UIT UNIVERSITY

Third Semester B.E. Tech. (Sofware) Mid Term Exam Fall 2024

Database System SET-211

Time allowed: 90 mins Max Marks: 20

Instructions:

* Attempt All Question.
* Carefully read all questions first, and seek clarification if needed.
* No queries regarding the question paper will be entertained after 20 minutes from the start.
* Do not write on the question paper unless specifically instructed to do so.
* Ensure all required information is filled out, and return the question paper along with your answer script.

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll No.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [Marks-05]- [CLO# 01]

What is the significance of a Database Management System (DBMS) in data management?? Explain its key features, including data independence, data integrity, concurrency control, and security. How do these features contribute to the overall efficiency and effectiveness of managing databases?

A Database Management System (DBMS) is a software system that is designed to manage and organize data in a structured manner. It allows users to create, modify, and query a database, as well as manage the security and access controls for that database. DBMS provides an environment to store and retrieve data in convenient and efficient manner. Key Features of DBMS

**Data Independency:** Data Independence is mainly defined as a property of DBMS that helps you to change the database schema at one level of a system without requiring to change the schema at the next level. it helps to keep the data separated from all program that makes use of it.

**Concurrency control:** A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.

**Data integrity:** Data integrity is a concept and process that ensures the accuracy, completeness, consistency, and validity of an organization’s data. By following the process, organizations not only ensure the integrity of the data but guarantee they have accurate and correct data in their database.

**Data Security:** Security of databases refers to the array of controls, tools, and procedures designed to ensure and safeguard confidentiality, integrity, and accessibility.

Data Independence reduces system downtime during upgrades or schema modifications. It also supports scalability and adaptability to evolving requirements. Data Integrity ensures reliable and meaningful data for decision-making. Reduces the risk of costly errors caused by data inconsistencies. Concurrency Control improves performance in multi-user environments by enabling safe simultaneous access. Minimizes bottlenecks and conflicts, ensuring a smooth user experience. Security safeguards against data breaches, maintaining organizational reputation and legal compliance. Ensures data is only accessed and modified by authorized personnel, reducing risks of internal misuse or external threats.

1. [Marks-05]- [CLO# 01]

Consider a relational database, identify and explain the constraints that should be applied to ensure data integrity. Additionally, discuss how these constraints can help prevent anomalies in the relational model, providing examples of potential anomalies.

While designing the Relational Model, we define some conditions which must hold for data present in the database are called Constraints. These constraints are checked before performing any operation (insertion, deletion, and updation) in the database. If there is a violation of any of the constraints, the operation will fail.

**Domain Constraints:** These are attribute-level constraints. An attribute can only take values that lie inside the domain range. e.g.; If a constraint AGE>0 is applied to STUDENT relation, inserting a negative value of AGE will result in failure.

**Key Integrity:** Every relation in the database should have at least one set of attributes that defines a tuple uniquely. Those set of attributes is called keys. e.g.; ROLL\_NO in STUDENT is key. No two students can have the same roll number. So a key has two properties:

1. It should be unique for all tuples.
2. It can’t have NULL values.

**Referential Integrity:** When one attribute of a relation can only take values from another attribute of the same relation or any other relation, it is called referential integrity.

An anomaly is an irregularity or something which deviates from the expected or normal state. When designing databases, we identify three types of anomalies: Insert, Update, and Delete.

**Insertion Anomaly in Referencing Relation:.** We can’t insert a row in REFERENCING RELATION if referencing attribute’s value is not present in the referenced attribute value. e.g.; Insertion of a student with BRANCH\_CODE ‘ME’ in STUDENT relation will result in an error because ‘ME’ is not present in BRANCH\_CODE of BRANCH.

**Deletion/ Updation Anomaly in Referenced Relation:**  We can’t delete or update a row from REFERENCED RELATION if the value of REFERENCED ATTRIBUTE is used in the value of REFERENCING ATTRIBUTE. e.g; if we try to delete a tuple from BRANCH having BRANCH\_CODE ‘CS’, it will result in an error because ‘CS’ is referenced by BRANCH\_CODE of STUDENT, but if we try to delete the row from BRANCH with BRANCH\_CODE CV, it will be deleted as the value is not been used by referencing relation.

1. [Marks-05]- [CLO# 02]

Briefly describe the Entity-Relationship (E-R) model, and how do you elaborate the concepts of Entity, Weak Entity, Attribute, and Relationship within it? Provide examples to illustrate each concept.

The ER model was created to provide a simple and understandable model for representing the structure and logic of databases. The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically. The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, the ER Diagram is the structural format of the database.

Entity: An Entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.

Strong Entity: A Strong Entity is a type of entity that has a key Attribute. Strong Entity does not depend on other Entity in the Schema. It has a primary key, that helps in identifying it uniquely, and it is represented by a rectangle. These are called Strong Entity Types.

Weak Entity: An Entity type has a key attribute that uniquely identifies each entity in the entity set. But some entity type exists for which key attributes can’t be defined.

Attributes: Attributes are the properties that define the entity type. For example, Roll\_No, Name, DOB, Age, Address, and Mobile\_No are the attributes that define entity type Student. In ER diagram, the attribute is represented by an oval.

Relation: A Relationship represents the association between entity types. For example, ‘Enrolled in’ is a relationship type that exists between entity type Student and Course. In ER diagram, the relationship type is represented by a diamond and connecting the entities with lines.

Consider an example of university database. It has three entities such as student, course and enrollment. Relation is student enrolls in course and course has enrollment. Primary keys are StudentID in Student, CourseID in Course. StudentID and CourseID in Enrollment referencing Student and Course respectively. Each student can enroll in multiple courses, but each enrollment entry is uniquely identified by the composite key. Foreign key constraints ensure that only valid students and courses exist in the Enrollment table.

1. [Marks-05]- [CLO# 02]

Design an Entity-Relationship Diagram (ERD) using Chen notation for given scenario. Justify your choices of attributes for each entity.

* A salesperson may manage many other salespeople. A salesperson is managed by only one salespeople.
* A salesperson can be an agent for many customers. A customer is managed by one salespeople.
* A customer can place many orders. An order can be placed by one customer.
* An order lists many inventory items. An inventory item may be listed on many orders.
* An inventory item is assembled from many parts. A part may be assembled into many inventory items.
* Many employees assemble an inventory item from many parts.
* A supplier supplies many parts. A part may be supplied by many suppliers.

