

SESSION WILL BE DIVIDED INTO

1. What is the concept of keys
2. Need of Keys in SQL
3. Exploring each key
4. Difference between each key

SECTION 1

Concept Of Keys

What is a key in SQL?

- A **key** is a field (or a combination of fields) used to **identify** rows in a table **uniquely**.
- In SQL and relational databases, keys play a crucial role in:
 - Maintaining Data Integrity
 - Preventing Duplication
 - Creating Relationships Between Tables

SECTION 2

Need of Keys

Why do we need Keys in SQL?

- **Ensure Uniqueness**
 - Keys help make sure **each row** in a table is **distinct**.
 - Without keys, duplicate data could be inserted, leading to inconsistencies.
 - *Example:* Two customers with the same name — we need a unique **customer_id** to distinguish them.
- **Enforce Data Integrity**
 - Keys enforce rules about the **validity** and **consistency** of data.
 - Example – A foreign key ensures that orders cannot be placed by non-existent customers.

SECTION 3

Types Of Keys In SQL

Different Types of Keys in SQL

- Primary Key
- Unique Key
- Foreign Key
- Composite Key

SECTION 4

Exploring Each Key

Different Types of Keys in SQL

- **Primary Key - Only 1 Primary Key is allowed in a table**
 - The Primary Key is like the DNA of a record – a unique identifier that distinguishes each entry within a table.
 - This key ensures that no two forms share the same value, maintaining data integrity and enabling effective data management.
 - It is the gold standard for identifying and differentiating records within a table. Just as each person has a unique fingerprint, each record has a unique primary key.
 - For a column to be considered as a primary key –
 - It must not contain any null value
 - All the values must be unique

SECTION 4

Exploring Each Key

Different Types of Keys in SQL

- **Unique Key**
 - The Unique Key, is a versatile SQL feature that bestows the power of distinctiveness upon specific columns or combinations of columns within a table.
 - Unique Keys are the solution to ensure that specific attributes within a table remain distinct but are open to a potential null entry.
 - Therefore, a unique column
 - Ensures all the values in the column are unique
 - Allows one NULL value
 - A table can have multiple unique keys

SECTION 4

Exploring Each Key

Different Types of Keys in SQL

- **Foreign Key**
 - Unlike other Keys primarily focusing on uniqueness, the Foreign Key focuses on relationships.
 - It forms the bridges that link information across tables, enabling a database to reflect real-world interactions and associations.
 - One of the remarkable features of these Keys are their ability to define cascading actions. These actions determine what happens to related records when a change occurs in the referenced table. Example – deleting customer data can affect orders data

SECTION 4

Exploring Each Key

Different Types of Keys in SQL

- **Composite Key**
 - Unlike single-column Keys that hinge on a solitary attribute, a Composite Key is a harmonious combination of multiple columns
 - In short combination of the columns, create a uniqueness in the records of a particular table.
 - Creating a Composite Key involves defining multiple columns as the primary or unique.
 - Composite Keys handle intricate scenarios where a single attribute can't guarantee uniqueness.

SECTION 5

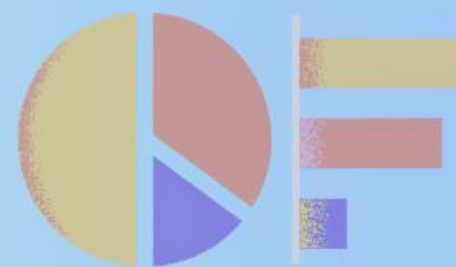
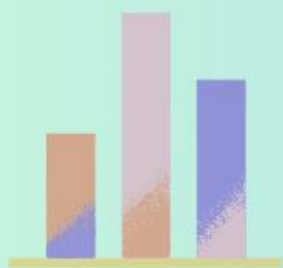
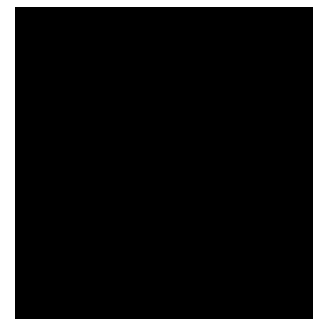
Surrogate Key

Key Type	Unique	Allows Null	Multiple per table	Main Purpose
Primary Key	Yes	No	No	Identify rows uniquely
Unique Key	Yes	Yes	Yes	Ensure Column uniqueness
Foreign Key	No	Yes	Yes	Link to Other tables
Composite Key	Yes	No	No	Uniquely identify using multiple cols

THANK YOU

A. Con at **EY**
KARAN SHAH
COMPUTER ENGINEER

LinkedIn



Power BI



python

