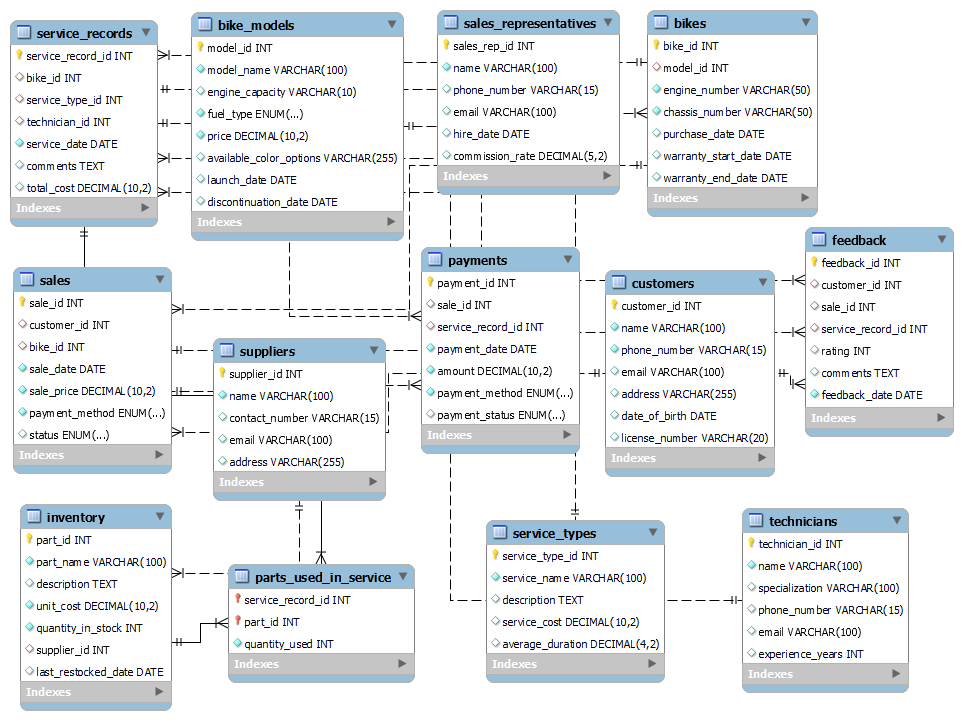
TVS Motor’S

Database management System

Overview: -

The **TVS Motor Database** is designed to manage a comprehensive set of data related to a bike showroom, including customers, bikes, sales, services, inventory, payments, and feedback. It tracks details about customers such as their personal information, contact details, and license numbers. The database includes information on various bike models, their specifications, pricing, availability, and launch details. Each bike is linked to its corresponding model and its unique bike details like engine and chassis numbers. The system also manages sales transactions, capturing the sale date, customer, bike sold, sale price, and payment method, ensuring smooth sales management. Additionally, the database handles the employees (sales representatives and technicians) involved in the sales and service processes.

For after-sales support, the database captures detailed service records, service types, technician information, and the parts used in service. It ensures that all parts and services are recorded for warranty and future servicing needs. The inventory system tracks available parts in stock, their costs, and suppliers. Payments are integrated into the system to track both sales and service payments, ensuring complete financial oversight. Feedback is captured from customers post-sale or service, allowing the showroom to monitor satisfaction and address any issues. This integrated structure ensures seamless management of all aspects of the showroom's operations, from sales to service to customer feedback.

ER Diagram

CREATE DATABASE **BIKE\_SHOWROOM**;

USE **BIKE\_SHOWROOM**;

Create TABLES

-- 01

CREATE TABLE **Customers** (

customer\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

phone\_number VARCHAR(15) NOT NULL UNIQUE,

email VARCHAR(100) UNIQUE,

address VARCHAR(255),

date\_of\_birth DATE,

license\_number VARCHAR(20) UNIQUE

);

-- 02

CREATE TABLE **Bike\_Models** (

model\_id INT PRIMARY KEY,

model\_name VARCHAR(100) NOT NULL,

engine\_capacity VARCHAR(10),

fuel\_type ENUM('Petrol', 'Electric') NOT NULL,

price DECIMAL(10, 2) NOT NULL,

available\_color\_options VARCHAR(255),

launch\_date DATE,

discontinuation\_date DATE

);

-- 03

CREATE TABLE **Bikes** (

bike\_id INT PRIMARY KEY,

model\_id INT,

engine\_number VARCHAR(50) UNIQUE NOT NULL,

chassis\_number VARCHAR(50) UNIQUE NOT NULL,

purchase\_date DATE,

warranty\_start\_date DATE,

warranty\_end\_date DATE,

FOREIGN KEY (model\_id) REFERENCES Bike\_Models(model\_id)

);

-- 04

CREATE TABLE **Sales** (

sale\_id INT PRIMARY KEY,

customer\_id INT,

bike\_id INT,

sale\_date DATE NOT NULL,

sale\_price DECIMAL(10, 2) NOT NULL,

payment\_method ENUM('Cash', 'Financing', 'Debit Card') NOT NULL,

status ENUM('Completed', 'Pending', 'Canceled') DEFAULT 'Pending',

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (bike\_id) REFERENCES Bikes(bike\_id)

);

-- 05

CREATE TABLE **Sales\_Representatives** (

sales\_rep\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

phone\_number VARCHAR(15) UNIQUE,

email VARCHAR(100) UNIQUE,

hire\_date DATE,

commission\_rate DECIMAL(5, 2)

);

-- 06

CREATE TABLE **Service\_Types** (

service\_type\_id INT PRIMARY KEY,

service\_name VARCHAR(100) NOT NULL,

description TEXT,

service\_cost DECIMAL(10, 2),

average\_duration DECIMAL(4, 2) -- in hours

);

-- 07

CREATE TABLE **Technicians** (

technician\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

specialization VARCHAR(100),

phone\_number VARCHAR(15) UNIQUE,

email VARCHAR(100) UNIQUE,

experience\_years INT

);

-- 08

CREATE TABLE **Service\_Records** (

service\_record\_id INT PRIMARY KEY,

bike\_id INT,

service\_type\_id INT,

technician\_id INT,

service\_date DATE NOT NULL,

comments TEXT,

total\_cost DECIMAL(10, 2),

FOREIGN KEY (bike\_id) REFERENCES Bikes(bike\_id),

FOREIGN KEY (service\_type\_id) REFERENCES Service\_Types(service\_type\_id),

FOREIGN KEY (technician\_id) REFERENCES Technicians(technician\_id)

);

-- 09

CREATE TABLE **Inventory** (

part\_id INT PRIMARY KEY,

part\_name VARCHAR(100) NOT NULL,

description TEXT,

unit\_cost DECIMAL(10, 2) NOT NULL,

quantity\_in\_stock INT NOT NULL,

supplier\_id INT,

last\_restocked\_date DATE,

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id)

);

-- 10

CREATE TABLE **Suppliers** (

supplier\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

contact\_number VARCHAR(15),

email VARCHAR(100),

address VARCHAR(255)

);

-- 11

CREATE TABLE **Parts\_Used\_In\_Service** (

service\_record\_id INT,

part\_id INT,

quantity\_used INT NOT NULL,

FOREIGN KEY (service\_record\_id) REFERENCES Service\_Records(service\_record\_id),

FOREIGN KEY (part\_id) REFERENCES Inventory(part\_id),

PRIMARY KEY (service\_record\_id, part\_id)

);

-- 12

CREATE TABLE **Payments** (

payment\_id INT PRIMARY KEY,

sale\_id INT,

service\_record\_id INT,

payment\_date DATE NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

payment\_method ENUM('Cash', 'Debit Card', 'UPI') NOT NULL,

payment\_status ENUM('Paid', 'Unpaid', 'Pending') DEFAULT 'Pending',

FOREIGN KEY (sale\_id) REFERENCES Sales(sale\_id),

FOREIGN KEY (service\_record\_id) REFERENCES Service\_Records(service\_record\_id)

