



DATABASE SYSTEM FOR TAXI SERVICE

DATABSE MANAGEMENT SYSTEM
MUTHUPANDIAN S(2135010017)
ANNAMALAI UNIVERSITY 608 001

REQUIREMENTS

Database Overview: TAXI

This database is designed to manage taxi operations, including taxi information, user details, driver information, trip details, and billing. It consists of five tables:

1. **TAXI:**
 - Stores information about the taxis.
 - Key attributes: Taxi_id, Registration_no, Taxi_Model, Taxi_Year, Taxi_type, Status, Driver_id.
 - Constraints: Taxi_id is the primary key, and Registration_no is unique.
2. **USER_TBL:**
 - Stores information about the users of the taxi service.
 - Key attributes: User_id, user_name, Contact_no, Gender, Address, Taxi_id.
 - Constraints: User_id is the primary key.
3. **DRIVER:**
 - Stores information about the taxi drivers.
 - Key attributes: Driver_id, driver_name, Gender, Contact_no, Rating, Age.
 - Constraints: Driver_id is the primary key.
4. **TRIP_DETAILS:**
 - Stores information about individual taxi trips.
 - Key attributes: Trip_id, Trip_date, Trip_amt, Driver_id, User_id, Taxi_id, Strt_time, End_time.
 - Constraints: Trip_id is the primary key.
5. **BILL_DETAILS:**
 - Stores billing information for each trip.
 - Key attributes: Bill_no, Bill_date, Advance_amt, Discount_amt, Total_amt, User_id, Trip_id.
 - Constraints: Bill_no is the primary key and Trip_id is unique.

Potential Requirements and Use Cases

Based on this schema, here are some potential requirements and use cases:

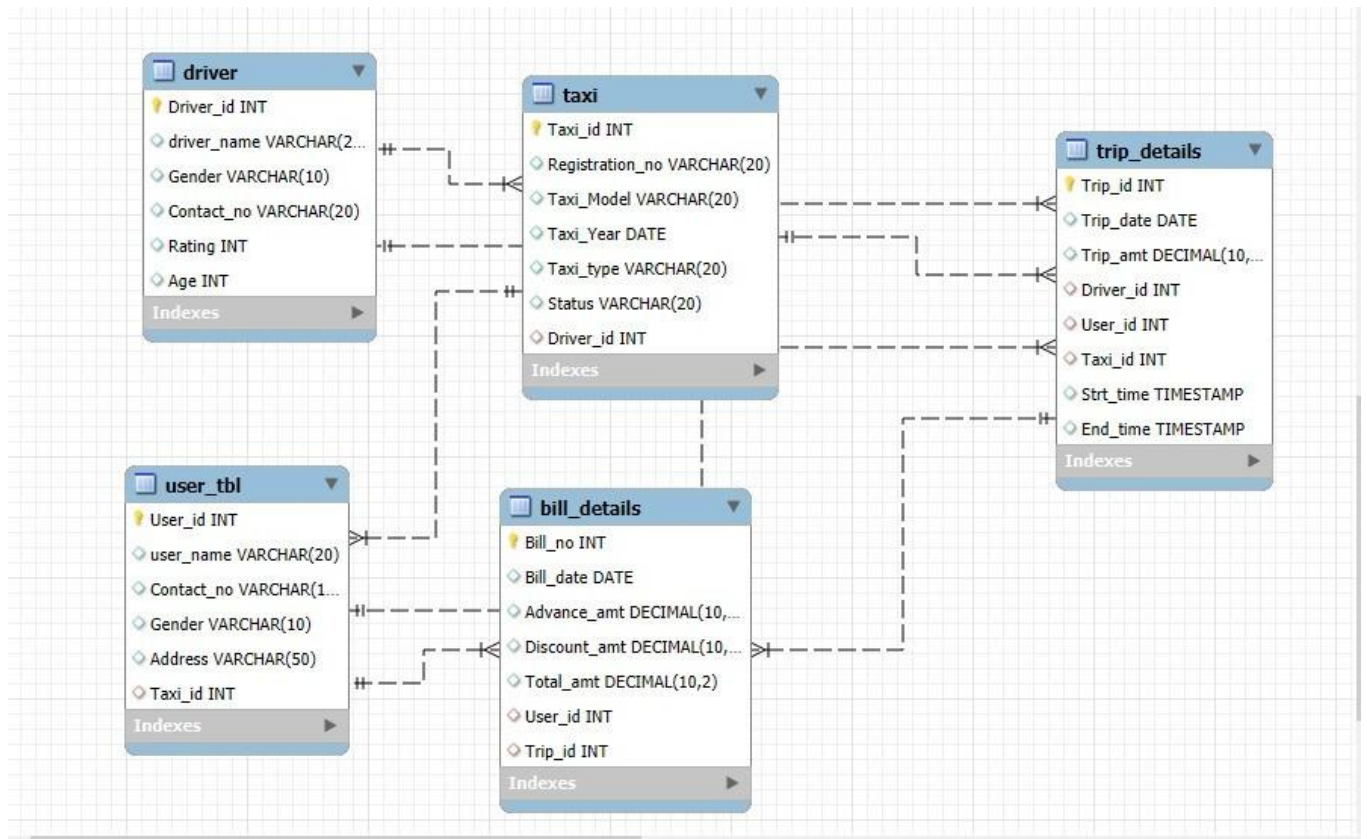
- **Taxi Management:**
 - Adding, updating, and deleting taxi information.
 - Tracking taxi availability (status).
 - Assigning drivers to taxis.
 - Searching taxis by model, type or registration number.
- **User Management:**
 - Registering new users.
 - Updating user information.
 - Searching users.
- **Driver Management:**
 - Adding and updating driver information.
 - Tracking driver ratings and age.
 - Searching drivers.

