**Table Level vs Field Level Constraints in SQL**

| **Aspect** | **Field Level Constraint** | **Table Level Constraint** |
| --- | --- | --- |
| **Definition** | Defined **while declaring a column**. | Defined **after all columns are declared**. |
| **Scope** | Applies to **single column only**. | Can apply to **multiple columns** together. |
| **Syntax Placement** | Right after the column definition. | After the column definitions, within the CREATE TABLE statement. |
| **Example Constraints** | NOT NULL, DEFAULT, UNIQUE, CHECK. | PRIMARY KEY (multiple columns), FOREIGN KEY, CHECK (on multiple columns). |
| **Use Case** | For simple, **column-specific rules**. | For **relationships** or conditions across columns. |

Field level-

CREATE TABLE employee (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR(50) NOT NULL,

age INT CHECK (age >= 18),

salary DECIMAL(10,2) DEFAULT 30000

);

Table level-

CREATE TABLE orders (

order\_id INT,

product\_id INT,

quantity INT,

PRIMARY KEY (order\_id, product\_id),

CHECK (quantity > 0)

);

 PRIMARY KEY (order\_id, product\_id) → defines a **composite primary key**.

 CHECK (quantity > 0) applies at **table level**.

### ****Summary****

* Use **field level** for individual column restrictions.
* Use **table level** when:
  + Applying constraints on **multiple columns**.
  + Defining **composite primary keys or foreign keys**.
  + Applying complex CHECK constraints involving multiple fields.