ADBMS
SQL Operations

Scenario 1: Library

Schema:-

- BOOKS: (Book_ID, Title, Author, Genre, Price, Publication_Year, Copies)
- BORROWERS: (Borrower_ID, Name, Address, Phone, Membership_Type)
- ISSUES: (Issue_ID, Borrower_ID, Book_ID, Issue_Date, Return_Date)

Easy Questions (10)

1. Create a table BOOKS with the given schema
2. Insert at least 5 rows into the BOOKS table.
3. Display all the details of books available in the library.
4. Display the list of books published after 2015.
5. Create a table BORROWERS with the given schema.
6. Insert at least 5 rows into the BORROWERS table.
7. Display the names and phone numbers of all borrowers.
8. Display the list of borrowers who have a "Gold" membership type.
9. Create a table ISSUES with the given schema.
10. Insert 5 records into the ISSUES table.

Medium Questions :

1. Display the title and author of all books priced above 500.
2. Update the price of all books in the "Fiction" genre by increasing it by 10%.
3. Delete the records of books that have no copies left.
4. Delete the records of books that have no copies left
5. Create a view AVAILABLE_BOOKS showing all books with more than 5 copies.
6. Retrieve all the books sorted by Publication_Year in descending order.
7. Retrieve the details of borrowers who borrowed more than 2 books

Hard Questions:

1. Find customers who rented a movie and never returned it (use LEFT JOIN and NULL check).
2. Display the movie(s) rented the most times.

Scenario 2: Movie

Schema:-

- MOVIES: (Movie_ID, Title, Genre, Release_Date, Rating, Director)
- CUSTOMERS: (Customer_ID, Name, Email, Phone, Membership_Type)
- RENTALS: (Rental_ID, Customer_ID, Movie_ID, Rental_Date, Return_Date)

Easy Questions (10)

1. Create a table MOVIES with the given schema.
2. Insert at least 5 rows into the MOVIES table.
3. Display all the details of movies available.
4. Display the list of movies in the "Action" genre.
5. Create a table CUSTOMERS with the given schema.
6. Insert at least 5 rows into the CUSTOMERS table.
7. Display the names and emails of all customers.
8. Display the list of customers with a "Premium" membership.
9. Create a table RENTALS with the given schema.
10. Insert 5 records into the RENTALS table.

Medium Questions :-

1. Add a NOT NULL constraint to the Genre column of the MOVIES table.
2. Add a UNIQUE constraint to the Email column in the CUSTOMERS table.
3. Add a foreign key constraint on Movie_ID in the RENTALS table referencing MOVIES(Movie_ID).
4. Create an index on the Rating column in the MOVIES table to optimize queries
5. Find the average rating of all movies in the MOVIES table.
6. Find the total number of movies rented and the maximum fees paid in a single rental.
7. Update all movies with a rating below 5 to change their genre to 'Uncategorized'.
8. Delete all customer records where the Membership_Type is NULL.

Hard Questions:-

1. Retrieve customers who rented movies but have not returned them (use WHERE and NULL check).
2. Find the most rented movie(s) and the number of times they were rented

Scenario 3: Hospital

Schema:-

1) **DOCTORS:-**

- Doctor_ID: INT
- Name: VARCHAR(50) Specialty: VARCHAR(50)
- Phone: VARCHAR(15) Salary: DECIMAL(10, 2) PATIENTS
- Patient_ID: INT Name: VARCHAR(50)
- Address: VARCHAR(100)
- Disease: VARCHAR(50)
- Phone: VARCHAR(15)

2) **APPOINTMENTS:-**

- Appointment_ID: INT Doctor_ID: INT Patient_ID: INT Appointment_Date: DATE
- Fees: DECIMAL(8, 2)

Easy Level Questions (5):-

1. Display all details of doctors specializing in "Cardiology".
2. Retrieve the names and phone numbers of all patients.
3. Display the appointment details for a specific patient named "John Doe".
4. List all appointments scheduled on or after December 1, 2023.
5. Find all doctors with a salary greater than 1,00,000.

Medium Questions (10):-

1. Create an index on the Speciality column in the DOCTORS table.
2. Update the salary of all doctors in the "Pediatrics" specialty by 10%.
3. Delete the records of patients who have never been assigned an appointment.
4. Find the total consultation fees collected by each doctor.
5. Display the first three characters of each doctor's name (use a single-row function).
6. Create a savepoint after updating the salary of a specific doctor.
7. Find the average consultation fees for appointments made in January 2024.
8. Add a NOT NULL constraint to the Phone column in the DOCTORS table.

Hard Questions (5):-

1. Display the details of patients treated by doctors specializing in "Cardiology" (use JOIN).
2. Find the doctor names along with the number of patients they treated.

3. Find doctors who treated patients with more than one disease (use GROUP BY and HAVING).

Scenario 4: Inventory Management

Schema:

1) **PRODUCTS**:
- Product_ID: INT
- Name: VARCHAR(50)
- Category: VARCHAR(50)
- Price: DECIMAL(10, 2)
- Stock: INT

2) **ORDERS**:
- Order_ID: INT
- Product_ID: INT
- Quantity: INT
- Order_Date: DATE
- Customer_ID: INT

Easy Level Questions (5):

1. Display all details of products in the "Electronics" category.
2. Retrieve the names and stock levels of all products priced above 1,000.
3. Display the order details for a product named "Laptop".
4. List all orders placed on or after December 1, 2023.
5. Find all products with stock less than 10.

Medium Questions :

1. Update the price of all products in the "Groceries" category by 5%.
2. Delete records of products with zero stock.
3. Find the total quantity of each product ordered (use GROUP BY).
4. Display the first five characters of each product's name (use a single-row function).
5. Create a savepoint after updating the price of a specific product.
6. Find the average quantity ordered for all orders made in December 2023.
7. Add a composite primary key for Order_ID and Product_ID in the ORDERS table.
8. Retrieve all orders sorted by Order_Date in ascending order.

Hard Questions (5):

1. Find customers who ordered products from more than two different categories (use GROUP BY and HAVING).
2. Find the product with the highest sales revenue (use JOIN and aggregate functions).

Scenario 5: Bank

Schema:

1) **CUSTOMERS**:
- Customer_ID: INT
- Name: VARCHAR(50)
- Account_No: VARCHAR(20)
- Phone: VARCHAR(15)
- Balance: DECIMAL(12, 2)

2) **TRANSACTIONS**:
- Transaction_ID: INT
- Account_No: VARCHAR(20)
- Transaction_Date: DATE
- Amount: DECIMAL(12, 2)
- Type: VARCHAR(10) (Deposit/Withdrawal)

Easy Level Questions (5):

1. Display all details of customers with a balance greater than 50,000.
2. Retrieve the names and phone numbers of all customers.
3. Display the transaction details for a specific customer named "Jane Doe".
4. List all transactions made on or after December 1, 2023.
5. Find all customers with an account balance less than 5,000.

Medium Questions (10):

1. Create an index on the Type column in the TRANSACTIONS table.
2. Update the balance of all customers who made deposits greater than 1,00,000.
3. Delete records of customers with no transactions in the last year.
4. Find the total transaction amount for each customer (use GROUP BY).
5. Display the first three characters of each customer's name (use a single-row function).

6. Create a savepoint after updating the balance of a specific customer.
7. Find the average transaction amount for all withdrawals made in December 2023.
8. Add a NOT NULL constraint to the Phone column in the CUSTOMERS table.

Hard Questions (5):

1. Display the details of customers with transactions exceeding 1,00,000 (use JOIN).
2. Find the customer names along with their total transaction amounts (use GROUP BY).