- **Trip Management:**
 - Recording trip details (date, time, amount, driver, user, taxi).
 - Calculating trip duration and cost.
 - Retrieving trip history for users and drivers.
- **Billing:**
 - Generating bills for trips.
 - Applying discounts and handling advance payments.
 - Calculating total trip cost.
 - Retrieving billing history.
- **Reporting:**
 - Generating reports on taxi usage.
 - Analyzing driver performance.
 - Tracking revenue and expenses.
 - Generating reports on user trip frequency.
- **Relationships:**
 - Linking users to trips.
 - Linking drivers to taxis and trips.
 - Linking Taxis to trips.
 - Linking trips to bills.

Potential Improvements and Considerations

- **Foreign Keys:**
 - It's crucial to add foreign key constraints to enforce relationships between tables. For example:
 - TAXI.Driver_id should reference DRIVER.Driver_id.
 - USER_TBL.Taxi_id should reference TAXI.Taxi_id.
 - TRIP_DETAILS.Driver_id should reference DRIVER.Driver_id.
 - TRIP_DETAILS.User_id should reference USER_TBL.User_id.
 - TRIP_DETAILS.Taxi_id should reference TAXI.Taxi_id.
 - BILL_DETAILS.User_id should reference USER_TBL.User_id.
 - BILL_DETAILS.Trip_id should reference TRIP_DETAILS.Trip_id.
- **Data Types:**
 - Verify the suitability of data types. For example, Contact_no in USER_TBL and DRIVER tables should be checked for consistent length.
 - Consider using DECIMAL instead of FLOAT for monetary values to avoid precision issues.
- **Status Management:**
 - The Status field in the TAXI table could benefit from an enumeration (ENUM) or a separate lookup table for better data integrity.
- **Location Data:**
 - Consider adding location data (start and end locations) to the TRIP_DETAILS table for more detailed trip information.
- **Indexes:**
 - Add indexes to frequently queried columns to improve performance (e.g., User_id, Driver_id, Taxi_id, Trip_date).
- **Time Zones:**
 - When using Timestamps, consider the time zone of the trips, and store that information.

MODELLING OF REQUIREMENTS AS ER-DIAGRAM



SQL STATEMENTS FOR TABLE CREATION

```
-----  
-- Table Creation  
-----
```

```
CREATE TABLE TAXI (  
    Taxi_id integer NOT NULL,  
    Registration_no VARCHAR(20),  
    Taxi_Model VARCHAR(20),  
    Taxi_Year DATE,  
    Taxi_type VARCHAR(20),  
    Status VARCHAR(20),  
    Driver_id integer,  
    PRIMARY KEY (Taxi_id),  
    UNIQUE (Registration_no)  
);
```

```
CREATE TABLE USER_TBL (  
    User_id int NOT NULL,  
    user_name VARCHAR(20),  
    Contact_no varchar(10),  
    Gender VARCHAR(10),  
    Address VARCHAR(50),  
    Taxi_id int,  
    PRIMARY KEY (User_id)  
);
```

```
CREATE TABLE DRIVER (  
    Driver_id int NOT NULL,  
    driver_name VARCHAR(20),  
    Gender VARCHAR(10),  
    Contact_no VARCHAR(20),  
    Rating int,  
    Age int,  
    PRIMARY KEY (Driver_id)  
);
```

```
CREATE TABLE TRIP_DETAILS (  
    Trip_id int NOT NULL,  
    Trip_date DATE,  
    Trip_amt decimal(10,2),  
    Driver_id int,  
    User_id int,  
    Taxi_id int,  
    Strt_time TIMESTAMP,  
    End_time TIMESTAMP,  
    PRIMARY KEY (Trip_id)  
);
```

```

CREATE TABLE BILL_DETAILS (
    Bill_no int NOT NULL,
    Bill_date DATE,
    Advance_amt decimal(10,2),
    Discount_amt decimal(10,2),
    Total_amt decimal(10,2),
    User_id int,
    Trip_id int,
    PRIMARY KEY (Bill_no),
    UNIQUE (Trip_id)
);

