Lecture-6

Keys and Constraints in SQL

Keys in SQL

Keys are used to uniquely identify rows in a table and establish relationships between tables.

1. **Primary Key**
    - Uniquely identifies each row in a table.
    - **Example**: A student ID in a university database, where each ID corresponds to a unique student.
2. **Foreign Key**
    - Links two tables by referencing the primary key of another table.
    - **Example**: A "Class" table with a column for "Teacher ID" that refers to the primary key in the "Teachers" table.
3. **Unique Key**
    - Ensures all values in a column are unique.
    - **Example**: An email address in a user registration system where no two users can have the same email.
4. **Candidate Key**
    - Any column(s) that can uniquely identify rows in a table.
    - **Example**: In an employee database, both "Employee ID" and "National Insurance Number" can act as candidate keys.
5. **Composite Key**
    - A combination of two or more columns that together uniquely identify rows in a table.
    - **Example**: A flight booking system where "Passenger ID" and "Flight ID" together uniquely identify a booking.
6. **Super Key**
    - A set of one or more columns that uniquely identify rows in a table (includes all keys).
    - **Example**: In a company database, "Employee ID" alone is a super key, and "Employee ID + Department" is also a super key.
7. **Alternate Key**
    - Candidate keys not chosen as the primary key.
    - **Example**: In a vehicle database, "License Plate" could be an alternate key if "Vehicle Identification Number (VIN)" is the primary key.

Constraints in SQL

Constraints enforce rules on table data for consistency and integrity.

1. **Primary Key**
    - Ensures uniqueness and no null values.
    - **Example**: A product catalog where "Product ID" is the unique identifier for each product.
2. **Foreign Key**
    - Maintains a relationship between two tables.
    - **Example**: In a library system, a "Borrowed Books" table with a "User ID" column links to the "Users" table.
3. **Check**
    - Ensures a column's values meet a specified condition.
    - **Example**: A bank system where the "Account Balance" cannot be less than zero.
4. **Not Null**
    - Ensures a column cannot have null (empty) values.
    - **Example**: In a registration form, the "Email Address" field cannot be left empty.
5. **Unique**
    - Ensures all values in a column are unique.
    - **Example**: A mobile number in a contact list where no two entries can have the same number.
6. **Default**
    - Sets a default value for a column if no value is provided.
    - **Example**: In an order management system, the "Order Status" column defaults to "Pending" if not explicitly specified.
