Software EngineeringAssignment

DataBase

Module 5

1). What do you understand about databases?
Ans.

A database is a structured collection of data organized in such a way that it can be easily accessed, managed, and updated. It typically consists of tables, each containing rows and columns, where data is stored in a structured format. Databases are commonly used in computing environments to store and retrieve information efficiently.

They provide mechanisms for querying, updating, and manipulating data, as well as ensuring data integrity, security, and concurrency control.

Various types of databases exist, including relational databases, noSQL databases, object-oriented databases, and more, each designed to cater to different data storage and retrieval requirements.

2).What is Normalization?
Ans.

Normalization is the process of organizing the data in the database.

Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like insertion, update and deletion anomalies.

Normalization divides the larger table into smaller and links them using relationships.

The normal form is used to reduce redundancy from the database table.

Normalization works through a series of stages called normal forms. The normal forms apply to individual relations. The relation is said to be in particular normal form if it satisfies constraints.

3). What is Difference between DBMS and RDBMS? Ans

A database management system is software that stores and manages data. The database management system was first established in the 1960s to store any type of data.

It also allows for data modification such as insertion, deletion, and updating.

The DBMS system also manages the database by defining, generating,modifying, and regulating it. It’s built to develop and preserve data while also allowing each business application to retrieve the information it needs.

RDBMS(Relational Database Management System)

RDBMS stands for relational database management system, and it is a software system that is used to store only data in the form of tables.

Data is handled and stored in rows and columns, which are referred to as tuples and attributes, in this type of system.

RDBMS is a strong database management system that is extensively used across the world.

4). What is MF Cod Rule of RDBMS Systems?
Ans

Rule 1: The Information Rule
All information, whether it is user information or metadata, that is stored in a database must be entered as a value in a cell of a table. It is said that everything within the database is organized in a table layout.

Rule 2: The Guaranteed Access Rule
Each data element is guaranteed to be accessible logically with a combination of the table name, primary key (row value), and attribute name (column value).

Rule 3: Systematic Treatment of NULL Values
Every Null value in a database must be given a systematic and uniform treatment.

Rule 4: Active Online Catalog Rule
The database catalog, which contains metadata about the database, must be stored and accessed using the same relational database management system.

Rule 5: The Comprehensive Data Sublanguage Rule
A crucial component of any efficient database system is its ability to offer an easily understandable data manipulation language (DML) that facilitates defining, querying, and modifying information within the database.

Rule 6: The View Updating Rule
All views that are theoretically updatable must also be updatable by the system.

Rule 7: High-level Insert, Update, and Delete
A successful database system must possess the feature of facilitating high-level insertions, updates, and deletions that can grant users the ability to conduct these operations with ease through a single query.

Rule 8: Physical Data Independence
Application programs and activities should remain unaffected when changes are made to the physical storage structures or methods.

Rule 9: Logical Data Independence
Application programs and activities should remain unaffected when changes are made to the logical structure of the data, such as adding or modifying tables.

Rule 10: Integrity Independence
Integrity constraints should be specified separately from application programs and stored in the catalog. They should be automatically enforced by the database system.

Rule 11: Distribution Independence
The distribution of data across multiple locations should be invisible to users, and the database system should handle the distribution transparently.

Rule 12: Non-Subversion Rule
If the interface of the system is providing access to low-level records, then the interface must not be able to damage the system and bypass security and integrity constraints.

5). What do you understand about Data Redundancy?
Ans

Data redundancy refers to the practice of keeping data in two or more places within a database or data storage system. Data redundancy ensures an organization can provide continued operations or services in the event something happens to its data for example, in the case of data corruption or data loss.

The concept applies to areas such as databases, computer memory and file storage systems.

Data redundancy can occur within an organization intentionally or accidentally. If done intentionally the same data is kept in different locations with the organization making a conscious effort to protect it and ensure its consistency.

This data is often used for backups or disaster recovery.

When it comes to usage of storage, redundancy can be a safeguard to take the form of unwanted overhead. Data volume will often contain redundant storage blocks to reduce storage consumption within the volume or to minimize the volume of data that must be backed up.

6).What is a DDL interpreter?
Ans

A DDl (data definition language) interpreter is a component or module within a database management system responsible for processing and DDL statements. DDL statements are used to define, modify, and manage the structure and schema of a database, including tables,
Indexes,views, constraints, and other database objects.

Create:
Used to create new database objects such as tables, indexes, views, etc.

ALTER:
Used to modify the structure of existing database objects, such as adding or removing columns from a table.

DROP:
Used to delete or remove existing database objects from the database.

RENAME:
Used To rename existing Database objects.

TRUNCATE:
Used to remove all data from a table while keeping the table structure intact.

7).What is a DML Compiler in SQL?
Ans

DML is an abbreviation of Data Manipulation Language.

The DML commands in structured query language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

SELECT DML Command
Select is the most important data manipulation command in Structured Query Language. The Select command shows the records of the specified table. It also shows the particular record of a particular column by using the Where clause.