```

SQL STATEMENTS FOR FOREIGN KEY CREATION

```

-----
-- Foreign key creation
-----
ALTER TABLE TAXI ADD CONSTRAINT fketadr FOREIGN KEY (Driver_id) REFERENCES
DRIVER(Driver_id) ON DELETE CASCADE;
ALTER TABLE USER_TBL ADD CONSTRAINT fkusta FOREIGN KEY (Taxi_id) REFERENCES
TAXI(Taxi_id) ON DELETE CASCADE;
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fktddr FOREIGN KEY (Driver_id)
REFERENCES DRIVER(Driver_id) ON DELETE CASCADE;
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fktdusr FOREIGN KEY (User_id)
REFERENCES USER_TBL(User_id) ON DELETE CASCADE;
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fktdtax FOREIGN KEY (Taxi_id)
REFERENCES TAXI(Taxi_id) ON DELETE CASCADE;
ALTER TABLE BILL_DETAILS ADD CONSTRAINT fkbtdr FOREIGN KEY (Trip_id)
REFERENCES TRIP_DETAILS(Trip_id) ON DELETE CASCADE;
ALTER TABLE BILL_DETAILS ADD CONSTRAINT fkbdsr FOREIGN KEY (User_id)
REFERENCES USER_TBL(User_id) ON DELETE CASCADE;

```

SQL STATEMENTS FOR INSERT COMMANDS

```

-----
-- Insert Commands
-----

INSERT INTO TAXI (Taxi_id, Registration_no, Taxi_Model, Taxi_Year,
Taxi_type, Status, Driver_id) VALUES
(1, 'KA01AA1111', 'Toyota Camry', '2022-01-15', 'Sedan', 'Available', 101),
(2, 'MH02BB2222', 'Honda Civic', '2021-05-20', 'Sedan', 'Unavailable', 102),
(3, 'DL03CC3333', 'Hyundai i20', '2023-03-10', 'Hatchback', 'Available',
103),
(4, 'TN04DD4444', 'Maruti Swift', '2020-11-01', 'Hatchback', 'Available',
104),
(5, 'KL05EE5555', 'Tata Nexon', '2022-08-25', 'SUV', 'Unavailable', 105),
(6, 'AP06FF6666', 'Mahindra XUV700', '2023-06-12', 'SUV', 'Available', 106),
(7, 'GJ07GG7777', 'Renault Kwid', '2021-02-18', 'Hatchback', 'Available',
107),
(8, 'RJ08HH8888', 'Kia Seltos', '2022-12-05', 'SUV', 'Unavailable', 108),
(9, 'PB09II9999', 'Volkswagen Polo', '2020-09-30', 'Hatchback', 'Available',
109),
(10, 'HR10JJ0000', 'MG Hector', '2023-04-17', 'SUV', 'Available', 110),
(11, 'KA11KK1112', 'Toyota Innova', '2021-07-02', 'MPV', 'Unavailable',
111),
(12, 'MH12LL2223', 'Maruti Ertiga', '2022-03-08', 'MPV', 'Available', 112),
(13, 'DL13MM3334', 'Honda City', '2020-12-22', 'Sedan', 'Available', 113),
(14, 'TN14NN4445', 'Hyundai Verna', '2023-01-05', 'Sedan', 'Unavailable',
114),
(15, 'KL15OO5556', 'Tata Altroz', '2021-09-11', 'Hatchback', 'Available',
115),
(16, 'AP16PP6667', 'Mahindra Thar', '2022-06-28', 'SUV', 'Available', 116),
(17, 'GJ17QQ7778', 'Renault Triber', '2020-10-15', 'MPV', 'Unavailable',
117),
(18, 'RJ18RR8889', 'Kia Sonet', '2023-02-20', 'SUV', 'Available', 118),