);

-- 13

CREATE TABLE **Feedback** (

feedback\_id INT PRIMARY KEY,

customer\_id INT,

sale\_id INT,

service\_record\_id INT,

rating INT CHECK (rating BETWEEN 1 AND 5),

comments TEXT,

feedback\_date DATE NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (sale\_id) REFERENCES Sales(sale\_id),

FOREIGN KEY (service\_record\_id) REFERENCES Service\_Records(service\_record\_id)

);

INSERT VALUES

INSERT INTO **Customers** (customer\_id, name, phone\_number, email, address, date\_of\_birth, license\_number) VALUES

(1, 'Rohan Sharma', '9876543210', 'rohan.sharma@example.com', 'Mumbai, Maharashtra', '1990-05-15', 'MH01A12345'),

(2, 'Aarti Gupta', '9123456789', 'aarti.gupta@example.com', 'Delhi, Delhi', '1985-11-20', 'DL05B67890'),

(3, 'Vikram Desai', '9823456780', 'vikram.desai@example.com', 'Bangalore, Karnataka', '1993-04-12', 'KA03C11223'),

(4, 'Sanjana Iyer', '9876543123', 'sanjana.iyer@example.com', 'Chennai, Tamil Nadu', '1988-09-18', 'TN09D22456'),

(5, 'Abhishek Verma', '9988776655', 'abhishek.verma@example.com', 'Pune, Maharashtra', '1995-02-02', 'MH12E33445'),

(6, 'Rahul Mehta', '9123456721', 'rahul.mehta@example.com', 'Ahmedabad, Gujarat', '1989-03-11', 'GJ01A78945'),

(7, 'Pooja Singh', '9876543222', 'pooja.singh@example.com', 'Kolkata, West Bengal', '1992-08-27', 'WB09B12345'),

(8, 'Aniket Kumar', '9801234567', 'aniket.kumar@example.com', 'Lucknow, Uttar Pradesh', '1985-07-16', 'UP14C56789'),

(9, 'Rekha Patel', '9776543219', 'rekha.patel@example.com', 'Indore, Madhya Pradesh', '1993-12-23', 'MP09D34567'),

(10, 'Neeraj Kapoor', '9445678901', 'neeraj.kapoor@example.com', 'Hyderabad, Telangana', '1987-01-29', 'TS08E98765');

INSERT INTO **Bike\_Models** (model\_id, model\_name, engine\_capacity, fuel\_type, price, available\_color\_options, launch\_date) VALUES

(1, 'TVS Apache RTR 160', '160cc', 'Petrol', 114000, 'Red, Blue, Black', '2020-01-15'),

(2, 'TVS Ntorq 125', '125cc', 'Petrol', 83000, 'Yellow, Red, Blue, Black', '2019-06-10'),

(3, 'TVS Jupiter', '110cc', 'Petrol', 75000, 'Grey, Blue, White, Black', '2018-09-20'),

(4, 'TVS XL100', '100cc', 'Petrol', 54000, 'Green, Black', '2017-04-25'),

(5, 'TVS iQube Electric', '0cc', 'Electric', 140000, 'White, Black', '2021-02-01'),

(6, 'TVS Apache RTR 200', '200cc', 'Petrol', 130000, 'Red, Black, White', '2021-07-10'),

(7, 'TVS Radeon', '110cc', 'Petrol', 70000, 'Black, Grey, Beige', '2018-10-12'),

(8, 'TVS Star City+', '110cc', 'Petrol', 69000, 'Red, Black, Grey', '2019-08-01'),

(9, 'TVS Sport', '100cc', 'Petrol', 62000, 'Blue, Black, Red', '2020-05-15'),

(10, 'TVS Zest Electric', '0cc', 'Electric', 145000, 'Yellow, Black', '2023-01-30'),

(11, 'TVS Electron+', '0cc', 'Electric', 160000, 'Blue, Black, White', '2024-02-15'),

(12, 'TVS Blaze E+', '0cc', 'Electric', 175000, 'Red, Silver', '2024-03-22'),

(13, 'TVS Spark E', '0cc', 'Electric', 140000, 'Black, Blue', '2024-04-05'),

(14, 'TVS Evo Electric', '0cc', 'Electric', 180000, 'White, Silver', '2024-06-01'),

(15, 'TVS Creon', '0cc', 'Electric', 170000, 'White, Red, Grey', '2023-08-05');

INSERT INTO **Bikes** (bike\_id, model\_id, engine\_number, chassis\_number, purchase\_date, warranty\_start\_date, warranty\_end\_date) VALUES

(1, 1, 'ENG123456789', 'CHS987654321', '2024-01-20', '2024-01-20', '2026-01-20'),

(2, 2, 'ENG987654321', 'CHS123456789', '2023-11-05', '2023-11-05', '2025-11-05'),

(3, 3, 'ENG567890123', 'CHS456789012', '2024-03-15', '2024-03-15', '2026-03-15'),

(4, 4, 'ENG135790246', 'CHS357924680', '2024-05-30', '2024-05-30', '2026-05-30'),

(5, 5, 'ENG246813579', 'CHS468135792', '2023-10-10', '2023-10-10', '2025-10-10'),

(6, 6, 'ENG112233445', 'CHS554433221', '2023-09-18', '2023-09-18', '2025-09-18'),

(7, 7, 'ENG223344556', 'CHS665544332', '2024-01-10', '2024-01-10', '2026-01-10'),

(8, 8, 'ENG334455667', 'CHS776655443', '2024-02-22', '2024-02-22', '2026-02-22'),

(9, 9, 'ENG445566778', 'CHS887766554', '2023-12-01', '2023-12-01', '2025-12-01'),

(10, 10, 'ENG556677889', 'CHS998877665', '2024-03-14', '2024-03-14', '2026-03-14'),

(11, 11, 'ENG667788990', 'CHS009988776', '2024-04-05', '2024-04-05', '2026-04-05'),

(12, 12, 'ENG778899001', 'CHS110099887', '2024-05-20', '2024-05-20', '2026-05-20'),

(13, 13, 'ENG889900112', 'CHS221100998', '2023-11-22', '2023-11-22', '2025-11-22'),

(14, 14, 'ENG990011223', 'CHS332211009', '2024-06-01', '2024-06-01', '2026-06-01'),

(15, 15, 'ENG001122334', 'CHS443322110', '2024-07-10', '2024-07-10', '2026-07-10');

INSERT INTO **Sales** (sale\_id, customer\_id, bike\_id, sale\_date, sale\_price, payment\_method, status) VALUES

(1, 1, 1, '2024-01-20', 114000, 'Cash', 'Completed'),

(2, 1, 2, '2023-11-05', 83000, 'Debit Card', 'Completed'),

(3, 3, 3, '2024-03-15', 75000, 'Financing', 'Pending'),

(4, 2, 4, '2024-05-30', 54000, 'Cash', 'Completed'),

(5, 1, 5, '2023-10-10', 140000, 'Debit Card', 'Completed'),

(6, 6, 6, '2023-12-15', 130000, 'Cash', 'Completed'),

(7, 10, 7, '2024-02-01', 70000, 'Financing', 'Completed'),

(8, 8, 8, '2024-03-22', 69000, 'Debit Card', 'Completed'),

(9, 9, 9, '2024-01-10', 62000, 'Cash', 'Completed'),

(10, 5, 10, '2024-04-30', 145000, 'Cash', 'Pending'),

(11, 4, 11, '2024-05-05', 160000, 'Debit Card', 'Completed'),

(12, 4, 12, '2024-06-10', 175000, 'Financing', 'Completed'),

(13, 5, 13, '2024-07-15', 140000, 'Cash', 'Completed'),

(14, 7, 14, '2024-08-20', 180000, 'Debit Card', 'Pending'),

(15, 10, 15, '2024-09-25', 170000, 'Cash', 'Completed');

