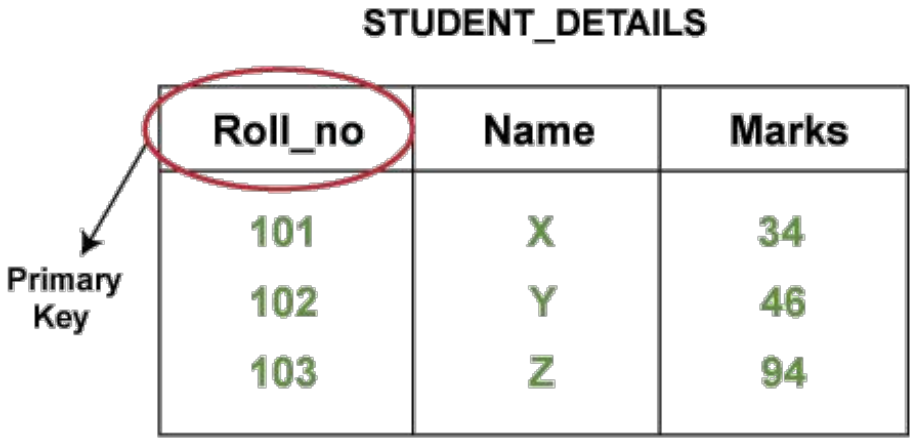
Primary Key

GROW DATA SKILLS

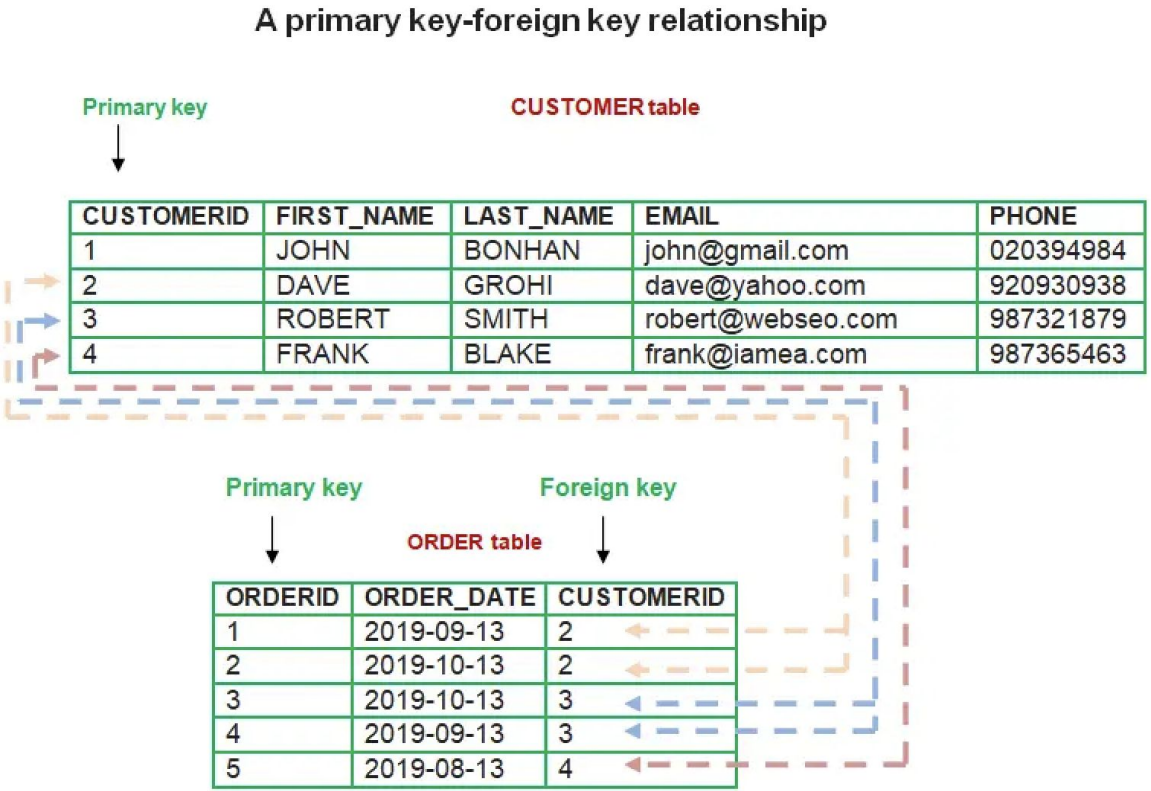
- **Uniqueness**: Each primary key value must be unique per table row.
- **Immutable**: Primary keys should not change once set.
- **Simplicity**: Ideal to keep primary keys as simple as possible.
- **Non-Intelligent**: They shouldn't contain meaningful information.
- **Indexed**: Primary keys are automatically indexed for faster data retrieval.
- **Referential Integrity**: They serve as the basis for foreign keys in other tables.
- **Data Type:** Common types are integer or string.



