

COURSE OUTCOMES

At the end of the course students will be able to

CO1: understand fundamentals of database management systems.

CO 2: design database models and learn database languages to write queries to extract information from databases.

CO 3: Identify database anomalies and improve the design of database management system

CO 4: understand transaction management and concurrency control.

CO 5: understand storage organization and database recovery.

COURSE CONTENT

UNIT 1

Introduction: Database management system Characteristics of the Database, Database Systems and Architecture, Data Models, Schemes & Instances, DBMS Architecture & Data Independence, Database administrator & Database Users, Database Languages & Interfaces, DDL, DML, DCL, Overview Relational Data Base Management Systems

UNIT 2

Data Modeling: Data modeling using The Entity-Relationship Model – Entities, Attributes and Relationships, Cardinality of Relationships, Strong and Weak Entity Sets, Generalization, Specialization, and Aggregation, Translating your ER Model into Relational Model, Relationships of higher degree.

UNIT 3

Relational Model, Languages & Systems: Relational Data Model concepts, Relational Model Constraints, integrity constraints, Keys domain constraints, referential integrity, assertions triggers, foreign key

Relational Algebra and calculus, SQL. Database security.

Relational Data Base Design: Functional Dependencies & Normalization for Relational Databases, Functional Dependencies, Normal Forms Based on Primary Keys, (1NF, 2NF, 3NF & BCNF), Lossless Join and Dependency Preserving Decomposition, Functional dependencies and its closure, covers and equivalence.

UNIT 4

Transaction Management: Transaction Concept and State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability: Testing of serializability, Serializability of schedules, conflict & view serializable schedule.

Concurrency Control Techniques: Lock-Based Protocols, Timestamp-based Protocols, validation based protocol. Deadlock Handling

UNIT 5

Recovery System

Recoverability: Failure Classification, Storage Structure, Recovery and Atomicity, Log-based Recovery, Shadow Paging, Recovery with Concurrent Transactions

Storage organization : Indexing, Hashing ,file storage.

List of Experiments:

Following is only a suggestive list of experiments. For better coverage faculty may increase the list of experiments.

Q 1: Consider the following relational schema

SAILORS (sid, sname, rating, date_of_birth)

BOATS (bid, bname, color)

RESERVES (sid, bid, date, time slot)

Q1. Consider the following relational schema

SAILORS (sid, sname, rating, date_of_birth)

BOATS (bid, bname, color)

RESERVES (sid, bid, date, time slot)

Write the following queries in SQL and relational algebra

- Find sailors who've reserved at least one boat
- Find names of sailors who've reserved a red or a green boat in the month of March.
- Find names of sailors who've reserved a red and a green boat
- Find sid of sailors who have not reserved a boat after Jan 2018.

- Find sailors whose rating is greater than that of all the sailors named "John"
- Find sailors who've reserved all boats
- Find name and age of the oldest sailor(s)
- Find the age of the youngest sailor for each rating with at least 2 such sailors

Q2. Consider the following relational schema:

CUSTOMER (cust_num, cust_lname, cust_fname, cust_balance);

PRODUCT (prod_num, prod_name, price)

INVOICE (inv_num, prod_num, cust_num, inv_date, unit_sold, inv_amount);

Write SQL queries and relational algebraic expression for the following

- Find the names of the customer who have purchased no item. Set default value of Cust_balance as 0 for such customers.
- Write the trigger to update the CUST_BALANCE in the CUSTOMER table when a new invoice record is entered for the customer.
- Find the customers who have purchased more than three units of a product on a day.
- Write a query to illustrate Left Outer, Right Outer and Full Outer Join.
- Count number of products sold on each date.
- As soon as customer balance becomes greater than Rs. 100,000, copy the customer_num in new table called "GOLD CUSTOMER"

called "GOLD_CUSTOMER"

- g) Add a new attribute CUST_DOB in customer table

Q 3: Consider the following relational schema

DEPARTMENT(Department_ID, Name, Location_ID)

JOB(Job_ID, Function)

EMPLOYEE(Employee_ID, name, DOB, Job_ID, Manager_ID, Hire_Date, Salary, department_id)

Answer the following queries using SQL and relational algebra:

- a) Write a query to count number of employees who joined in March 2015
- b) Display the Nth highest salary drawing employee details.
- c) Find the budget (total salary) of each department.
- d) Find the department with maximum budget.
- e) Create a view to show number of employees working in Delhi and update it automatically when the database is modified.
- f) Write a trigger to ensure that no employee of age less than 25 can be inserted in the database.

Q4: PROJECT

Students are required to develop a DBMS for the applications assigned to them. Following items are required to be submitted for the project

- a) Problem Statement
- b) ER model/ Relational Model
- c) Integrity Constraints implemented
- d) Suitable Queries to create and manage database

called "GOLD_CUSTOMER"

- g) Add a new attribute CUST_DOB in customer table

Q 3: Consider the following relational schema

DEPARTMENT(Department_ID, Name, Location_ID)

JOB(Job_ID, Function)

EMPLOYEE(Employee_ID, name, DOB, Job_ID, Manager_ID, Hire_Date, Salary, department_id)

Answer the following queries using SQL and relational algebra:

- a) Write a query to count number of employees who joined in March 2015
- b) Display the Nth highest salary drawing employee details.
- c) Find the budget (total salary) of each department.
- d) Find the department with maximum budget.
- e) Create a view to show number of employees working in Delhi and update it automatically when the database is modified.
- f) Write a trigger to ensure that no employee of age less than 25 can be inserted in the database.

Q4: PROJECT

Students are required to develop a DBMS for the applications assigned to them. Following items are required to be submitted for the project

- a) Problem Statement
- b) ER model/ Relational Model
- c) Integrity Constraints implemented
- d) Suitable Queries to create and manage database