TrackingNumber, t.DeliveryStatus

FROM DeliveryCustomers c

JOIN ShippingPackage p ON c.CustomerID = p.CustomerID

JOIN PackageTrackingInformation t ON p.PackageID = t.PackageID

WHERE t.DeliveryStatus = 'Failed';

Result:

```

SELECT c.FullName, t.TrackingNumber, t.DeliveryStatus
FROM DeliveryCustomers c
JOIN ShippingPackage p ON c.CustomerID = p.CustomerID
JOIN PackageTrackingInformation t ON p.PackageID = t.PackageID
WHERE t.DeliveryStatus = 'Failed'

```

FUName	TrackingNumber	DeliveryStatus
Alice Johnson	TRK0001	Failed
Sophia Martinez	TRK0003	Failed
Daniel Thompson	TRK0006	Failed
Benjamin White	TRK0012	Failed
Logan Evans	TRK0020	Failed

Result 1 ×

Action Output

Time	Action	Message	Duration / Fetch
232	23:19:18 SELECT * FROM DeliveryAttempt LIMIT 0, 1000	20 row(s) returned	0.000 sec / 0.000 sec
233	23:19:47 INSERT INTO CustomerFeedback VALUES (8001, 1001, TRK0001, 4, 'Package arrived in good condition.', '...')	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0	0.000 sec
234	23:20:08 SELECT * FROM CustomerFeedback LIMIT 0, 1000	20 row(s) returned	0.000 sec / 0.000 sec
235	23:22:27 SELECT c.FullName, t.TrackingNumber, t.DeliveryStatus FROM DeliveryCustomers c JOIN ShippingPackage ...	Error Code: 1054. Unknown column 't.PackageDeliveryStatus' in 'where clause'	0.000 sec
236	23:24:03 SELECT c.FullName, t.TrackingNumber, t.DeliveryStatus FROM DeliveryCustomers c JOIN ShippingPackage ...	5 row(s) returned	0.000 sec / 0.000 sec

Query 2 : Total Shipping fees By Customer

```

SELECT c.FullName, SUM(d.TotalShippingFee) AS TotalShippingPaid
FROM DeliveryCustomers c
JOIN ShippingPackage p ON c.CustomerID = p.CustomerID
JOIN ShippingPayment d ON p.PackageID = d.PackageID
GROUP BY c.FullName;

```

Result:

```

SELECT c.FullName, SUM(d.TotalShippingFee) AS TotalShippingPaid
FROM DeliveryCustomers c
JOIN ShippingPackage p ON c.CustomerID = p.CustomerID
JOIN ShippingPayment d ON p.PackageID = d.PackageID
GROUP BY c.FullName;

```

FUName	TotalShippingPaid
Alice Johnson	145.5
Michael Smith	175.8999994824219
Sophia Martinez	121.9999994824211
David Lee	181.8000003517579
Emily Nguyen	106.80000035175791
Daniel Thompson	193.199999482422
Olivia Parker	110.8000003517579
Samuel Wilson	129.699999482422
Ava Patel	175.199999482422
William Clark	132.3000003517579
Liam Parker	120.8000003517579
Benjamin White	85.1999994824219
Mia Robinson	193.40000010351562
Lucas King	176.3000003517579
Charlotte Wright	108.5999994824211
Henry Baker	192.89999939548438
Anneke Scott	120.6999994824218
Ryder Turner	174.5
Grace Adams	120

Result 1 ×

Action Output

Time	Action	Message	Duration / Fetch
237	23:24:44 SELECT c.FullName, SUM(d.TotalShippingFee) AS TotalShippingPaid	20 row(s) returned	0.000 sec / 0.000 sec
238	23:24:54 FROM DeliveryCustomers c	20 row(s) returned	0.000 sec
239	23:24:55 JOIN ShippingPackage p ON c.CustomerID = p.CustomerID	20 row(s) returned	0.000 sec
240	23:24:56 JOIN ShippingPayment d ON p.PackageID = d.PackageID	20 row(s) returned	0.000 sec
241	23:24:57 GROUP BY c.FullName;	20 row(s) returned	0.000 sec / 0.000 sec

Query 3 : Out for Delivery Packages with Drivers and Routes

```
SELECT c.FullName, t.TrackingNumber, dr.DriverName, r.RouteName  
FROM PackageTrackingInformation t  
JOIN ShippingPackage p ON t.PackageID = p.PackageID  
JOIN DeliveryCustomers c ON p.CustomerID = c.CustomerID  
JOIN DeliveryRoute r ON p.PackageID = r.PackageID  
JOIN DeliveryDriver dr ON r.DriverID = dr.DriverID  
WHERE t.DeliveryStatus = 'Out for Delivery';
```

Result:

```
1 • SELECT c.FullName, t.TrackingNumber, dr.DriverName, r.RouteName  
2   FROM PackageTrackingInformation t  
3   JOIN ShippingPackage p ON t.PackageID = p.PackageID  
4   JOIN DeliveryCustomers c ON p.CustomerID = c.CustomerID  
5   JOIN DeliveryRoute r ON p.PackageID = r.PackageID  
6   JOIN DeliveryDriver dr ON r.DriverID = dr.DriverID  
7   WHERE t.DeliveryStatus = 'Out for Delivery'  
8
```

FullName	TrackingNumber	DriverName	RouteName
Michael Smith	TRX0002	Samantha Hayes	Northside Loop
David Lee	TRX0004	Natalie Brooks	Lakeshore Delivery
Jessica Wilson	TRX0006	Natalie Brooks	Highland Route
Mia Robinson	TRX0013	Natalie Brooks	Seaview Delivery
Eliah Turner	TRX0018	Chris Parker	Golden Heights Drive