INSERT INTO **Service\_Types** (service\_type\_id, service\_name, description, service\_cost, average\_duration) VALUES

(1, 'Regular Maintenance', 'Basic checkup and oil change', 1000, 1.5),

(2, 'Engine Overhaul', 'Complete engine inspection and repair', 5000, 4),

(3, 'Electrical Repair', 'Repair of electrical components', 1500, 2),

(4, 'Brake Service', 'Brake inspection and replacement', 800, 1),

(5, 'Tyre Replacement', 'Replace old tyres with new ones', 1200, 1);

INSERT INTO **Service\_Records** (service\_record\_id, bike\_id, service\_type\_id, technician\_id, service\_date, comments, total\_cost) VALUES

(1, 1, 1, 101, '2024-01-22', 'Regular maintenance done', 1000),

(2, 2, 3, 102, '2023-11-10', 'Fixed electrical wiring issues', 1500),

(3, 3, 2, 101, '2024-03-20', 'Engine overhaul completed', 5000),

(4, 4, 4, 103, '2024-06-01', 'Brake pads replaced', 800),

(5, 5, 5, 104, '2023-10-15', 'Front tyre replaced', 1200),

(6, 6, 1, 101, '2024-01-15', 'Regular maintenance performed', 900),

(7, 7, 2, 102, '2024-02-05', 'Oil change and filter replacement', 1200),

(8, 8, 3, 103, '2024-03-10', 'Battery replacement completed', 2000),

(9, 9, 4, 104, '2024-04-22', 'Chain lubrication and adjustment', 500),

(10, 10, 5, 101, '2024-05-01', 'General check-up', 700),

(11, 11, 1, 102, '2024-06-15', 'Brake system inspection', 800),

(12, 12, 2, 103, '2024-07-20', 'Tire rotation and balance', 1100),

(13, 13, 3, 104, '2024-08-30', 'Engine diagnostic check', 1500),

(14, 14, 4, 101, '2024-09-05', 'Clutch adjustment done', 900),

(15, 15, 5, 102, '2024-10-18', 'Battery check and replacement', 1800);

INSERT INTO **Technicians** (technician\_id, name, specialization, phone\_number, email, experience\_years) VALUES

(101, 'Arjun Patel', 'Engine Repair', '9876543120', 'arjun.patel@tvsmotors.com', 5),

(102, 'Priya Nair', 'Electrical Work', '9123456710', 'priya.nair@tvsmotors.com', 3),

(103, 'Manish Kumar', 'Brake Service', '9987654321', 'manish.kumar@tvsmotors.com', 4),

(104, 'Swati Joshi', 'Tyre Replacement', '9876501234', 'swati.joshi@tvsmotors.com', 2);

INSERT INTO **Inventory** (part\_id, part\_name, description, unit\_cost, quantity\_in\_stock, supplier\_id, last\_restocked\_date) VALUES

(1, 'Oil Filter', 'Engine oil filter', 200, 50, 201, '2024-01-10'),

(2, 'Brake Pads', 'Front and rear brake pads', 400, 30, 202, '2024-02-15'),

(3, 'Spark Plug', 'Ignition spark plug', 150, 100, 201, '2024-03-01'),

(4, 'Battery', 'Electric bike battery', 5000, 10, 203, '2024-01-25'),

(5, 'Tyre', 'Standard 17-inch tyre', 1200, 20, 204, '2024-02-05');

INSERT INTO **Suppliers** (supplier\_id, name, contact\_number, email, address) VALUES

(201, 'Auto Parts Ltd', '9123456789', 'contact@autopartsltd.in', 'Mumbai, Maharashtra'),

(202, 'BrakeMaster Supplies', '9876543210', 'sales@brakemaster.com', 'Delhi, Delhi'),

(203, 'Battery World', '9988776655', 'support@batteryworld.in', 'Bangalore, Karnataka'),

(204, 'TyreHouse', '9898989898', 'info@tyrehouse.com', 'Chennai, Tamil Nadu');

INSERT INTO **Payments** (payment\_id, sale\_id, service\_record\_id, payment\_date, amount, payment\_method, payment\_status) VALUES

(1, 1, NULL, '2024-01-20', 114000, 'Cash', 'Paid'),

(2, 2, NULL, '2023-11-05', 83000, 'Debit Card', 'Paid'),

(3, NULL, 1, '2024-01-22', 1000, 'Cash', 'Paid'),

(4, NULL, 3, '2024-03-20', 5000, 'Cash', 'Pending'),

(5, 5, NULL, '2023-10-10', 140000, 'Debit Card', 'Paid'),

(6, 3, NULL, '2024-03-15', 75000, 'Debit Card', 'Pending'),

(7, 4, NULL, '2024-05-30', 54000, 'Cash', 'Paid'),

(8, NULL, 2, '2023-11-15', 1500, 'UPI', 'Paid'),

(9, NULL, 4, '2024-06-01', 800, 'UPI', 'Paid'),

(10, 6, NULL, '2024-01-15', 130000, 'Debit Card', 'Paid'),

(11, NULL, 5, '2024-03-10', 1200, 'Cash', 'Pending'),

(12, 7, NULL, '2024-04-22', 70000, 'Debit Card', 'Paid'),

(13, NULL, 6, '2024-07-01', 500, 'UPI', 'Paid'),

(14, 8, NULL, '2024-05-10', 69000, 'Cash', 'Paid'),

(15, NULL, 7, '2024-08-25', 2000, 'UPI', 'Paid');

INSERT INTO **Feedback** (feedback\_id, customer\_id, sale\_id, service\_record\_id, rating, comments, feedback\_date) VALUES

(1, 1, 1, NULL, 5, 'Great service and bike quality!', '2024-01-25'),

(2, 2, 2, NULL, 4, 'Happy with the purchase, but delivery was delayed.', '2023-11-15'),

(3, 2, NULL, 3, 5, 'Excellent service, bike feels new!', '2024-03-25'),

(4, 4, NULL, 4, 3, 'Brake service was satisfactory.', '2024-06-05'),

(5, 5, 5, NULL, 5, 'Love the iQube, fantastic experience!', '2023-10-20'),

(6, 3, 6, NULL, 4, 'Overall good experience, bike is smooth.', '2024-02-05'),

(7, 7, NULL, 2, 5, 'Service was quick and effective!', '2024-04-10'),

(8, 8, NULL, 5, 2, 'Expected better quality after service.', '2024-05-15'),

(9, 9, NULL, 3, 4, 'Bike performance improved significantly after service.', '2024-03-30'),

(10, 1, NULL, NULL, 5, 'The sales staff was very helpful and knowledgeable.', '2024-01-10'),

(11, 9, 7, NULL, 4, 'Great bike, but the color options are limited.', '2024-02-20'),

(12, 10, NULL, 6, 3, 'Satisfactory service, but could be improved.', '2024-06-15'),

(13, 2, NULL, 4, 5, 'Amazing experience with the service team!', '2024-07-01'),

(14, 1, 8, NULL, 4, 'Pleased with the bike performance.', '2024-05-20'),

(15, 3, NULL, 1, 5, 'Love my new bike! Highly recommend!', '2024-08-10');

INSERT INTO **Sales\_Representatives** (sales\_rep\_id, name, phone\_number, email, hire\_date, commission\_rate) VALUES

