SQL Constraints and Key Concepts

1. NOT NULL

Ensures the column cannot be left empty.

Example:

```
CREATE TABLE users (
  id INT,
  name VARCHAR(100) NOT NULL
);
```

2. UNIQUE

Ensures all values in a column are different.

Example:

```
CREATE TABLE employees (
  email VARCHAR(100) UNIQUE
);
```

3. PRIMARY KEY

Uniquely identifies each row (NOT NULL + UNIQUE).

Example:

```
CREATE TABLE orders (
  order_id INT PRIMARY KEY
);
```

4. FOREIGN KEY

Links one table's column to another table's primary key.

Example:

SQL Constraints and Key Concepts

```
CREATE TABLE orders (
  user_id INT,
  FOREIGN KEY (user_id) REFERENCES users(id)
);
```

5. CHECK

Validates values meet a specific condition.

Example:

```
CREATE TABLE products (
  price DECIMAL,
  CHECK (price > 0)
);
```

6. DEFAULT

Sets a default value when none is provided.

Example:

```
CREATE TABLE accounts (
  status VARCHAR(10) DEFAULT 'active'
);
```

7. INDEX

Speeds up searches on specified columns.

Example:

```
CREATE INDEX idx_name ON users(name);
```

SQL Constraints and Key Concepts

8. AUTO_INCREMENT

Automatically increases the value of a numeric column.

Example:

```
CREATE TABLE customers (
  id INT AUTO_INCREMENT,
  PRIMARY KEY(id)
);
```

9. UNIQUE + NOT NULL (Composite)

Used together to enforce a combination of uniqueness and presence.

Example:

```
CREATE TABLE login (
  username VARCHAR(50) NOT NULL UNIQUE
);
```

10. COMPOSITE PRIMARY KEY

Combines two or more columns to uniquely identify a row.

Example:

```
CREATE TABLE enrollment (
  student_id INT,
  course_id INT,
  PRIMARY KEY (student_id, course_id)
);
```