**What are Constraints in Oracle?**

In Oracle, **constraints** are rules that are applied to columns in a database table to ensure the accuracy, integrity, and validity of the data.

Constraints enforce specific conditions on the data, preventing invalid or unwanted data from being inserted into the database.

**Types of Constraints in Oracle:**

1. **NOT NULL** – Ensures that a column cannot have a NULL value.
2. **UNIQUE** – Ensures that all values in a column are distinct (no duplicates).
3. **PRIMARY KEY** – A combination of NOT NULL and UNIQUE. It ensures that each row in a table has a unique identifier and that no null values are allowed in the primary key column.
4. **FOREIGN KEY** – Ensures that a column’s value matches a value in another table’s column, establishing a relationship between the tables.
5. **CHECK** – Ensures that all values in a column satisfy a specific condition or rule.
6. **DEFAULT** – Assigns a default value to a column if no value is provided.

**How to Create Constraints in Oracle:**

You can create constraints when you create a table or by adding them to an existing table.

**1. NOT NULL Constraint**

Ensures that a column cannot contain NULL values.

**Example:**

CREATE TABLE employees (

emp\_id INT NOT NULL,

emp\_name VARCHAR2(100) NOT NULL

);

Here, the emp\_id and emp\_name columns cannot have NULL values.

**Output:**

If you try to insert a NULL value into these columns, Oracle will throw an error.

-- Inserting a row with NULL values in NOT NULL columns

INSERT INTO employees (emp\_id, emp\_name) VALUES (1, NULL);

-- Error: ORA-01400: cannot insert NULL into ("EMPLOYEES"."EMP\_NAME")

**2. UNIQUE Constraint**

Ensures that all values in a column are unique (no duplicates).

**Example:**

CREATE TABLE employees (

emp\_id INT PRIMARY KEY,

emp\_email VARCHAR2(100) UNIQUE

);

Here, the emp\_email column must have unique values.

**Output:**

-- Inserting unique email addresses

INSERT INTO employees (emp\_id, emp\_email) VALUES (1, 'john.doe@example.com');

INSERT INTO employees (emp\_id, emp\_email) VALUES (2, 'jane.doe@example.com');

-- Inserting a duplicate email address will give an error

INSERT INTO employees (emp\_id, emp\_email) VALUES (3, 'john.doe@example.com');

-- Error: ORA-00001: unique constraint (EMPLOYEES.EMAIL\_UNIQUE) violated

**3. PRIMARY KEY Constraint**

Ensures that the column has a unique value for every row, and no NULL values are allowed.

**Example:**

CREATE TABLE employees (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR2(100)

);

Here, emp\_id is the primary key, so it must be unique and cannot be NULL.

**Output:**

If you try to insert duplicate or NULL values into the emp\_id column, it will raise an error.

-- Inserting rows with unique emp\_id

INSERT INTO employees (emp\_id, emp\_name) VALUES (1, 'John Doe');

INSERT INTO employees (emp\_id, emp\_name) VALUES (2, 'Jane Smith');

-- Trying to insert a duplicate emp\_id will give an error

INSERT INTO employees (emp\_id, emp\_name) VALUES (1, 'Mike Davis');

-- Error: ORA-00001: unique constraint (EMPLOYEES.PK\_EMPLOYEES) violated

**4. FOREIGN KEY Constraint**

Ensures that a column’s value matches a value in another table’s column, enforcing a relationship between the two tables.

**Example:**

CREATE TABLE departments (

dept\_id INT PRIMARY KEY,

dept\_name VARCHAR2(100)

);

CREATE TABLE employees (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR2(100),

dept\_id INT,

FOREIGN KEY (dept\_id) REFERENCES departments (dept\_id)

);

Here, the dept\_id column in the employees table must match a dept\_id in the departments table.

**Output:**

-- Inserting into departments table

INSERT INTO departments (dept\_id, dept\_name) VALUES (1, 'HR');

INSERT INTO departments (dept\_id, dept\_name) VALUES (2, 'Finance');

-- Inserting into employees table with valid dept\_id

INSERT INTO employees (emp\_id, emp\_name, dept\_id) VALUES (101, 'John Doe', 1);

-- Trying to insert a row with invalid dept\_id will give an error

INSERT INTO employees (emp\_id, emp\_name, dept\_id) VALUES (102, 'Jane Smith', 3);

-- Error: ORA-02291: integrity constraint (EMPLOYEES.FK\_EMPLOYEES\_DEPT\_ID) violated - parent key not found

**5. CHECK Constraint**

Ensures that a column's value satisfies a specific condition.

**Example:**

CREATE TABLE employees (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR2(100),

salary INT,

CHECK (salary > 0)

);

This CHECK constraint ensures that the salary column must always have a value greater than 0.

**Output:**

-- Inserting a valid salary

INSERT INTO employees (emp\_id, emp\_name, salary) VALUES (1, 'John Doe', 5000);

-- Inserting an invalid salary (negative value)

INSERT INTO employees (emp\_id, emp\_name, salary) VALUES (2, 'Jane Smith', -1000);

-- Error: ORA-02290: check constraint (EMPLOYEES.SYS\_C008443) violated

**6. DEFAULT Constraint**

Specifies a default value for a column when no value is provided.

**Example:**

CREATE TABLE employees (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR2(100),

hire\_date DATE DEFAULT SYSDATE

);

This DEFAULT constraint will automatically set the hire\_date to the current date if no date is provided during insertion.

**Output:**

-- Inserting a row without specifying hire\_date (defaults to current date)

INSERT INTO employees (emp\_id, emp\_name) VALUES (1, 'John Doe');

-- The hire\_date will be automatically set to the current date

SELECT \* FROM employees;

**Why Are Constraints Important?**

1. **Data Integrity**: Constraints ensure that the data in your tables is accurate and consistent. For example, the FOREIGN KEY constraint ensures that the relationship between tables remains valid.
2. **Prevent Invalid Data**: Constraints like NOT NULL, CHECK, and DEFAULT prevent invalid or unwanted data from being entered into the database.
3. **Enforce Business Rules**: Constraints enforce business rules, such as ensuring that salary values are positive or that employees belong to a valid department.
4. **Improve Query Performance**: Constraints like PRIMARY KEY and UNIQUE help the database optimize access and searches, as they ensure the uniqueness of data.

**Summary of Common Constraints:**

| **Constraint** | **Description** | **Example** |
| --- | --- | --- |
| **NOT NULL** | Ensures a column cannot have NULL values | emp\_name VARCHAR2(100) NOT NULL |
| **UNIQUE** | Ensures all values in a column are distinct | emp\_email VARCHAR2(100) UNIQUE |
| **PRIMARY KEY** | Combines NOT NULL and UNIQUE to uniquely identify rows | emp\_id INT PRIMARY KEY |
| **FOREIGN KEY** | Ensures a column value matches a value in another table | FOREIGN KEY (dept\_id) REFERENCES departments |
| **CHECK** | Ensures values meet a specified condition | CHECK (salary > 0) |
| **DEFAULT** | Specifies a default value for a column | hire\_date DATE DEFAULT SYSDATE |

Constraints help ensure that your database remains consistent, accurate, and adheres to your business rules.

Top of Form