**SQL Server**

**1. Security and Authentication in SQL Server**

SQL Server provides multiple security and authentication mechanisms to ensure only authorized users can access the database.

* **Authentication Modes**:
  + **Windows Authentication**: Relies on the Windows OS to authenticate users based on Active Directory credentials.
  + **SQL Server Authentication**: Uses SQL Server-specific usernames and passwords for login.
* **Authorization**:
  + **User roles**: You can assign roles to users like db\_owner, db\_reader, etc., to manage permissions.
  + **Permissions**: SQL Server provides **GRANT**, **REVOKE**, and **DENY** statements to control access to database objects like tables, views, etc.

**2. Data Integrity and Constraints in SQL Server**

Data integrity ensures the accuracy and consistency of data within a database. SQL Server supports several types of constraints to enforce data integrity:

* **Primary Key**: Ensures that each row in a table is unique.
* **Foreign Key**: Enforces a relationship between two tables, ensuring that records in one table correspond to records in another.
* **Unique Constraint**: Ensures that all values in a column are unique.
* **Check Constraint**: Limits the values that can be placed in a column.
* **Default Constraint**: Provides a default value for a column when no value is specified.

**3. Normalization and Denormalization**

* **Normalization**: Normalization is the process of structuring a relational database in such a way that it reduces redundancy and dependency by organizing fields and tables of data. There are several normal forms (NF), like 1NF, 2NF, 3NF, etc., each building on the previous one.
* 1. First Normal Form (1NF): Ensures that each column contains atomic (indivisible) values.
* 2. Second Normal Form (2NF): Ensures that all non-key columns are fully dependent on the primary key.
* 3. Third Normal Form (3NF): Ensures that there are no transitive dependencies, meaning non-key columns are dependent only on the primary key.

**Example (3NF)**: Before normalization:

Employee (EmployeeID, EmployeeName, Department, DepartmentLocation)

After normalization (3NF):

Employee (EmployeeID, EmployeeName, DepartmentID)

Department (DepartmentID, DepartmentName, DepartmentLocation)

* **Denormalization**: Denormalization is the process of combining normalized tables into larger tables to improve read performance. This may introduce redundancy but can reduce the complexity of queries.

**Example**: Combining the Employee and Department tables:

Employee (EmployeeID, EmployeeName, DepartmentName, DepartmentLocation)

**4. Backup and Recovery in SQL Server**

SQL Server provides various ways to back up databases and restore them to prevent data loss:

* **Full Backup**: A complete backup of the entire database.
* **Differential Backup**: Backs up only the data that has changed since the last full backup.
* **Transaction Log Backup**: Backs up the transaction log, which records all changes made to the database.