```

TAXI SERVICE SYSTEM

```
(19, 'PB19SS9990', 'Volkswagen Virtus', '2021-11-08', 'Sedan', 'Available',  
119),  
(20, 'HR20TT0001', 'MG Astor', '2022-09-03', 'SUV', 'Unavailable', 120),  
(21, 'KA21UU1113', 'Nissan Magnite', '2021-04-01', 'SUV', 'Available', 121),  
(22, 'MH22VV2224', 'Ford EcoSport', '2020-07-15', 'SUV', 'Unavailable',  
122),  
(23, 'DL23WW3335', 'Skoda Kushaq', '2022-02-28', 'SUV', 'Available', 123),  
(24, 'TN24XX4446', 'Jeep Compass', '2023-05-10', 'SUV', 'Unavailable', 124),  
(25, 'KL25YY5557', 'Citroen C5 Aircross', '2021-12-03', 'SUV', 'Available',  
125),  
(26, 'AP26ZZ6668', 'BMW X1', '2023-08-20', 'SUV', 'Unavailable', 126),  
(27, 'GJ27AA7779', 'Audi Q3', '2022-04-25', 'SUV', 'Available', 127),  
(28, 'RJ28BB8890', 'Mercedes GLA', '2021-06-18', 'SUV', 'Unavailable', 128),  
(29, 'PB29CC9991', 'Volvo XC40', '2023-01-08', 'SUV', 'Available', 129),  
(30, 'HR30DD0002', 'shift', '2022-10-12', 'SUV', 'Unavailable', 130),  
(31, 'KA31EE1114', 'Toyota Fortuner', '2020-05-05', 'SUV', 'Available',  
131),  
(32, 'MH32FF2225', 'Mahindra Scorpio', '2021-08-30', 'SUV', 'Unavailable',  
132),  
(33, 'DL33GG3336', 'Tata Safari', '2022-03-15', 'SUV', 'Available', 133),  
(34, 'TN34HH4447', 'Hyundai Alcazar', '2023-07-02', 'SUV', 'Unavailable',  
134),  
(35, 'KL35II5558', 'MG Gloster', '2021-11-20', 'SUV', 'Available', 135),  
(36, 'AP36JJ6669', 'Ford Endeavour', '2020-09-10', 'SUV', 'Unavailable',  
136),  
(37, 'GJ37KK7780', 'Skoda Kodiahq', '2022-06-05', 'SUV', 'Available', 137),  
(38, 'RJ38LL8891', 'Jeep Meridian', '2023-03-25', 'SUV', 'Unavailable',  
138),  
(39, 'PB39MM9992', 'Citroen C3 Aircross', '2021-10-18', 'SUV', 'Available',  
139),  
(40, 'HR40NN0003', 'BMW X3', '2023-09-01', 'SUV', 'Unavailable', 140);
```

```
INSERT INTO USER_TBL (User_id, user_name, Contact_no, Gender, Address,  
Taxi_id) VALUES
```

```
(201, 'Alice Smith', 9876543210, 'Female', '123 Main St, City A', 1),  
(202, 'Bob Johnson', 8765432109, 'Male', '456 Oak Ave, City B', 2),  
(203, 'Charlie Brown', 7654321098, 'Male', '789 Pine Ln, City C', 3),  
(204, 'David Lee', 6543210987, 'Male', '101 Elm Rd, City D', 4),  
(205, 'Eve Wilson', 5432109876, 'Female', '202 Maple Dr, City E', 5),  
(206, 'Frank Miller', 4321098765, 'Male', '303 Birch Ct, City F', 6),  
(207, 'Grace Davis', 3210987654, 'Female', '404 Willow Way, City G', 7),  
(208, 'Henry Garcia', 2109876543, 'Male', '505 Cedar Pl, City H', 8),  
(209, 'Ivy Rodriguez', 1098765432, 'Female', '606 Spruce St, City I', 9),  
(210, 'Jack Martinez', 9876543211, 'Male', '707 Redwood Ave, City J', 10),  
(211, 'Karen Anderson', 8765432110, 'Female', '808 Cherry Ln, City K', 11),  
(212, 'Larry Thomas', 7654321109, 'Male', '909 Walnut Rd, City L', 12),  
(213, 'Mary Jackson', 6543211098, 'Female', '1010 Pine Dr, City M', 13),  
(214, 'Nancy White', 5432110987, 'Female', '1111 Oak Ct, City N', 14),  
(215, 'Oscar Harris', 4321109876, 'Male', '1212 Elm Way, City O', 15),
```


TAXI SERVICE SYSTEM

```
(216, 'Patricia Martin', 3211098765, 'Female', '1313 Maple Pl, City P', 16),
(217, 'Quentin Thompson', 2110987654, 'Male', '1414 Birch St, City Q', 17),
(218, 'Rachel Garcia', 1109876543, 'Female', '1515 Cedar Ave, City R', 18),
(219, 'Samuel Perez', 9876543212, 'Male', '1616 Spruce Ln, City S', 19),
(220, 'Tina Wilson', 8765432111, 'Female', '1717 Redwood Rd, City T', 20),
(221, 'Uma Patel', 7654321110, 'Female', '1818 Lime Rd, City U', 21),
(222, 'Victor Singh', 6543211109, 'Male', '1919 Mango Ave, City V', 22),
(223, 'Wendy Kumar', 5432111098, 'Female', '2020 Kiwi Ln, City W', 23),
(224, 'Xavier Reddy', 4321110987, 'Male', '2121 Orange Ct, City X', 24),
(225, 'Yvonne Sharma', 3211109876, 'Female', '2222 Grape Way, City Y', 25),
(226, 'Zachary Gupta', 2111098765, 'Male', '2323 Berry Pl, City Z', 26),
(227, 'Ava Desai', 1110987654, 'Female', '2424 Apple St, City AA', 27),
(228, 'Ben Banerjee', 9876543213, 'Male', '2525 Cherry Ave, City BB', 28),
(229, 'Clara Joshi', 8765432112, 'Female', '2626 Plum Ln, City CC', 29),
(230, 'Daniel Kapoor', 7654321111, 'Male', '2727 Peach Rd, City DD', 30),
(231, 'Ella Khan', 6543211110, 'Female', '2828 Pear Dr, City EE', 31),
(232, 'Felix Verma', 5432111099, 'Male', '2929 Lemon Ct, City FF', 32),
(233, 'Gina Mehra', 4321110988, 'Female', '3030 Lime Way, City GG', 33),
(234, 'Harry Rao', 3211109877, 'Male', '3131 Mango Pl, City HH', 34),
(235, 'Isabella Singh', 2111098766, 'Female', '3232 Kiwi St, City II', 35),
(236, 'Jacob Kumar', 1110987655, 'Male', '3333 Orange Ave, City JJ', 36),
(237, 'Katie Reddy', 9876543214, 'Female', '3434 Grape Ln, City KK', 37),
(238, 'Liam Sharma', 8765432113, 'Male', '3535 Berry Rd, City LL', 38),
(239, 'Mia Gupta', 7654321112, 'Female', '3636 Apple Dr, City MM', 39),
(240, 'Noah Desai', 6543211111, 'Male', '3737 Cherry Ct, City NN', 40);
```

```
INSERT INTO DRIVER (Driver_id, driver_name, Gender, Contact_no, Rating, Age)
VALUES
```

```
(101, 'John Doe', 'Male', '9876500001', 4, 35),
(102, 'Jane Smith', 'Female', '8765400002', 5, 42),
(103, 'Mike Brown', 'Male', '7654300003', 3, 28),
(104, 'Sarah Lee', 'Female', '6543200004', 4, 31),
(105, 'David Kim', 'Male', '5432100005', 5, 45),
(106, 'Emily Chen', 'Female', '4321000006', 4, 38),
(107, 'Kevin Wang', 'Male', '3210900007', 3, 29),
(108, 'Lisa Liu', 'Female', '2109800008', 5, 40),
(109, 'Brian Yang', 'Male', '1098700009', 4, 33),
(110, 'Amy Xu', 'Female', '9876500010', 4, 36),
(111, 'Chris Zhou', 'Male', '8765400011', 5, 41),
(112, 'Grace Wu', 'Female', '7654300012', 3, 27),
(113, 'Henry Zheng', 'Male', '6543200013', 4, 32),
(114, 'Isabel Lin', 'Female', '5432100014', 5, 44),
(115, 'Jack Feng', 'Male', '4321000015', 4, 39),
(116, 'Kelly Cai', 'Female', '3210900016', 3, 30),
(117, 'Leo Pan', 'Male', '2109800017', 5, 43),
(118, 'Mia Guo', 'Female', '1098700018', 4, 37),
(119, 'Nathan Deng', 'Male', '9876500019', 4, 34),
(120, 'Olivia He', 'Female', '8765400020', 5, 46),
(121, 'Peter Rao', 'Male', '7654300021', 4, 38),
```

