Understanding `PRIMARY KEY` in MySQL

A `PRIMARY KEY` is a constraint in SQL that **uniquely identifies each row** in a table. It is one of the most important concepts in database design.

What is a Primary Key?

- A `PRIMARY KEY`:
    - Must be **unique**
    - Cannot be **NULL**
    - Is used to identify rows in a table
    - Can be a single column or a combination of columns
    - Each table can have **only one** primary key

Example:

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100)
);
```

How Is `PRIMARY KEY` Different from `UNIQUE`?

At first glance, `PRIMARY KEY` and `UNIQUE` might seem similar since both prevent duplicate values. But there are important differences:

Feature	`PRIMARY KEY`	`UNIQUE`
Must be unique	Yes	Yes
Allows NULL values	No	Yes (one or more NULLs allowed)
How many allowed	Only one per table	Can have multiple
Required by table	Recommended, often required	Optional
Dropping	Cannot be easily dropped	Can be dropped anytime

Example with UNIQUE

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(100) UNIQUE,
    name VARCHAR(100)
);
```

In this example:

- `id` is the unique identifier for each row.
- `email` must be unique, but is not the primary key.

Can I Drop a PRIMARY KEY?

Yes, but it is more restricted than dropping a `UNIQUE` constraint.

```
ALTER TABLE users DROP PRIMARY KEY;
```

> This may fail if the primary key is being used elsewhere (like in a foreign key or auto_increment column).

To drop a `UNIQUE` constraint:

```
ALTER TABLE users DROP INDEX email;
```

Auto Increment

In MySQL, a `PRIMARY KEY` is often used with the `AUTO_INCREMENT` attribute to automatically generate unique values for new rows.

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100)
);
```

This means that every time you insert a new row, MySQL will automatically assign a unique value to the `id` column. You can change the starting value of `AUTO_INCREMENT` using:

```
ALTER TABLE users AUTO_INCREMENT = 1000;
```

Key Takeaways

- Use `PRIMARY KEY` for the **main identifier** of a row.
- Use `UNIQUE` for enforcing **non-duplicate values** in other columns (like email or phone).
- You can have **only one primary key**, but you can have **many unique constraints**.