Result 1 < Action Output

#	Time	Action	Message	Duration / Fetch
235	23:22:27	SELECT c.FullName, t.TrackingNumber, t.DeliveryStatus FROM DeliveryCustomers c JOIN ShippingPackage p ON c.CustomerID = p.CustomerID WHERE t.DeliveryStatus = 'Out for Delivery'	Error Code: 1054. Unknown column 't.DeliveryStatus' in 'where clause'	0.000 sec
236	23:24:03	SELECT c.FullName, t.TrackingNumber, t.DeliveryStatus FROM DeliveryCustomers c JOIN ShippingPackage p ON c.CustomerID = p.CustomerID WHERE t.DeliveryStatus = 'Out for Delivery'	5 row(s) returned	0.000 sec / 0.000 sec
237	23:25:34	SELECT c.FullName, SUM(p.TotalShippingFee) AS TotalShippingPaid FROM DeliveryCustomer c JOIN ShippingPackage p ON c.CustomerID = p.CustomerID WHERE t.DeliveryStatus = 'Out for Delivery'	Error Code: 1146. Table 'Info_5707_project.deliverycustomer' doesn't exist.	0.000 sec
238	23:25:48	SELECT c.FullName, SUM(p.TotalShippingFee) AS TotalShippingPaid FROM DeliveryCustomers c JOIN ShippingPackage p ON c.CustomerID = p.CustomerID WHERE t.DeliveryStatus = 'Out for Delivery'	20 row(s) returned	0.015 sec / 0.000 sec
239	23:28:26	SELECT c.FullName, t.TrackingNumber, dr.DriverName, r.RouteName FROM PackageTrackingInformation t JOIN ShippingPackage p ON t.PackageID = p.PackageID JOIN DeliveryCustomers c ON p.CustomerID = c.CustomerID JOIN DeliveryRoute r ON p.PackageID = r.PackageID JOIN DeliveryDriver dr ON r.DriverID = dr.DriverID WHERE t.DeliveryStatus = 'Out for Delivery'	5 row(s) returned	0.000 sec / 0.000 sec

Query 4 : Delivery Status Count Summary

```
SELECT DeliveryStatus, COUNT(*) AS PackageCount  
FROM PackageTrackingInformation  
GROUP BY DeliveryStatus;
```

Result :

```

SELECT DeliveryStatus, COUNT(*) AS PackageCount
FROM PackageTrackingInformation
GROUP BY DeliveryStatus;

```

DeliveryStatus	PackageCount
Failed	5
Out for Delivery	5
In Transit	4
Pending	5
Delivered	1

Action Output:

- 236 23:24:03 SELECT c.FullName, t.TrackingNumber, t.DeliveryStatus FROM DeliveryCustomers c JOIN ShippingPackage t ON t.PackageTrackingInformationId = c.CustomerId WHERE t.DeliveryStatus = 'Delivered' GROUP BY t.DeliveryStatus
- 237 23:25:34 SELECT c.FullName, SUM(d.TotalShippingFee) AS TotalShippingPaid FROM DeliveryCustomer c JOIN Shipp... Error Code: 1146. Table 'Info_5707_project.deliverycustomer' doesn't exist.
- 238 23:25:48 SELECT c.FullName, SUM(d.TotalShippingFee) AS TotalShippingPaid FROM DeliveryCustomers c JOIN Shipp... 20 row(s) returned
- 239 23:28:26 SELECT c.FullName, t.TrackingNumber, dr.DrivenName, r.RouteName FROM PackageTrackingInformation t J... 5 row(s) returned
- 240 23:29:29 SELECT DeliveryStatus, COUNT(*) AS PackageCount FROM PackageTrackingInformation GROUP BY Deliv... 5 row(s) returned

Query 5 : Delivered Package Details for each Package on SCDelivery Date

SELECT d.TrackingNumber, d.StatusDescription, d.CurrentLocation

FROM PackageDeliveryStatus d

JOIN PackageTrackingInformation t ON d.TrackingNumber = t.TrackingNumber

WHERE t.LastUpdated = '2025-03-24' AND t.DeliveryStatus = 'Delivered';

Result :

```

SELECT d.TrackingNumber, d.StatusDescription, d.CurrentLocation
FROM PackageDeliveryStatus d
JOIN PackageTrackingInformation t ON d.TrackingNumber = t.TrackingNumber
WHERE t.LastUpdated = '2025-03-24' AND t.DeliveryStatus = 'Delivered';

```

TrackingNumber	StatusDescription	CurrentLocation
TRK0016	Returned to Sender	Denton Logistics Base

Action Output:

- 237 23:25:34 SELECT c.FullName, SUM(d.TotalShippingFee) AS TotalShippingPaid FROM DeliveryCustomer c JOIN Shipp... Error Code: 1146. Table 'Info_5707_project.deliverycustomer' doesn't exist.
- 238 23:25:48 SELECT c.FullName, SUM(d.TotalShippingFee) AS TotalShippingPaid FROM DeliveryCustomers c JOIN Shipp... 20 row(s) returned
- 239 23:28:26 SELECT c.FullName, t.TrackingNumber, dr.DrivenName, r.RouteName FROM PackageTrackingInformation t J... 5 row(s) returned
- 240 23:29:29 SELECT DeliveryStatus, COUNT(*) AS PackageCount FROM PackageTrackingInformation GROUP BY Deliv... 5 row(s) returned
- 241 23:30:14 SELECT d.TrackingNumber, d.StatusDescription, d.CurrentLocation FROM PackageDeliveryStatus d JOIN Pa... 1 row(s) returned