TAXI SERVICE SYSTEM

```
(122, 'Rachel Shah', 'Female', '6543200022', 5, 41),
(123, 'Simon Patel', 'Male', '5432100023', 3, 29),
(124, 'Tina Iyer', 'Female', '4321000024', 4, 33),
(125, 'Victor Menon', 'Male', '3210900025', 5, 45),
(126, 'Wendy Nair', 'Female', '2109800026', 4, 36),
(127, 'Xavier Pillai', 'Male', '1098700027', 3, 30),
(128, 'Yasmine Thomas', 'Female', '9876500028', 5, 42),
(129, 'Zachary George', 'Male', '8765400029', 4, 37),
(130, 'Alice Abraham', 'Female', '7654300030', 4, 39),
(131, 'Ben Mathews', 'Male', '6543200031', 5, 43),
(132, 'Clara Jacob', 'Female', '5432100032', 3, 28),
(133, 'Daniel Joseph', 'Male', '4321000033', 4, 31),
(134, 'Ella Alex', 'Female', '3210900034', 5, 46),
(135, 'Felix John', 'Male', '2109800035', 4, 35),
(136, 'Gina Paul', 'Female', '1098700036', 3, 32),
(137, 'Harry Roy', 'Male', '9876500037', 5, 40),
(138, 'Isabella Samuel', 'Female', '8765400038', 4, 34),
(139, 'Jacob David', 'Male', '7654300039', 4, 44),
(140, 'Katie Wilson', 'Female', '6543200040', 5, 41);
```

```
INSERT INTO TRIP_DETAILS (Trip_id, Trip_date, Trip_amt, Driver_id, User_id,
Taxi_id, Strt_time, End_time) VALUES
(301, '2023-10-26', 25.50, 101, 201, 1, '2023-10-26 08:00:00', '2023-10-26
08:30:00'),
(302, '2023-10-26', 30.75, 102, 202, 2, '2023-10-26 09:00:00', '2023-10-26
09:45:00'),
(303, '2023-10-26', 15.20, 103, 203, 3, '2023-10-26 10:00:00', '2023-10-26
10:20:00'),
(304, '2023-10-27', 40.00, 104, 204, 4, '2023-10-27 11:00:00', '2023-10-27
12:00:00'),
(305, '2023-10-27', 22.80, 105, 205, 5, '2023-10-27 12:30:00', '2023-10-27
13:15:00'),
(306, '2023-10-27', 35.60, 106, 206, 6, '2023-10-27 13:30:00', '2023-10-27
14:20:00'),
(307, '2023-10-28', 18.90, 107, 207, 7, '2023-10-28 14:30:00', '2023-10-28
15:00:00'),
(308, '2023-10-28', 50.25, 108, 208, 8, '2023-10-28 15:30:00', '2023-10-28
16:45:00'),
(309, '2023-10-28', 28.30, 109, 209, 9, '2023-10-28 16:30:00', '2023-10-28
17:10:00'),
(310, '2023-10-29', 45.90, 110, 210, 10, '2023-10-29 17:00:00', '2023-10-29
18:00:00'),
(311, '2023-10-29', 21.15, 111, 211, 11, '2023-10-29 18:30:00', '2023-10-29
19:15:00'),
(312, '2023-10-29', 32.40, 112, 212, 12, '2023-10-29 19:30:00', '2023-10-29
20:20:00'),
(313, '2023-10-30', 20.50, 113, 213, 13, '2023-10-30 20:30:00', '2023-10-30
21:00:00'),
```