(1, 'Amit Sharma', '9876543210', 'amit.sharma@example.com', '2021-04-15', 0.05),

(2, 'Priya Desai', '9823456789', 'priya.desai@example.com', '2022-06-20', 0.04),

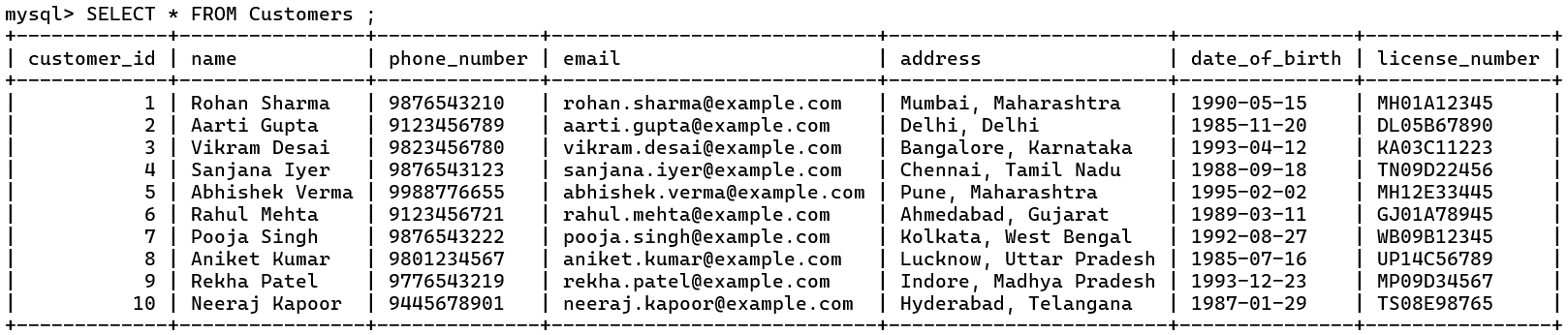
(3, 'Rajesh Kumar', '9988776655', 'rajesh.kumar@example.com', '2023-01-10', 0.06),

(4, 'Sneha Verma', '9876512345', 'sneha.verma@example.com', '2020-09-01', 0.03),

(5, 'Vikram Joshi', '9765432109', 'vikram.joshi@example.com', '2019-11-05', 0.07),

(6, 'Meera Patel', '9898765432', 'meera.patel@example.com', '2023-07-18', 0.05);

Select & DESC

SELECT \* FROM Customers ;

SELECT \* FROM Bike\_Models ;A group of black text

Description automatically generated

SELECT \* FROM Bikes ;A table of numbers and letters

Description automatically generated

SELECT \* FROM Sales ; A screenshot of a table

Description automatically generated

SELECT \* FROM Service\_Types ;A close-up of a document

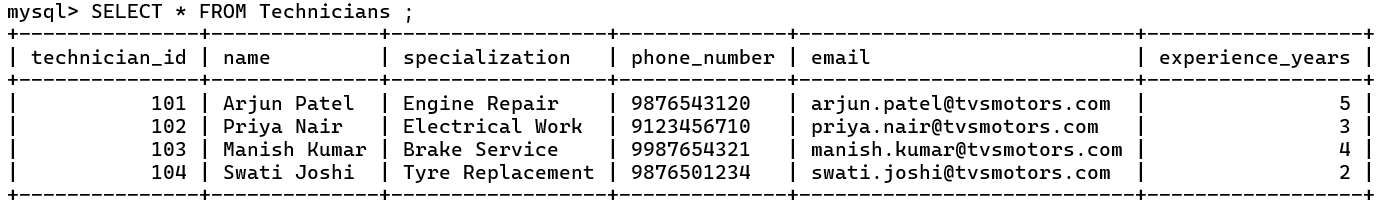
Description automatically generated

SELECT \* FROM Sales\_Representatives ;A close-up of a document

Description automatically generated

A table of numbers and numbers

Description automatically generated with medium confidenceSELECT \* FROM Service\_Records ;

SELECT \* FROM Technicians ;

SELECT \* FROM Inventory ;A close-up of a number

Description automatically generated

SELECT \* FROM Suppliers ;A close-up of a number

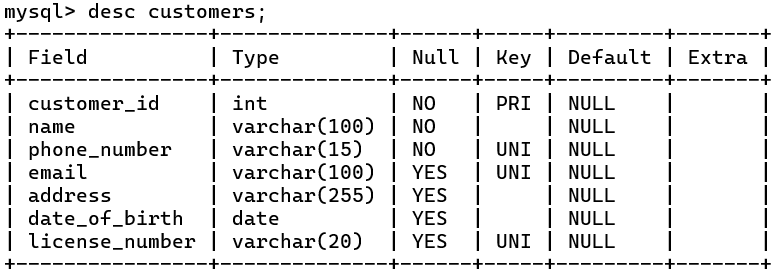
Description automatically generated

A screenshot of a document

Description automatically generatedSELECT \* FROM Payments ;

SELECT \* FROM Feedback ; A black and white text on a white background

Description automatically generated

DESC Customers ; 

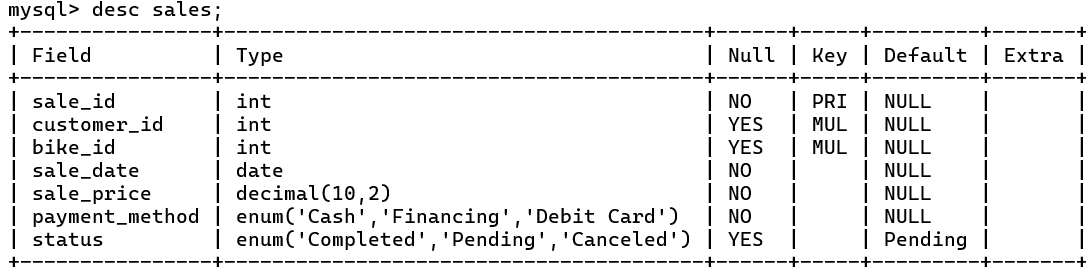
DESC Bike\_Models ;A black and white text on a white background

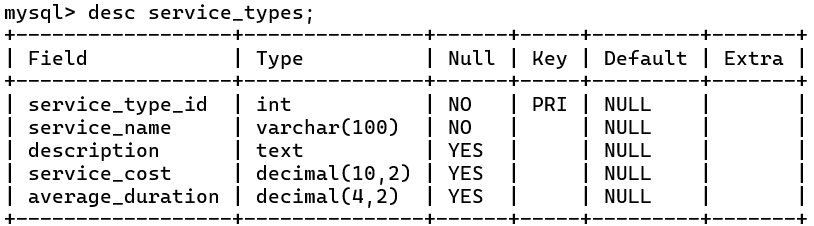
Description automatically generated

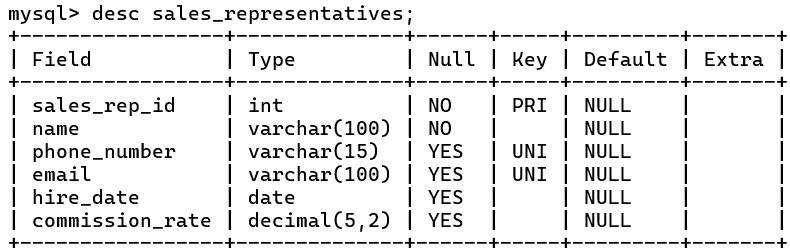
DESC Bikes ;

A white sheet with black text

Description automatically generated

DESC Sales ; 

DESC Service\_Types ;

DESC Sales\_Representatives ;

DESC Service\_Records ;A black and white sheet of paper with text

Description automatically generated

DESC Technicians ;A table with black text

Description automatically generated with medium confidence

DESC Inventory ;A close-up of a code

Description automatically generated

A table with black text

Description automatically generated with medium confidenceDESC Suppliers ;

A close-up of a card

Description automatically generatedDESC Payments ;