Foreign Key

- **Referential Integrity**: Foreign keys link records between tables, maintaining data consistency.
- **Nullable**: Foreign keys can contain null values unless specifically restricted.
- **Match Primary Keys**: Each foreign key value must match a primary key value in the parent table, or be null.
- **Ensure Relationships**: They define the relationship between tables in a database.
- **No Uniqueness**: Foreign keys don't need to be unique.

Foreign Key - Example



Delete command

The DELETE command in SQL is used to remove existing records from a table. Here's a basic syntax:

DELETE FROM table_name WHERE condition;

For example, to delete a record from a Students table where ID equals 5:

DELETE FROM Students WHERE ID = 5;

⚠️ **Be careful:** if you run the DELETE command without a WHERE clause, it will delete all records from the table.

Drop Vs Truncate vs Delete

GROW DATA SKILLS

	DROP	TRUNCATE	DELETE
Purpose	Completely removes the entire table structure from the database	Removes all rows from a table, but the table structure remains	Removes specific rows based on a condition or all rows from a table, but the table structure remains
Transaction Control	Cannot be rolled back	Cannot be rolled back	Can be rolled back
Space Reclaiming	Releases the object and its space	Frees the space containing the table	Doesn't free up space, but leaves empty space for future use
Speed	Fastest as it removes all data and structure	Faster than DELETE as it doesn't log individual row deletions	Slowest as it logs individual row deletions
Referential Integrity	Not checked	Checked	Checked
Where Clause	Not applicable	Not applicable	Applicable
Command Type	DDL (Data Definition Language)	DDL (Data Definition Language)	DML (Data Manipulation Language)

Functions

Grow Data Skills

Functions in MySQL are reusable blocks of code that perform a specific task and return a single value.

- ***Purpose***: Simplify complex calculations, enhance code reusability, and improve query performance.
- ***Types***: Built-in Functions and User-Defined Functions (UDFs).

1. **Built-in Functions**

   a. String Functions (e.g., CONCAT, LENGTH, SUBSTRING)
   b. Numeric Functions (e.g., ABS, ROUND, CEIL)
   c. Date and Time Functions (e.g., NOW, DATE_FORMAT, DATEDIFF)
   d. Aggregate Functions (e.g., COUNT, SUM, AVG)

2. **User-Defined Functions (UDFs)** Custom functions created by users to perform specific operations. It is customizable, reusable, and encapsulate complex logic.

```
DELIMITER $$
CREATE FUNCTION function_name(parameter(s))
RETURNS data_type
DETERMINISTIC
BEGIN
    -- function body
    RETURN value;
END $$
DELIMITER ;
```

Functions - Example

Grow Data Skills

Declare Function

```
DELIMITER $$
CREATE FUNCTION calculate_bonus(salary DECIMAL(10,2))
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
    RETURN salary * 0.10;
END $$
DELIMITER ;
```

Query

```
SELECT first_name, last_name, salary, calculate_bonus(salary) AS bonus FROM employees;
```

Output

```
+--------------+--------------+------------+-----------+
| first_name | last_name | salary     | bonus     |
+--------------+--------------+------------+-----------+
| John         | Doe          | 50000.00| 5000.00 |
| Jane         | Doe          | 60000.00| 6000.00 |
-------------------------------------------------------
```