TAXI SERVICE SYSTEM

(314, '2023-10-30', 48.75, 114, 214, 14, '2023-10-30 21:30:00', '2023-10-30 22:45:00'),
(315, '2023-10-30', 27.80, 115, 215, 15, '2023-10-30 22:30:00', '2023-10-30 23:10:00'),
(316, '2023-10-31', 42.60, 116, 216, 16, '2023-10-31 23:00:00', '2023-10-31 00:00:00'),
(317, '2023-10-31', 23.90, 117, 217, 17, '2023-10-31 00:30:00', '2023-10-31 01:15:00'),
(318, '2023-10-31', 38.20, 118, 218, 18, '2023-10-31 01:30:00', '2023-10-31 02:20:00'),
(319, '2023-11-01', 19.70, 119, 219, 19, '2023-11-01 02:30:00', '2023-11-01 03:00:00'),
(320, '2023-11-01', 52.50, 120, 220, 20, '2023-11-01 03:30:00', '2023-11-01 04:45:00'),
(321, '2023-11-02', 26.80, 121, 221, 21, '2023-11-02 05:00:00', '2023-11-02 05:40:00'),
(322, '2023-11-02', 33.15, 122, 222, 22, '2023-11-02 06:00:00', '2023-11-02 06:50:00'),
(323, '2023-11-02', 17.60, 123, 223, 23, '2023-11-02 07:00:00', '2023-11-02 07:25:00'),
(324, '2023-11-03', 44.20, 124, 224, 24, '2023-11-03 08:00:00', '2023-11-03 09:10:00'),
(325, '2023-11-03', 24.30, 125, 225, 25, '2023-11-03 09:30:00', '2023-11-03 10:20:00'),
(326, '2023-11-03', 39.70, 126, 226, 26, '2023-11-03 10:30:00', '2023-11-03 11:30:00'),
(327, '2023-11-04', 21.50, 127, 227, 27, '2023-11-04 11:00:00', '2023-11-04 11:35:00'),
(328, '2023-11-04', 54.80, 128, 228, 28, '2023-11-04 12:00:00', '2023-11-04 13:15:00'),
(329, '2023-11-04', 30.20, 129, 229, 29, '2023-11-04 13:30:00', '2023-11-04 14:10:00'),
(330, '2023-11-05', 47.30, 130, 230, 30, '2023-11-05 14:00:00', '2023-11-05 15:00:00'),
(331, '2023-11-05', 22.75, 131, 231, 31, '2023-11-05 15:30:00', '2023-11-05 16:15:00'),
(332, '2023-11-05', 34.90, 132, 232, 32, '2023-11-05 16:30:00', '2023-11-05 17:20:00'),
(333, '2023-11-06', 22.10, 133, 233, 33, '2023-11-06 17:00:00', '2023-11-06 17:30:00'),
(334, '2023-11-06', 51.65, 134, 234, 34, '2023-11-06 17:30:00', '2023-11-06 18:45:00'),
(335, '2023-11-06', 29.50, 135, 235, 35, '2023-11-06 18:30:00', '2023-11-06 19:10:00'),
(336, '2023-11-07', 43.80, 136, 236, 36, '2023-11-07 19:00:00', '2023-11-07 20:00:00'),
(337, '2023-11-07', 25.15, 137, 237, 37, '2023-11-07 20:30:00', '2023-11-07 21:15:00'),