DESC Feedback ; A table of text and words

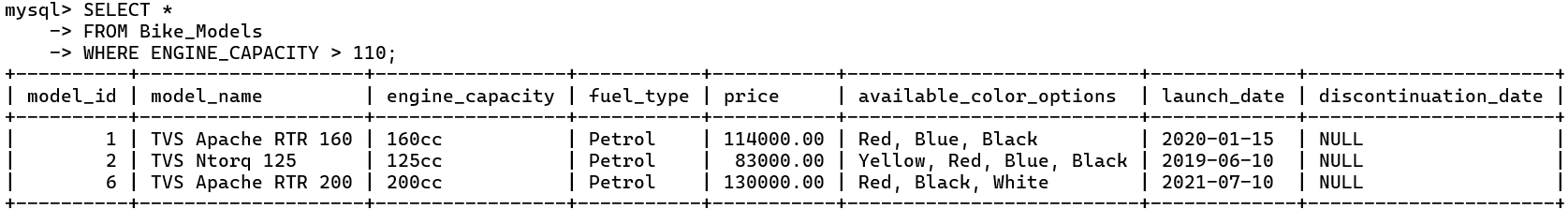
Description automatically generated with medium confidence

\*\*\* QUESTIONS \*\*\*

-- find engine capacities greater than 110cc

SELECT \*

FROM Bike\_Models

WHERE ENGINE\_CAPACITY > 110;

-- CHANGE CUSTOMER ADD FOR CUST\_ID 3?

UPDATE CUSTOMERS

SET ADDRESS = 'MUMBAI,MAHARASHTRA'

WHERE CUSTOMER\_ID = 3;

-- List all bike models with their price and fuel type

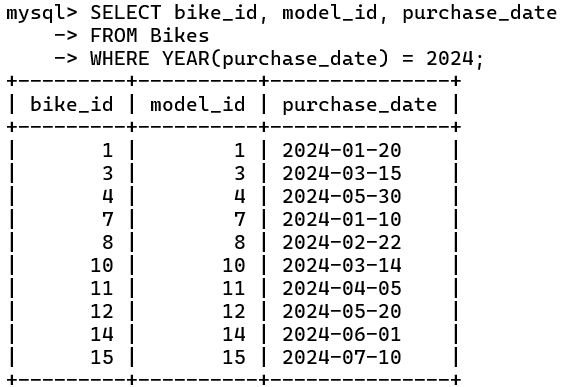
SELECT model\_name, price, fuel\_type FROM Bike\_Models;A screenshot of a computer

Description automatically generated

-- Find all bikes purchased in the year 2024

SELECT bike\_id, model\_id, purchase\_date

FROM Bikes

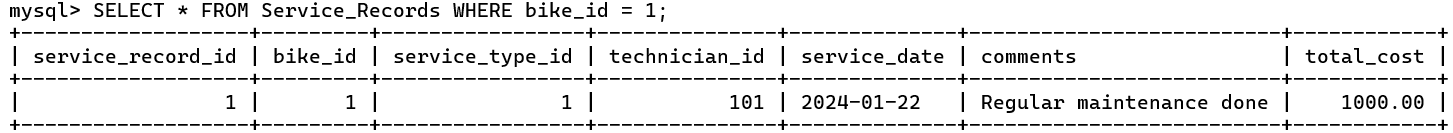
WHERE YEAR(purchase\_date) = 2024;

-- Get the total sales amount

SELECT SUM(sale\_price) AS total\_sales FROM Sales;A white background with black text

Description automatically generated

-- Find all service records for a specific bike (e.g., bike\_id = 1)

SELECT \* FROM Service\_Records WHERE bike\_id = 1;

-- List customers who bought bikes with a specific model (e.g., model\_id = 5)

SELECT c.name, c.phone\_number

FROM Customers AS c

JOIN Sales AS s ON c.customer\_id = s.customer\_id

JOIN Bikes AS b ON s.bike\_id = b.bike\_id

WHERE b.model\_id = 5;A black text on a white background

Description automatically generated

-- Get feedback for all services with a rating of 4 or higher

SELECT customer\_id, sale\_id, service\_record\_id, rating, comments

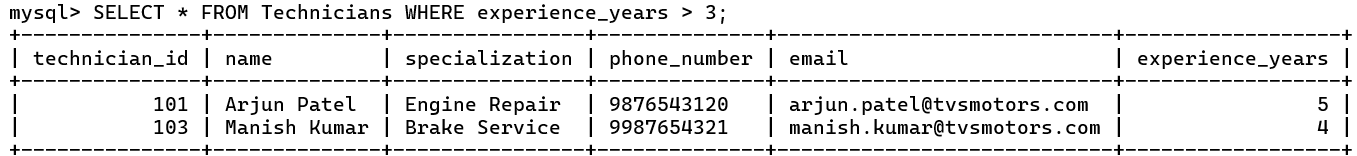
FROM Feedback

WHERE rating >= 4;

A screenshot of a computer

Description automatically generated

-- Find all technicians with more than 3 years of experience

SELECT \* FROM Technicians WHERE experience\_years > 3;

-- Show all bike models available in Electric fuel type

SELECT model\_name, price

FROM Bike\_Models

WHERE fuel\_type = 'Electric';A screenshot of a computer code

Description automatically generated

-- List out sales representatives hired before 2024

SELECT \* FROM Sales\_Representatives

WHERE hire\_date < '2023-01-01';

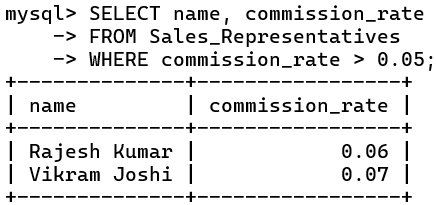
A close-up of a document

Description automatically generated

-- Find Sales Representatives with a Commission Rate Above 5%

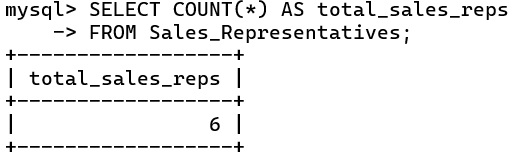
SELECT name, commission\_rate

FROM Sales\_Representatives

WHERE commission\_rate > 0.05;

-- Count Total Number of Sales Representatives

SELECT COUNT(\*) AS total\_sales\_reps

FROM Sales\_Representatives;

-- Find Sales Representative by Phone Number

SELECT \*

FROM Sales\_Representatives

A close-up of a list

Description automatically generatedWHERE phone\_number = '9876543210';

-- Update Commission Rate for a Specific Sales Representative

UPDATE Sales\_Representatives

SET commission\_rate = 0.06

WHERE sales\_rep\_id = 3;

-- List Sales Representatives and Their Hire Dates in Descending Order

SELECT name, hire\_date

FROM Sales\_Representatives

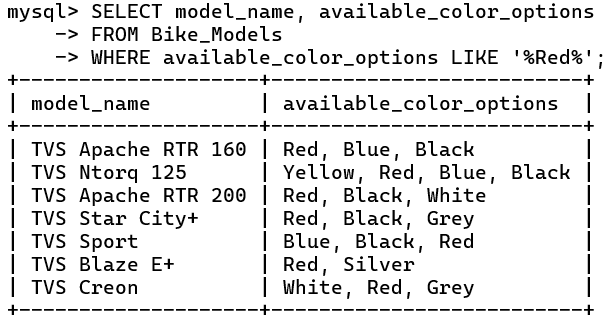
ORDER BY hire\_date DESC;A white text with black text

Description automatically generated

-- Find Bikes Available in a Specific Color:

SELECT model\_name, available\_color\_options

FROM Bike\_Models

WHERE available\_color\_options LIKE '%Red%';

-- Retrieve Bike Models Launched After a Certain Date:

