Lab 8

Use of Different Keys in MySQL

**Introduction**

In MySQL, keys are essential components used to establish relationships between tables, enforce data integrity, and optimize query performance. Keys are crucial for defining uniqueness, establishing relationships, and improving the efficiency of database operations. This report provides an overview of different types of keys in MySQL, including primary keys, unique keys, foreign keys, and indexes.

**Primary Key**

* **Definition**: A primary key uniquely identifies each row in a table. It ensures that the values in the key column(s) are unique and not null.
* **Properties**:
  + Uniqueness: Each value in the primary key column(s) must be unique.
  + Non-null: Values in the primary key column(s) cannot be null.
  + Automatically indexed: A primary key is automatically indexed, which means it improves search and retrieval performance.

CREATE TABLE students (

student\_id INT PRIMARY KEY,

student\_name VARCHAR(50)

);

**Unique Key**

* **Definition**: A unique key ensures that the values in the key column(s) are unique but allows null values.
* **Properties**:
  + Uniqueness: Each value in the unique key column(s) must be unique.
  + Null values: Unlike primary keys, unique keys permit null values.
* **Example**:

CREATE TABLE employees (

employee\_id INT UNIQUE,

employee\_name VARCHAR(50)

);

**Foreign Key**

* **Definition**: A foreign key establishes a relationship between two tables by linking a column in one table to the primary key or unique key of another table.
* **Properties**:
  + Referential integrity: It enforces referential integrity, ensuring that values in the foreign key column(s) match values in the referenced table's primary key or unique key column(s).
  + Cascading actions: You can specify cascading actions like **CASCADE**, **SET NULL**, or **RESTRICT** to define what happens when a referenced record is modified or deleted.
* **Example**:

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

**Conclusion**

In MySQL, keys are fundamental for ensuring data integrity, establishing relationships between tables, and optimizing database performance. Understanding the differences between primary keys, unique keys, foreign keys, and indexes is essential for designing efficient and effective database schemas. By appropriately utilizing these keys, database administrators and developers can build robust and performant MySQL databases for a wide range of applications.