AUGUST 29, 2024

HOMEWORK 1 A

CS 457 B

UMAIR DADA

2O281 - SFBU

1) What is a DBMS? What do we use it for?

a. A DBMS stands for Database Management System. It allows a user to create and manage a specific database. It contains different kinds of database options such as SQL and NoSQL.

b. It's used for storing and retrieving data using different queries as per requirement.

c. Example – A Bank would use a database to keep track of pending credit card payments of users. The DBMS would allow the bank to manage the interest rate, multiple kinds of cards and mark the credit card as blocked to avoid more transaction in case of unauthorized transactions or unpaid amounts.

2) What is a Relational DBMS? How is it different from DBMS? Give examples for both.

a. A RDBMS (Relational Database Management System) is type of DBMS which arranges the data in into tables with relations to other tables. All the tables represent a specific set of data and the relation between tables are managed using foreign keys.

b. Difference – DBMS has multiple options such as NoSQL, but an RDBMS uses a table to structure the data in a proper manner while having relationships with other tables as well.

c. DBMS Example – Oracle Berkeley DB is a popular DBMS that can be used to manage data in various formats.

d. RDBMS Example – PostgreSQL is a famous RDBMS that shows the data in table with cleared defined relationships, empowering complex queries and data integrity using foreign key and other constraint

3) Which user groups interact with a running DBMS?

a. Database Designers – Design the schema with proper tables, relationships and constraints. Example: Create a schema using (https://dbdiagram.io/) for an e-commerce stores that has tables for customers, products etc.

b. Database Administrator – A person responsible to maintain the Database, creating backups, recovery methods and ensuring performance and security with proper standards. Example – A Database Administrator may implement extra limitations for access controls to ensure authorized users read the data.

c. End Users – People who interact with products that use a DBMS to complete tasks such as reports. Example – A accounting clerk retrieving a specific client's quote from the application that is utilizing a DBMS.

d. Developers – Many developers create a software, website or a mobile app that require a DBMS for handling the data. Example: A developer would create a web application that would fetch and display the data in a chart on the dashboard of the application.

4) Describe the following terms:

a. Data Model – A representation of data structures required for a database with the relationships defined. Kind of a blueprint for creating the database
    i. 3 Parts:
        1. Structure – The structure of the DB like the tables, fields and records
        2. Manipulation – How the data can be modified / manipulated like the operations of the DB: Insert ,Select or Update
        3. Integrity – Terms for proper data accurate and consistency like the primary key and foreign keys.
    ii. An ER diagram is one the types of a data model for designing the DB schema by defining the entities, attributes and relationships.

b. Logical Schema – A detailed plan of the database structure, where the tables, columns, data types, and relationships are outlined.
    i. A logical schema would include tables for products, suppliers etc. with relationships via foreign keys for an Inventory Management System.

c. Conceptual Schema: A high-level design for database which contains core concepts and the relationships in them. The schema is high-level, but it doesn’t contain enough information to develop a functional database.
    i. A model for creating and managing appointments that involve calendars and diaries.

d. Physical Schema: a description of how the data will be stored in a database and how it will be managed on a storage device.
    i. This schema can specify that the student table is kept in specific server with an index of “studentIDnumber’ field for the lookup.

5) How is SQL used in RDBMS?

a. SQL is the standard language to manage and interact with the data in a RDBMS. Used in several tasks like defining the structure, manipulation and controlling the access.

b. Example – To get all employees from a table where department id is 3 then the query should select the "Name" Field from the "Employees" table, and filtering the results to only show where the "DepartmentID" is 3

6) What is redundancy? Give one advantage and disadvantage of redundancy.

a. It refers to data that is duplicate inside the database. Occurs to improve reliability or sometimes due to inefficiency.

   i. Advantage – Ensure the data is available if a copy is disturbed such as corrupt. Lost or a power outage at a specific location.

      1. Example – Storing sensitive information in multiple locations ensures that it will be available if one database is disturbed.

   ii. Disadvantage – It could lead to a more expensive storage cost and become complex to store data at multiple locations.

      1. If a user information is stored in multiple tables, updating the changes at one location and not others could lead to inconsistency and possibly cause errors.

Book Exercises

1. **(1.8) Identify some informal queries and update operations that you would expect to apply to the database shown in Figure 1.2.**

    a. Informal

        i. Retrieve a student’s grades

        ii. Courses taken by student

        iii. Credit hours for a course

        iv. Courses by a department

        v. Instructors for a course

    b. Update

        i. Update a student’s grade

        ii. Update course details

        iii. Delete a course prerequisite

        iv. Assign instructor for a course

        v. Add a new course

2. **(1.10) Specify all the relationships among the records of the database shown in Figure 1.2.**

    a. STUDENT <-> GRADE_REPORT: ‘Student_number’

    b. STUDENT <-> COURSE: ‘Major’ / ‘Department’

    c. COURSE <-> SECTION: ‘Course_number’

    d. SECTION <-> GRADE_REPORT: ‘Section_identifier’

    e. COURSE <-> PREQUISITE: ‘Course_number’ / ‘Prerequisite_number’’

3. **(1.12) Cite some examples of integrity constraints that you think can apply to the database shown in Figure 1.2.**

    a. Primary Key

        i. 'Student_number', 'Course_number' and 'Section_identifier' should be unique for each section

    b. Foreign Key

        i. GRADE_REPORT Table

            1. 'Student_number' shall exist in the STUDENT table
            2. 'Section_identifier' shall exist in the SECTION table

        ii. SECTION Table

            1. 'Couse_number' must exist in the COURSE table

        iii. PREREQUISITE Table

            1. "Couse_number' must exist in the COURSE table

    c. Domain Constraints:

        i. STUDENT Table

            1. 'Class' should be restricted to valid numeric values

        ii. COURSE Table

            1. 'Course_number' values need to be alphanumeric and unique
            2. 'Credit_hours' shall be a positive integer only

    d. Referential Integrity Constraints:

        i. Course Table: Every 'Course_number' in the SECTION table must be valid.

        ii. Section Table: Ensure that every 'Course_number' in the GRADE_REPORT table corresponds to a valid course.

    e. Entity Integrity Constraints:

        i. All records in each table must uniquely identifiable by their primary key attribute.

4. **(1.14) Consider Figure 1.2.**

   a. If the name of the ‘CS’ (Computer Science) Department changes to ‘CSSE’ (Computer Science and Software Engineering) Department and the corresponding prefix for the course number also changes, identify the columns in the database that would need to be updated.

      i. COURSE Table:
         1. Update ‘CS’ to ‘CSSE’ in the ‘Department’ columns.
         2. Update the prefix from ‘CS’ to ‘CSSE’ in ‘Course_number’ column.

      ii. SECTION Table:
         1. Update the prefix from ‘CS’ to ‘CSSE’ in ‘Course_number’ column.

      iii. PREREQUISITE Table:
         1. Update the prefix from ‘CS’ to ‘CSSE’ for both ‘Course_number’ and ‘Prerequisite_number’ columns.

   b. Can you restructure the columns in the COURSE, SECTION, and PREREQUISITE tables so that only one column will need to be updated?

      i. Create a new column named ‘Course_prefix’ in COURSE, SECTION and PREREQUISITE tables

      ii. Store the department name prefix in the ‘Course_prefix’ columns and the ‘Course_number’ only contains the numerical value of the course code.

      iii. For the PREREQUISITE table, an extra column needs to be created which would be named ‘Prerequisite_prefix’ and the course code for the prerequisite course would be in this column.