SELECT model\_name, launch\_date

FROM Bike\_Models

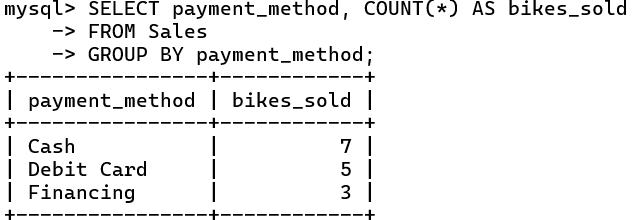
WHERE launch\_date > '2023-01-01';A white paper with black text

Description automatically generated

-- Count Bikes Sold by Payment Method:

SELECT payment\_method, COUNT(\*) AS bikes\_sold

FROM Sales

GROUP BY payment\_method;

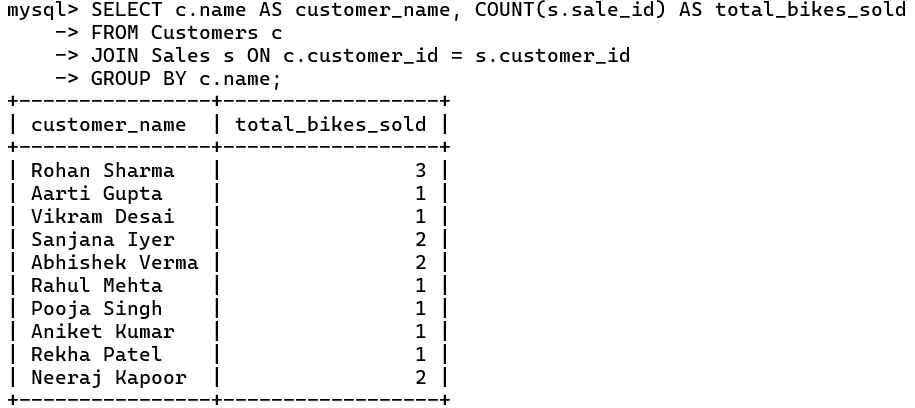
-- Find the Total Number of Bikes Sold to Each Customer:

SELECT c.name AS customer\_name, COUNT(s.sale\_id) AS total\_bikes\_sold

FROM Customers c

JOIN Sales s ON c.customer\_id = s.customer\_id

GROUP BY c.name;

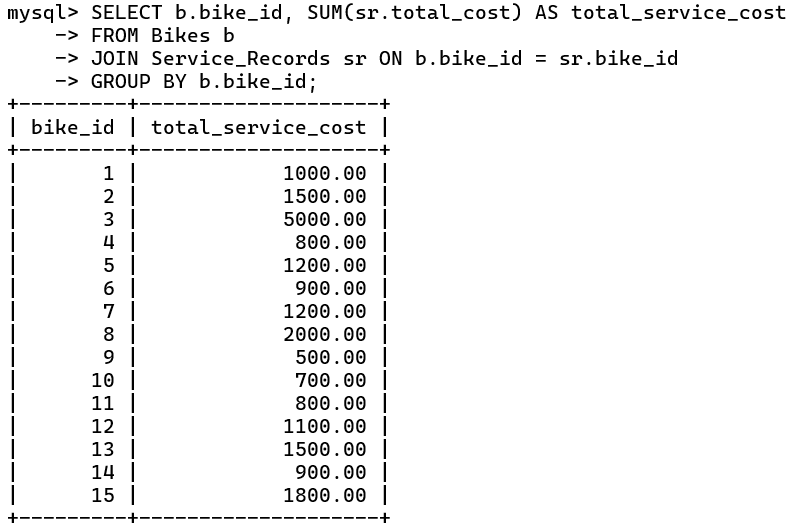


-- Calculate Total Service Costs per Bike:

SELECT b.bike\_id, SUM(sr.total\_cost) AS total\_service\_cost

FROM Bikes b

JOIN Service\_Records sr ON b.bike\_id = sr.bike\_id

GROUP BY b.bike\_id;

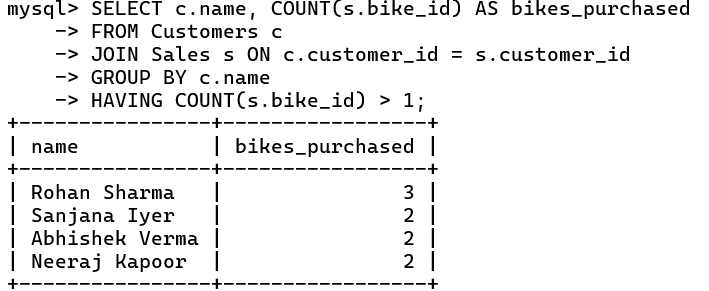
-- Find Customers Who Bought More Than One Bike:

SELECT c.name, COUNT(s.bike\_id) AS bikes\_purchased

FROM Customers c

JOIN Sales s ON c.customer\_id = s.customer\_id

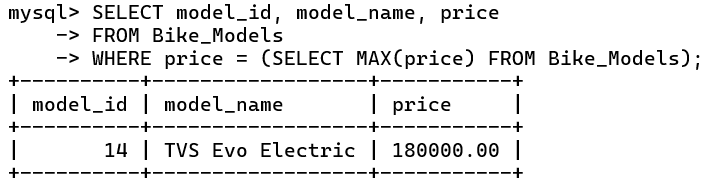
GROUP BY c.name

HAVING COUNT(s.bike\_id) > 1;

-- Find the bike model with the highest price.

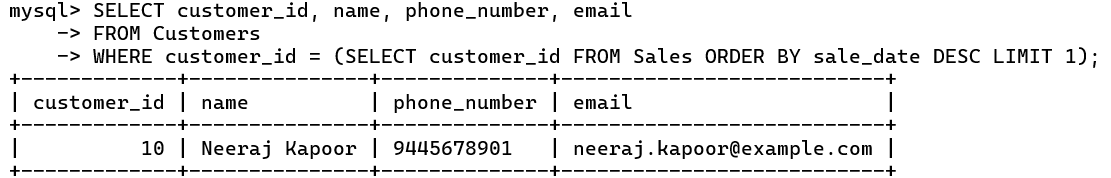
SELECT model\_id, model\_name, price

FROM Bike\_Models

WHERE price = (SELECT MAX(price) FROM Bike\_Models);

-- Which customer made the most recent purchase?

SELECT customer\_id, name, phone\_number, email FROM Customers

WHERE customer\_id = (SELECT customer\_id FROM Sales ORDER BY sale\_date DESC LIMIT 1);

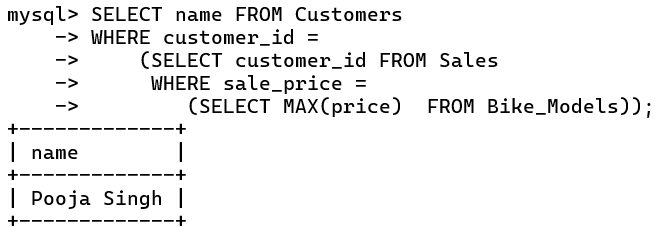
-- Find the customer who made the highest sale for the most expensive bike model.

