

SQL – Keys and Constraints (Understanding Notes)

1. Introduction

Keys and constraints are essential in SQL to maintain data integrity and establish relationships between tables.

2. Primary Key

- Uniquely identifies each record in a table.
- Cannot contain NULL values.
- Must be unique.

Example:

```
CREATE TABLE Employees ( employee_id SERIAL PRIMARY KEY, name VARCHAR(50),  
department VARCHAR(50) );
```

3. Composite Key

- Combination of two or more columns that uniquely identify a record.

Example:

```
CREATE TABLE Orders ( order_id SERIAL, customer_id INT, order_date DATE, PRIMARY KEY  
(order_id, customer_id) );
```

4. Foreign Key

- Links one table to another.
- Ensures referential integrity.

Example:

```
CREATE TABLE Orders ( order_id SERIAL PRIMARY KEY, customer_id INT, order_date DATE,  
FOREIGN KEY (customer_id) REFERENCES Employees (employee_id) );
```

5. Unique Constraint

- Ensures all values in a column are distinct.

Example:

```
CREATE TABLE Employees ( employee_id SERIAL PRIMARY KEY, email VARCHAR(50)  
UNIQUE, name VARCHAR(50) );
```

6. Composite Unique Constraint

- Ensures combination of multiple columns is unique.

Example:

```
UNIQUE (email, phone);
```

7. Check Constraint

- Enforces a condition on column values.

Example:

```
age INT CHECK (age >= 18);
```

Key Points

- Primary Key → Unique + Not NULL
- Foreign Key → Maintains relationships
- Unique → Prevents duplicate values
- Check → Validates data conditions
- Composite Key → Multiple columns form uniqueness

Conclusion

Keys uniquely identify records, while constraints enforce rules that maintain data accuracy and consistency. Mastering these concepts is crucial for building reliable and scalable database systems.