Task: Payment Gateway System

Schema Design

Design a **payment gateway system** that includes the following entities:

1. **Users** – Customers who make payments.
2. **Merchants** – Businesses that receive payments.
3. **Transactions** – Payment details for each transaction.
4. **Payment Methods** – Different modes of payment (Credit Card, UPI, Net Banking, etc.).
5. **Refunds** – Handling of failed or refunded transactions.
6. **Audit Logs** – Tracking payment activities.

Table Schema

1. Users Table

```
CREATE TABLE users (
   user_id SERIAL PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   email VARCHAR(150) UNIQUE NOT NULL,
   phone VARCHAR(15) UNIQUE NOT NULL,
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2. Merchants Table

```
CREATE TABLE merchants (
   merchant_id SERIAL PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   business_email VARCHAR(150) UNIQUE NOT NULL,
   business_phone VARCHAR(15) UNIQUE NOT NULL,
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

3. Payment Methods Table

```
CREATE TABLE payment_methods (
    method_id SERIAL PRIMARY KEY,
    method_name VARCHAR(50) UNIQUE NOT NULL CHECK (method_name IN ('Credit Card', 'Debit Card', 'UPI', 'Net Banking', 'Wallet')),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

4. Transactions Table

```
CREATE TABLE transactions (
    transaction_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(user_id),
    merchant_id INT REFERENCES merchants(merchant_id),
    amount DECIMAL(10,2) NOT NULL,
    currency VARCHAR(10) DEFAULT 'INR',
    method_id INT REFERENCES payment_methods(method_id),
    status VARCHAR(20) CHECK (status IN ('Pending', 'Success', 'Failed', 'Refunded')),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

5. Refunds Table

```
CREATE TABLE refunds (
    refund_id SERIAL PRIMARY KEY,
    transaction_id INT REFERENCES transactions(transaction_id) UNIQUE,
    refund_amount DECIMAL(10,2) NOT NULL,
    refund_status VARCHAR(20) CHECK (refund_status IN ('Initiated', 'Processed', 'Failed')),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

6. Audit Logs Table

```
CREATE TABLE audit_logs (
   log_id SERIAL PRIMARY KEY,
   transaction_id INT REFERENCES transactions(transaction_id),
   action VARCHAR(50) NOT NULL CHECK (action IN ('Payment Initiated', 'Payment Success', 'Payment Failed', 'Refund Initiated', 'Refund Processed')),
   log_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

SQL Questions

1. **Retrieve all transactions made by a specific user.**
2. **Find all failed transactions for a specific merchant.**
3. **List all available payment methods.**
4. **Get the total number of transactions per user.**
5. **Find the total amount spent by a specific user.**
6. **Retrieve transaction details along with user details.**
7. **List all transactions with merchant names.**
8. **Get transactions that have refunds.**
9. **List all transactions including those without refunds.**
10. **Get all merchants including those without transactions.**
11. **Find total revenue per merchant.**
12. **Find users who have never made a transaction.**
13. **Retrieve the top 5 merchants with the highest transaction amounts.**
14. **Find users who have made at least one refund.**
15. **Get the most frequently used payment method.**
16. **Find the transaction with the highest amount.**
17. **Find users who spent more than the average transaction amount.**
18. **Rank users based on total spending.**
19. **Get the last 5 transactions for a user.**

20. Find transactions higher than the average amount for that merchant.

21. Find the percentage of successful transactions per merchant.

22. Get a rolling sum of total transaction amounts per user.

23. Find duplicate transactions.

24. Find the last transaction for each user.

25. Find users with payments but no successful transactions.

26. Detect fraudulent transactions occurring within 10 minutes.

27. Analyze daily revenue trends.

28. Find transactions refunded within 24 hours.

29. Calculate monthly transaction statistics.

30. Find the top 3 merchants for each month.