**SQL Assignment 1**

1. What is a relational database management system (RDBMS)? What are the advantages of a database management system over a file system?

* A relational database management system (RDBMS) is a software system that allows users to create, manage, and access relational databases. It is a specialized application designed to efficiently store, retrieve, and manipulate large sets of structured data. RDBMS follows the principles of the relational model, proposed by Edgar F. Codd in 1970, which organizes data into tables with rows and columns, and establishes relationships between these tables using keys.
* Advantages of a database management system over a file system:
* Data Integrity: RDBMS enforces data integrity through various constraints like primary keys, foreign keys, and unique constraints, ensuring that the data remains accurate and consistent. In contrast, file systems lack such mechanisms, making it easier for data inconsistencies and errors to occur.
* Data Redundancy and Duplication: RDBMS reduces data redundancy and duplication by promoting normalization techniques. It stores data in a structured manner, preventing the same information from being duplicated in multiple places, leading to a more efficient use of storage space.
* Data Security: RDBMS provides robust security features, allowing administrators to control access at different levels. Users can define access permissions, ensuring that only authorized individuals can view, modify, or delete specific data. File systems typically lack such granular control.
* Data Retrieval and Querying: RDBMS supports powerful querying languages like SQL (Structured Query Language), making it easier to retrieve specific data or perform complex operations on the data. File systems require custom data access routines, making it more challenging to efficiently retrieve information.
* Data Scalability and Performance: RDBMS is designed for scalability, enabling databases to handle large amounts of data and concurrent users. It optimizes data retrieval and manipulation, leading to better performance than traditional file systems.
* Data Backup and Recovery: RDBMS offers built-in mechanisms for data backup and recovery, ensuring data can be restored in case of hardware failures or other emergencies. File systems usually lack automated backup and recovery features.
* Data Concurrency: RDBMS manages data concurrency by implementing locking mechanisms and transaction management, allowing multiple users to access and modify data simultaneously without conflicting with each other. File systems may have issues with concurrent access and may lead to data corruption

1. In a database management system, explain the ACID properties.

* ACID Properties in Database Management Systems:
* ACID is an acronym that stands for Atomicity, Consistency, Isolation, and Durability. These properties are fundamental principles that ensure the reliability and integrity of data in a database management system. They define the behavior of transactions, which are units of work that represent a series of database operations. Here's a brief explanation of each ACID property:
* Atomicity: Atomicity ensures that a transaction is treated as a single, indivisible unit of work. Either all the operations within the transaction are executed successfully, or none of them are. If any part of the transaction fails, the entire transaction is rolled back, and the database returns to its state before the transaction started. This property ensures that the database remains consistent even in the event of failures.
* Consistency: Consistency guarantees that a transaction brings the database from one valid state to another. The database should satisfy a set of predefined rules, constraints, and conditions after the successful execution of a transaction. If a transaction violates any of these rules, the entire transaction is rolled back, preserving data consistency.
* Isolation: Isolation ensures that each transaction is executed in isolation from other transactions. Transactions operate independently and are unaware of other transactions running concurrently. This prevents interference or data corruption due to simultaneous access by multiple users. Isolation is achieved through mechanisms like locking, concurrency control, and transaction isolation levels.
* Durability: Durability guarantees that once a transaction is committed, its changes are permanent and will survive any subsequent failures, such as system crashes or power outages. The changes made by a committed transaction are stored in non-volatile storage (disk), making them persist even if the system restarts. This property ensures that the data remains reliable even in the face of unexpected events.
* Together, the ACID properties provide a strong foundation for maintaining data integrity, consistency, and reliability in a database management system. They ensure that the database can handle multiple concurrent transactions while preserving data accuracy and preventing data corruption.

1. Explain the concept of normalization.

* Concept of Normalization:
* Normalization is a database design technique used to organize data efficiently in relational databases. It involves breaking down a large, complex table into smaller, well-structured tables to eliminate data redundancy and improve data integrity. The primary goal of normalization is to minimize data anomalies and improve the efficiency of data retrieval and modification operations.
* There are several normal forms in the normalization process, each building upon the previous one. The most common normal forms are:
* First Normal Form (1NF): Ensures that each column in a table contains atomic (indivisible) values, and there are no repeating groups or arrays within rows.
* Second Normal Form (2NF): Requires that the table is in 1NF and each non-key column is fully functionally dependent on the entire primary key. It eliminates partial dependencies by moving non-key attributes to separate tables.
* Third Normal Form (3NF): Requires that the table is in 2NF and there are no transitive dependencies. Transitive dependencies occur when a non-key column depends on another non-key column, which is dependent on the primary key.
* Other normal forms, such as Boyce-Codd Normal Form (BCNF) and Fourth Normal Form (4NF), further refine the design by addressing more complex dependencies and ensuring that the database remains free from data anomalies like insertion, update, and deletion anomalies.
* Normalization helps optimize data storage, reduces data duplication, and simplifies data maintenance. It promotes a more structured and maintainable database schema, making it easier to manage and query the data effectively. However, it's important to strike a balance between normalization and denormalization, as excessive normalization can lead to complex join operations and reduced query performance.