SELECT name FROM Customers

WHERE customer\_id =

(SELECT customer\_id FROM Sales

WHERE sale\_price =

(SELECT MAX(price) FROM Bike\_Models));

-- Finding the Most Expensive Bike Sold by Each Sales Representative IN DESC ORDER:

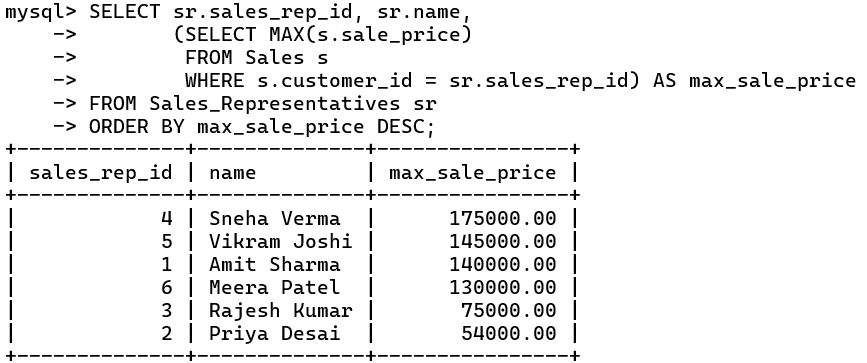
SELECT sr.sales\_rep\_id, sr.name,

(SELECT MAX(s.sale\_price)

FROM Sales s

WHERE s.customer\_id = sr.sales\_rep\_id) AS max\_sale\_price

FROM Sales\_Representatives sr

ORDER BY max\_sale\_price DESC;

-- Finding the Top 3 Most Popular Bike Models Sold

SELECT bm.model\_name, COUNT(s.bike\_id) AS sales\_count

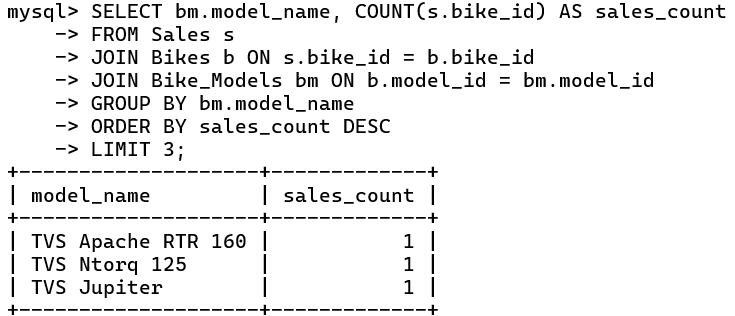
FROM Sales s

JOIN Bikes b ON s.bike\_id = b.bike\_id

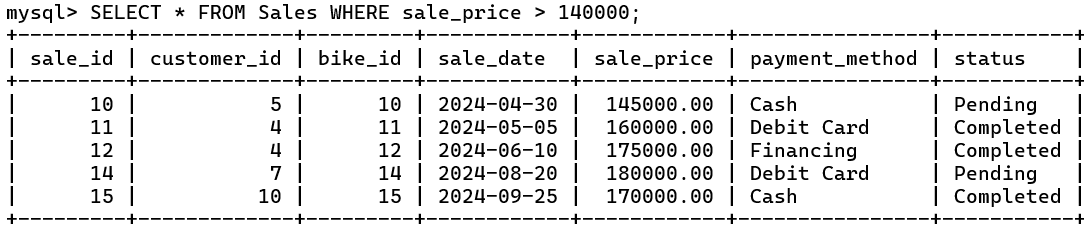
JOIN Bike\_Models bm ON b.model\_id = bm.model\_id

GROUP BY bm.model\_name

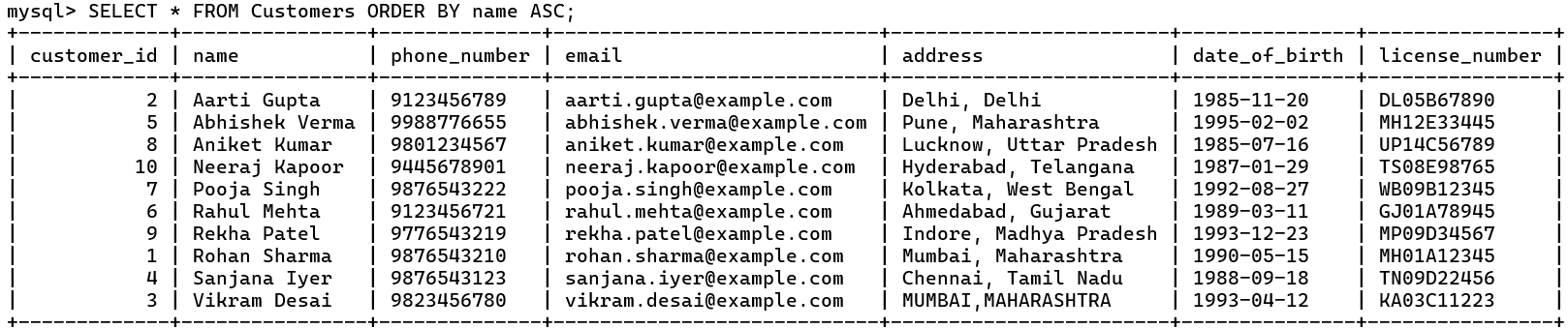
ORDER BY sales\_count DESC

LIMIT 3;

-- retrieves all columns from the Sales table where the sale\_price is greater than 1,40,000.

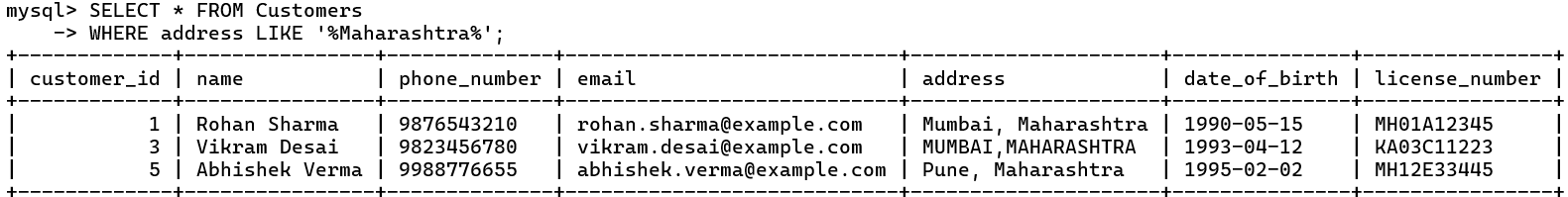
SELECT \* FROM Sales WHERE sale\_price > 140000;

-- retrieves CUST NAME IN ASC ORDER

SELECT \* FROM Customers ORDER BY name ASC;

-- retrieves THE CUSTOMER WHO LIVE IN MAHARASHTRA

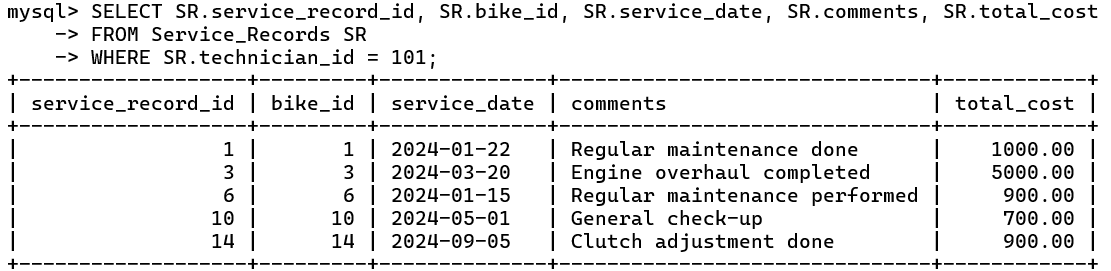
SELECT \* FROM Customers

WHERE address LIKE '%Maharashtra%';

-- Get all the services performed by a specific technician

SELECT SR.service\_record\_id, SR.bike\_id, SR.service\_date, SR.comments, SR.total\_cost

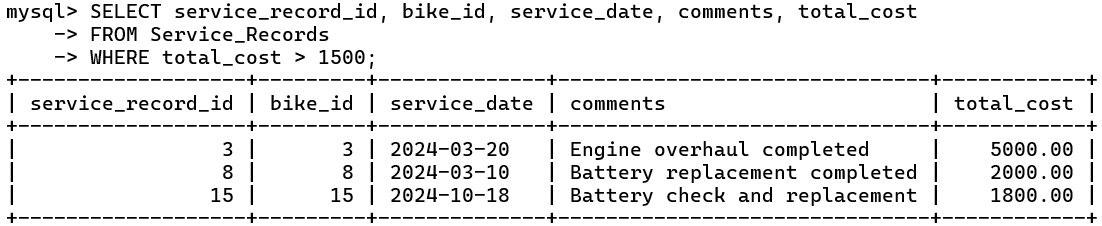
FROM Service\_Records SR

WHERE SR.technician\_id = 101;