TAXI SERVICE SYSTEM

```
(338, '2023-11-07', 40.30, 138, 238, 38, '2023-11-07 21:30:00', '2023-11-07  
22:20:00'),  
(339, '2023-11-08', 20.90, 139, 239, 39, '2023-11-08 22:00:00', '2023-11-08  
22:30:00'),  
(340, '2023-11-08', 53.95, 140, 240, 40, '2023-11-08 22:30:00', '2023-11-08  
23:45:00');
```

```
INSERT INTO BILL_DETAILS (Bill_no, Bill_date, Advance_amt, Discount_amt,  
Total_amt, User_id, Trip_id) VALUES
```

```
(401, '2023-10-26', 10.00, 2.55, 25.50, 201, 301),  
(402, '2023-10-26', 15.00, 3.08, 30.75, 202, 302),  
(403, '2023-10-26', 5.00, 1.52, 15.20, 203, 303),  
(404, '2023-10-27', 20.00, 4.00, 40.00, 204, 304),  
(405, '2023-10-27', 10.00, 2.28, 22.80, 205, 305),  
(406, '2023-10-27', 18.00, 3.56, 35.60, 206, 306),  
(407, '2023-10-28', 8.00, 1.89, 18.90, 207, 307),  
(408, '2023-10-28', 25.00, 5.03, 50.25, 208, 308),  
(409, '2023-10-28', 12.00, 2.83, 28.30, 209, 309),  
(410, '2023-10-29', 22.00, 4.59, 45.90, 210, 310),  
(411, '2023-10-29', 9.00, 2.12, 21.15, 211, 311),  
(412, '2023-10-29', 16.00, 3.24, 32.40, 212, 312),  
(413, '2023-10-30', 9.00, 2.05, 20.50, 213, 313),  
(414, '2023-10-30', 24.00, 4.88, 48.75, 214, 314),  
(415, '2023-10-30', 13.00, 2.78, 27.80, 215, 315),  
(416, '2023-10-31', 21.00, 4.26, 42.60, 216, 316),  
(417, '2023-10-31', 11.00, 2.39, 23.90, 217, 317),  
(418, '2023-10-31', 19.00, 3.82, 38.20, 218, 318),  
(419, '2023-11-01', 9.00, 1.97, 19.70, 219, 319),  
(420, '2023-11-01', 26.00, 5.25, 52.50, 220, 320),  
(421, '2023-11-02', 14.00, 2.68, 26.80, 221, 321),  
(422, '2023-11-02', 17.00, 3.32, 33.15, 222, 322),  
(423, '2023-11-02', 7.00, 1.76, 17.60, 223, 323),  
(424, '2023-11-03', 22.00, 4.42, 44.20, 224, 324),  
(425, '2023-11-03', 12.00, 2.43, 24.30, 225, 325),  
(426, '2023-11-03', 20.00, 3.97, 39.70, 226, 326),  
(427, '2023-11-04', 10.00, 2.15, 21.50, 227, 327),  
(428, '2023-11-04', 27.00, 5.48, 54.80, 228, 328),  
(429, '2023-11-04', 15.00, 3.02, 30.20, 229, 329),  
(430, '2023-11-05', 23.00, 4.73, 47.30, 230, 330),  
(431, '2023-11-05', 11.00, 2.28, 22.75, 231, 331),  
(432, '2023-11-05', 17.00, 3.49, 34.90, 232, 332),  
(433, '2023-11-06', 10.00, 2.21, 22.10, 233, 333),  
(434, '2023-11-06', 26.00, 5.17, 51.65, 234, 334),  
(435, '2023-11-06', 14.00, 2.95, 29.50, 235, 335),  
(436, '2023-11-07', 22.00, 4.38, 43.80, 236, 336),  
(437, '2023-11-07', 12.00, 2.52, 25.15, 237, 337),  
(438, '2023-11-07', 20.00, 4.03, 40.30, 238, 338),  
(439, '2023-11-08', 10.00, 2.09, 20.90, 239, 339),  
(440, '2023-11-08', 27.00, 5.40, 53.95, 240, 340);
```

PL/SQL – PROCEDURES

-- VIEW

-- Find the names and contact numbers of all drivers.

```
CREATE VIEW DRV_DETAILS AS
SELECT driver_name, Contact_no
FROM DRIVER;
```

```
select*from drv_details;
```

-- List all taxis of type 'Sedan'.

```
CREATE VIEW TAXI_TYP AS
SELECT *
FROM TAXI
WHERE Taxi_type = 'Sedan';
```

```
select*from taxi_typ;
```

-- Find the average rating of all drivers.

```
CREATE VIEW MAX_RATE AS
SELECT MAX(Rating)
FROM DRIVER;
```

```
select*from max_rate;
```

-- Count the number of trips.

```
CREATE VIEW COUNT_TRIP AS
SELECT COUNT(*)
FROM TRIP_DETAILS;
```

```
select*from count_trip;
```