1. Explain the many types of query languages used in relational databases. DQL, DML, DCL, and DDL are some examples.

* In relational databases, different types of query languages are used to interact with and manipulate data. Each query language serves a specific purpose and is designed to perform various tasks. Here are the main types of query languages used in relational databases:

1. Data Query Language (DQL): Data Query Language is used to retrieve data from the database. It allows users to query the database to fetch specific information or extract subsets of data. The most common Data Query Language is SQL (Structured Query Language), which is a standardized language used to interact with relational databases. SQL provides various commands such as SELECT, FROM, WHERE, GROUP BY, and ORDER BY to query and retrieve data based on specified criteria.

Example SQL query:

SELECT first\_name, last\_name FROM customers WHERE age > 30;

1. Data Manipulation Language (DML): Data Manipulation Language is used to insert, update, and delete data within the database. DML commands allow users to modify the contents of tables, add new records, update existing records, and delete unwanted data. The primary DML commands in SQL are INSERT, UPDATE, and DELETE.

Example SQL DML commands:

INSERT INTO employees (id, name, salary) VALUES (1, 'John Doe', 50000); UPDATE employees SET salary = 55000 WHERE id = 1; DELETE FROM employees WHERE id = 1;

1. Data Control Language (DCL): Data Control Language is used to manage permissions and access rights in the database. It allows database administrators to control who can access and manipulate data within the database. The main DCL commands in SQL are GRANT, REVOKE, and DENY.

Example SQL DCL commands:

GRANT SELECT, INSERT ON employees TO user1; REVOKE UPDATE ON employees FROM user2;

1. Data Definition Language (DDL): Data Definition Language is used to define and manage the structure of the database, including creating, altering, and dropping database objects like tables, indexes, and views. DDL commands are used to define the schema and data organization within the database.

Example SQL DDL commands:

CREATE TABLE employees ( id INT PRIMARY KEY, first\_name VARCHAR(50), last\_name VARCHAR(50), age INT ); ALTER TABLE employees ADD COLUMN email VARCHAR(100); DROP TABLE employees;

1. What is the difference between the main key and a composite key? Give instances of how primary key and composite are used.

In a relational database, both the primary key and composite key are used to uniquely identify records within a table, but they differ in their structure and purpose.

1. Primary Key: A primary key is a single column or a combination of columns in a table that serves as a unique identifier for each record in that table. It must have the following properties:

* Uniqueness: Each value in the primary key column(s) must be unique, ensuring that no two records have the same primary key value.
* Non-nullability: The primary key column(s) cannot contain null values. Every record must have a valid primary key value.
* Irreducibility: The primary key should not be decomposable into smaller components, meaning it should be a minimal set of columns that uniquely identify the record.

A table can have only one primary key. The primary key is crucial for maintaining data integrity, as it ensures that each record is uniquely identifiable.

Example of a Primary Key: Consider a table named "Students" with the following columns:

* Student\_ID (Primary Key)
* Name
* Age
* Address

The "Student\_ID" column, marked as the primary key, will ensure that each student in the table has a unique identifier.

1. Composite Key: A composite key is a key that consists of two or more columns working together to uniquely identify records in a table. Unlike a primary key, a composite key does not have to be a single column; it can be a combination of multiple columns. The combination of columns in the composite key provides the uniqueness for each record.

Composite keys are used when no single column can uniquely identify the records, but the combination of multiple columns does. Each column in the composite key may not be unique individually, but when combined, they form a unique identifier for each record.

Example of a Composite Key: Consider a table named "Orders" with the following columns:

* Order\_ID
* Customer\_ID
* Product\_ID

In this example, "Order\_ID" alone might not be unique, as a customer can place multiple orders. Similarly, "Customer\_ID" and "Product\_ID" alone may not be unique. However, the combination of all three columns (Order\_ID, Customer\_ID, and Product\_ID) will be unique for each record in the table. Therefore, the composite key for this table would be (Order\_ID, Customer\_ID, Product\_ID).

1. Create a table with a primary key, a column default value, and a column unique constraint in SQL.

* Below is an example of creating a table in SQL with a primary key, a column with a default value, and a column with a unique constraint:

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Department VARCHAR(50),

EmploymentDate DATE DEFAULT CURRENT\_DATE,

Email VARCHAR(100) UNIQUE

);

Explanation of the table structure:

* EmployeeID (INT): This column is set as the primary key using the PRIMARY KEY constraint. It will contain unique integer values that uniquely identify each employee.
* FirstName (VARCHAR(50)): This column will store the first name of the employees.
* LastName (VARCHAR(50)): This column will store the last name of the employees.
* Department (VARCHAR(50)): This column will store the department to which the employees belong.
* EmploymentDate (DATE): This column has a default value of CURRENT\_DATE, which means that if no value is provided when inserting a new record, the current date will be automatically assigned as the default value for this column.
* Email (VARCHAR(100)): This column has a UNIQUE constraint, meaning that each email address must be unique across all records in the table. It ensures that no two employees can have the same email address.

Please note that the exact syntax may vary slightly depending on the SQL database management system you are using (e.g., MySQL, PostgreSQL, SQL Server, etc.). The example provided here is using a generic SQL syntax that should work with most SQL databases.