Maharashtra Education Society's

Institute of Management and Career Courses (IMCC), Pune

AUTONOMOUS

A.Y. 2025-26

IT13: ADBMS

Chapter 2: Structured Query Language

Handson Session

Using SQL

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 singlerental.
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 WHEREand NULL check).
2. Find the most rented movie(s) and the number of times they were rent

Chapter 3: Introduction to NoSQL Database

Handson session

Using MongoDB

1. Inventory Management System

Easy Questions

1. Create a database called Inventory.
2. Create a collection called Products with attributes: ProductID, Name, Category, Stock, and Price.
3. Insert 10 product documents into the Products collection.
4. Retrieve all products with stock greater than 10.
5. Find all products from the 'Electronics' category.
6. Update the price of the product with ProductID 7 to 19.99.
7. Delete the product with ProductID 2.
8. Count how many products are in stock.
9. Sort products by Price in descending order.
10. Find products whose price is either 10 or 15.

Hard Questions

1. Retrieve products whose Category is 'Home Appliances' and Stock is greater than 30.
2. Update the Category of the product with ProductID 4 to 'Furniture'.
3. Use $nin to find exclude products with stock 10, 15 units.
4. Find products with Stock less than 5 and Price greater than 10..
5. Find products with a Name containing the character 'c'.
6. Delete all products with stock equal to 0.
7. Retrieve the first 5 products, skipping the first 2 documents.

2. Banking System

Easy Questions

1. Create a database called Bank.
2. Create a collection called Customers with attributes: CustomerID, Name, AccountBalance, and AccountType.
3. Insert 10 documents representing customer information.
4. Find all customers with AccountType 'Savings'.
5. Update the AccountBalance of the customer with CustomerID 4 to 5000.
6. Delete the customer with CustomerID 2.
7. Count the number of customers with an AccountBalance greater than 1000.
8. Sort customers by AccountBalance in ascending order.
9. Find customers whose AccountType is either 'Checking' or 'Business'.
10. Find customers with an AccountBalance between 2000 and 5000.

Hard Questions

1. Update the AccountBalance of all customers with AccountType 'Savings' by 1000.
2. Retrieve customers with an AccountBalance greater than 5000.
3. Find customers with AccountType 'Checking' and AccountBalance less than 1000.
4. Use $in to find customers with AccountType either 'Business' or 'Checking'.
5. Find customers whose AccountBalance is between 3000 and 7000.
6. Find all customers with AccountBalance less than 1000 and update it to 1500.
7. Delete all the customers .

Chapter 4: Graph Database and Analytics

Handson Session

Using Neo4j

1. Social Media Network :

Question: "Create a social media network where:

- There are five users: Alex, Maya, Ryan, Priya, and Sam
- Alex posts two photos: 'Vacation' and 'Party'
- Maya posts one photo: 'Graduation'
- Ryan comments on Alex's 'Party' photo
- Priya likes both of Alex's photos
- Sam follows everyone but hasn't posted anything
- Include post dates, comment text, and user join dates"

Execute the following Queries:

1. Find all posts by Alex
2. Find all users who liked Alex's posts
3. Find users who haven't posted anything
4. Find who follows whom and their follow
5. Find posts with comments and their commenters

Extra Query Challenges (Difficult Level):

1. Find users who both posted and commented
2. Count the number of likes per post
3. Find the most active user (combining posts, comments, and likes)

2. University Course Management System:

Question: "Create **University Course Management System where:**

- Create two students, "Arun" (age 20) and "Bala" (age 21).
- Create two courses, "CSE" and "IT".
- Create one professor, "Dr. Smith".
- Establish relationships: "Arun" enrolled in "CSE", "Bala" enrolled in "IT", and "Dr. Smith" taught both "CSE" and "IT".

Execute the following Queries:

1. Find all students and the courses they are enrolled in
2. Find which courses are taught by "Dr. Smith"
3. Find all students who are enrolled in a course taught by "Dr. Smith"
4. Find which course Bala enrolled in
5. Find all students enrolled in CSE
6. Find all students enrolled in IT
7. Find professor teaching Arun's course
8. Show complete system details
9. List all students
10. List all professors