INSERT DML Command
Insert is another most important data manipulation command in Structured Query Language, which allows users to insert Data in database tables.

UPDATE DML Command
Update is another most important data manipulation command in SQL, which allows users to update or modify the existing data in database tables.

DELETE DML Command
Delete is a DML command which allows SQL users to remove single or multiple existing records from the database tables.

8). What is SQL Key Constraints writing an Example of SQL Key Constraints?

SQL key constraints are rules applied to columns in a database table to enforce data integrity and relationships between tables. There are several types of key constraints in SQL, including primary keys, foreign keys, unique keys, and check constraints.

Here's a brief overview of each type:

Primary Key: A primary key uniquely identifies each record in a table. It ensures that each row in the table has a unique identifier.

Foreign Key: A foreign key establishes a relationship between two tables. It ensures referential integrity by enforcing that values in a column (or columns) of one table must match the values in a referenced column (or columns) in another table.

Unique Key: A unique key constraint ensures that all values in a column (or combination of columns) are unique. Unlike a primary key, a table can have multiple unique keys.

Check Constraint: A check constraint enforces domain integrity by limiting the values that can be inserted into a column. It specifies a condition that must be true for every row in the table.

EXAMPLE:

Here's an example of how you might define these constraints in SQL:

```
CREATE TABLE Students (
  student_id INT PRIMARY KEY,
  student_name VARCHAR(50),
  date_of_birth DATE,
  CONSTRAINT unique_name UNIQUE (student_name)
);

CREATE TABLE Courses (
  course_id INT PRIMARY KEY,
  course_name VARCHAR(50),
  CONSTRAINT check_name CHECK (course_name <> ''),
  CONSTRAINT check_length CHECK (LENGTH(course_name)
<= 50)
);

CREATE TABLE Enrollments (
  enrollment_id INT PRIMARY KEY,
  student_id INT,
  course_id INT,
  enrollment_date DATE,
  CONSTRAINT fk_student FOREIGN KEY (student_id)
REFERENCES Students(student_id),
  CONSTRAINT fk_course FOREIGN KEY (course_id)
REFERENCES Courses(course_id)
);
```

The Students table has a primary key constraint on the student_id column and a unique key constraint on the student_name column.

The Courses table has a primary key constraint on the course_id column, and it also has check constraints to ensure that the course_name is not empty and does not exceed 50 characters.

The Enrollments table has a primary key constraint on the enrollment_id column, and it has foreign key constraints referencing the Students and Courses tables to maintain referential integrity.

9). What is save Point? How to create a save Point and write a Query?
Ans

A savepoint in SQL is a marker within a transaction that allows you to roll back part of the transaction to a specific point without rolling back the entire transaction. This can be useful when you want to preserve some parts of the transaction while undoing changes made after a certain point.

Here's how you create a savepoint using SQL:

SAVEPOINT savepoint_name;

In this query:
SAVEPOINT is the SQL keyword used to create a savepoint.

savepoint_name is the name you choose for your savepoint. You can name it whatever is meaningful for your application.

Here's an example of how you might use a savepoint in a transaction

```
START TRANSACTION;

UPDATE account SET balance = balance - 100 WHERE account_id = 123;

SAVEPOINT before_deposit;

UPDATE account SET balance = balance + 100 WHERE account_id = 456;

-- If something goes wrong, rollback to the savepoint
ROLLBACK TO before_deposit;

COMMIT;
```

==========

In this example:

We start a transaction using START TRANSACTION.

We deduct 100 from the balance of an account with ID 123.

We create a savepoint called before_deposit.

We then add 100 to the balance of an account with ID 456.

If something goes wrong after the deposit, we can rollback to the before_deposit savepoint, which will undo only the changes made after creating the savepoint.

Finally, we commit the transaction to make the changes permanent.

10). What is trigger and how to create a Trigger in SQL?
Ans

In a database, a trigger is a special type of stored procedure that automatically executes or fires in response to certain events or actions occurring within the database. These events can include data manipulation operations such as INSERT, UPDATE, DELETE, or even database schema changes like CREATE,ALTER,or DROP.

Triggers are commonly used to enforce business rules, maintain data integrity,or automate tasks.

Some Common use cases for triggers include:

Enforcing data integrity constraints:
Triggers can be used to enforce complex business rules or data validation requirements that cannot be handled solely by declarative constraints like foreign key or check constraints.

Auditing changes:
Triggers can be used to log changes made to specific tables, providing an audit trail for data modifications.

Synchronization:
Triggers can be employed to maintain consistency between related tables, ensuring that changes made to one table are reflected in another.

Generating derived data:
Triggers can automatically compute and update derived data based on changes made to other data in the database.

Overall, triggers offer a powerful mechanism for automating tasks and enforcing data integrity within a database system.

However, they should be used judiciously, as poorly designed or overly complex triggers can impact database performance and maintainability.