-- List drivers and the number of trips they have taken.

```
CREATE VIEW DRV_TRIP_DETAILS AS
SELECT d.driver_name, COUNT(t.Trip_id) AS NumberOfTrips
FROM DRIVER d
LEFT JOIN TRIP_DETAILS t ON d.Driver_id = t.Driver_id
GROUP BY d.driver_name;
```

```
select*from drv_trip_details;
```

```
-- List users and the number of trips they have taken.
```

```
CREATE VIEW USR_TRIP_DETAILS AS
SELECT u.user_name, COUNT(t.Trip_id) AS NumberOfTrips
FROM USER_TBL u
LEFT JOIN TRIP_DETAILS t ON u.User_id = t.User_id
GROUP BY u.user_name;
```

```
select*from usr_trip_details;
```

```
-- List taxis and the number of trips they have been used for.
```

```
CREATE VIEW TAXI_TRIPS_DETAILS AS
SELECT t.Taxi_Model, COUNT(td.Trip_id) AS NumberOfTrips
FROM TAXI t
LEFT JOIN TRIP_DETAILS td ON t.Taxi_id = td.Taxi_id
GROUP BY t.Taxi_Model;
```

```
select*from taxi_trips_details;
```

```
-- Find the user who has taken the most trips.
```

```
CREATE VIEW MOST_TRIPS AS
SELECT u.*
FROM USER_TBL u
WHERE u.User_id = (SELECT User_id FROM TRIP_DETAILS GROUP BY User_id ORDER BY
COUNT(*) DESC LIMIT 1);
```

```
select*from most_trips;
```

```
-- Creating a View for Trip Summary
```

```
CREATE VIEW Trip_Summary AS
SELECT
    T.Trip_id,
    T.Trip_date,
    T.Trip_amt,
    U.user_name AS UserName,
    D.driver_name AS DriverName,
    X.Registration_no AS TaxiNumber,
    T.Strt_time,
    T.End_time
FROM TRIP_DETAILS AS T
```

```
JOIN USER_TBL U ON T.User_id = U.User_id
JOIN DRIVER D ON T.Driver_id = D.Driver_id
JOIN TAXI X ON T.Taxi_id = X.Taxi_id;

SELECT * FROM Trip_Summary;

-- STORE PROCEDURE

-- Procedure to get trip details for a specific date

DELIMITER //

CREATE PROCEDURE GetTripsByDate(IN inputDate DATE)
BEGIN
    select*from trip_details where Trip_date = inputDate;
END //

DELIMITER ;

CALL GetTripsByDate('2023-10-26');

-- Procedure to get taxi details for a specific year

DELIMITER //

CREATE PROCEDURE GetTaxisByYear(IN inputYear DATE)
BEGIN
    select*from taxi where taxi_year=inputyear;
END //
DELIMITER ;

CALL GetTaxisByYear('2022-01-15');

-- Procedure to get user details for a specific taxi_id

DELIMITER //

CREATE PROCEDURE GetUsersByTaxiID(IN taxiID INT)
BEGIN
    select*from user_tbl where taxi_id=taxi_id;
END //
```

```
DELIMITER ;
```

```
CALL GetUsersByTaxiID(123);
```

```
-- Procedure to get driver details for a specific rating
```

```
DELIMITER //
```

```
CREATE PROCEDURE GetDriversByRating(IN driverRating INT)
```

```
BEGIN
```

```
select*from driver where rating=driverrating;
```

```
END //
```

```
DELIMITER ;
```

```
CALL GetDriversByRating(4);
```

```
-- Store procedure to get user details by gender.
```

```
DELIMITER //
```

```
CREATE PROCEDURE GetUserDetailsByGender(IN gender VARCHAR(10))
```

```
BEGIN
```

```
SELECT
```

```
    User_id,
```

```
    user_name,
```

```
    Contact_no,
```

```
    Gender,
```

```
    Address,
```

```
    Taxi_id
```

```
FROM
```

```
    USER_TBL
```

```
WHERE
```

```
    Gender = gender;
```

```
END //
```

```
DELIMITER ;
```

```
CALL GetUserDetailsByGender('Male');
```


-- show the user name and the bill date.

DELIMITER //

```
CREATE PROCEDURE Get_User_Bill_Details(IN p_User_id INT)
BEGIN
    SELECT U.user_name, B.Bill_date
    FROM USER_TBL AS U
    JOIN BILL_DETAILS AS B ON U.User_id = B.User_id
    WHERE U.User_id = B.User_id;
END //
```

DELIMITER ;

CALL Get_User_Bill_Details(10);

-- show the user name and the bill date.

DELIMITER //

```
CREATE PROCEDURE Get_Unused_Taxis(IN p_Taxi_Type VARCHAR(20))
BEGIN
    SELECT T.Taxi_id, T.Registration_no, T.Taxi_Model, T.Status
    FROM TAXI as T
    LEFT JOIN TRIP_DETAILS as TD ON T.Taxi_id = TD.Taxi_id
    WHERE TD.Taxi_id IS NULL
    AND T.Taxi_type = p_Taxi_Type;
END //
```

DELIMITER ;

CALL Get_Unused_Taxis('MPV');

-- show the user name and the bill date.

DELIMITER //

```
CREATE PROCEDURE Get_Popular_Taxi_Models(IN A INT)
BEGIN
    SELECT T.Taxi_Model, COUNT(DISTINCT TD.User_id) AS User_Count
    FROM TAXI as T
```

```

    JOIN TRIP_DETAILS as TD ON T.Taxi_id = TD.Taxi_id
    GROUP BY T.Taxi_Model
    HAVING COUNT(DISTINCT TD.User_id)= A;
END //

```

```

DELIMITER ;

```

```

CALL Get_Popular_Taxi_Models(7);

```

```

DELIMITER //

```

```

CREATE PROCEDURE Count_Taxi_Trips(
    IN p_Taxi_id INT,
    OUT p_Trip_Count INT
)
BEGIN
    SELECT COALESCE(COUNT(Trip_id), 0)
    INTO p_Trip_Count
    FROM TRIP_DETAILS
    WHERE Taxi_id = p_Taxi_id;
END //

```

```

DELIMITER ;
CALL Count_Taxi_Trips(201, @Trip_Count);
SELECT @Trip_Count;

```

```

-- Calculate Driver's Earnings;

```

```

DELIMITER //

```

```

CREATE PROCEDURE Calculate_Driver_Earnings(
    IN p_Driver_id INT,
    IN p_Year INT,
    IN p_Month INT,
    OUT p_Total_Earnings DECIMAL(10,2)
)
BEGIN
    SELECT COALESCE(SUM(Trip_amt), 0)
    INTO p_Total_Earnings
    FROM TRIP_DETAILS
    WHERE Driver_id = p_Driver_id

```

```
AND YEAR(Trip_date) = p_Year  
AND MONTH(Trip_date) = p_Month;  
END //
```

```
DELIMITER ;
```

```
CALL Calculate_Driver_Earnings(101, 2024, 3, @Total_Earnings);  
SELECT @Total_Earnings;
```