-- Find all service records where the total cost is greater than 1500:

SELECT service\_record\_id, bike\_id, service\_date, comments, total\_cost

FROM Service\_Records

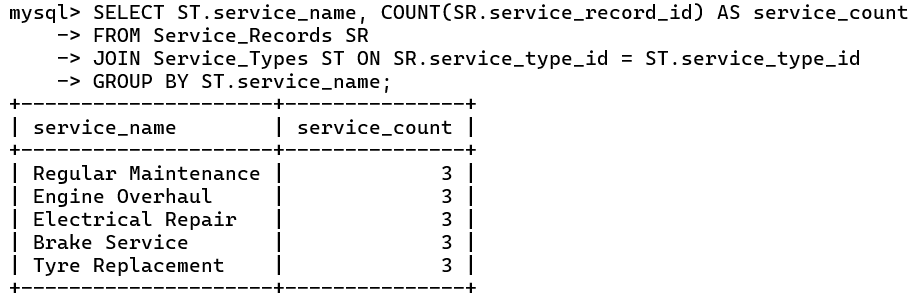
WHERE total\_cost > 1500;

-- Find the number of services performed for each service type:

SELECT ST.service\_name, COUNT(SR.service\_record\_id) AS service\_count

FROM Service\_Records SR

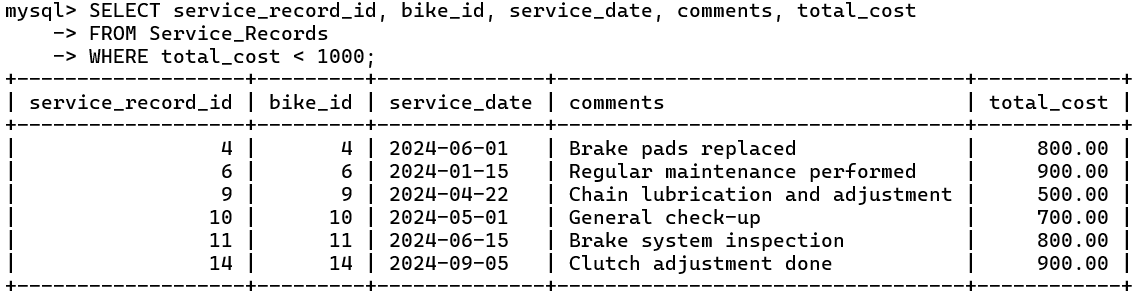
JOIN Service\_Types ST ON SR.service\_type\_id = ST.service\_type\_id

GROUP BY ST.service\_name;

-- Get all records where the service cost is less than 1000:

SELECT service\_record\_id, bike\_id, service\_date, comments, total\_cost

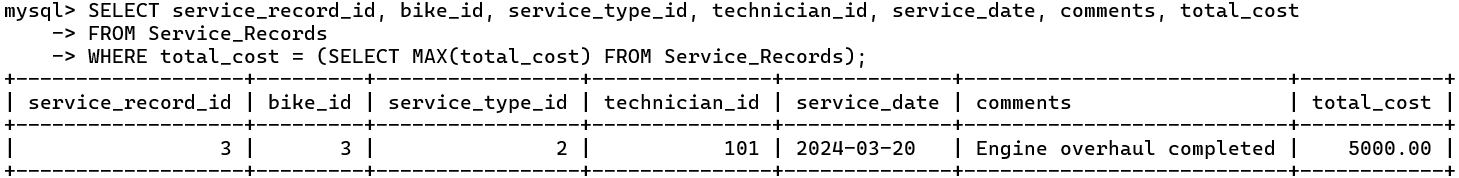
FROM Service\_Records

WHERE total\_cost < 1000;

-- Find the most expensive service performed (maximum total cost):

SELECT service\_record\_id, bike\_id, service\_type\_id, technician\_id, service\_date, comments, total\_cost

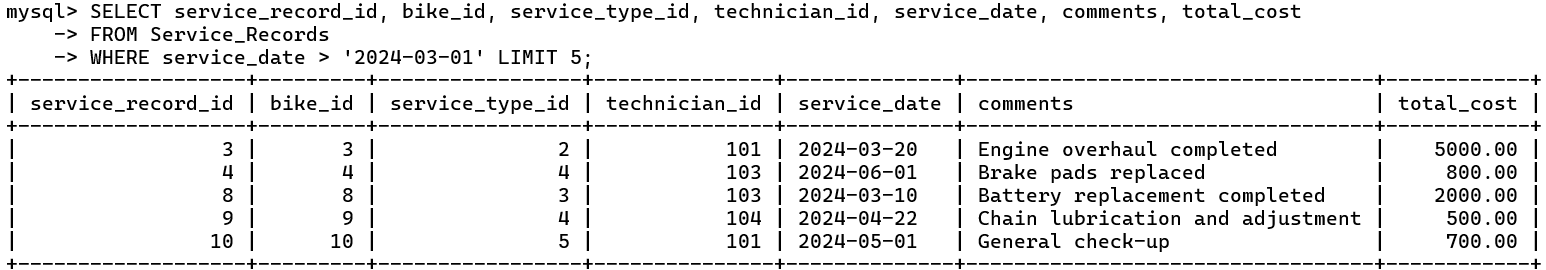
FROM Service\_Records

WHERE total\_cost = (SELECT MAX(total\_cost) FROM Service\_Records);

-- Find all services where the service date is after March 1, 2024:

SELECT service\_record\_id, bike\_id, service\_type\_id, technician\_id, service\_date, comments, total\_cost

FROM Service\_Records

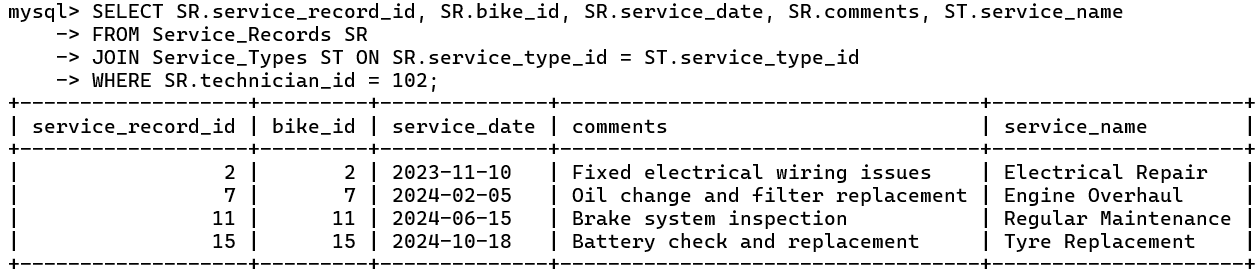
WHERE service\_date > '2024-03-01' LIMIT 5;

-- Find the service records for bikes serviced by technician 102:

SELECT SR.service\_record\_id, SR.bike\_id, SR.service\_date, SR.comments, ST.service\_name

FROM Service\_Records SR

JOIN Service\_Types ST ON SR.service\_type\_id = ST.service\_type\_id

WHERE SR.technician\_id = 102;

Conclusion :-

The **Bike Showroom Database** is a well-rounded schema that provides efficient handling of sales, inventory, customer, service, and financial data. Its design ensures smooth showroom operations, supports data-driven decision-making, and enhances customer satisfaction by organizing key information across different departments. With this schema, a showroom can maintain comprehensive records, streamline operations, and create a customer-focused approach in both sales and after-sales service areas.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\* THE END \*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**T h a n k Y o u**