Difference Between Unique and Distinct

Contents

Overview

This document is intended to provide Difference between Unique and Distinct in oracle.

Difference between unique and distinct

Feature	DISTINCT	UNIQUE
Functionality	Eliminates duplicate rows from the result set	Eliminates duplicate rows from the result set (same as DISTINCT). Affects the table structure and ensures data integrity.
Use Case	Used to select unique rows in a SELECT query DISTINCT only affects the result set of a query.	Used to select unique rows in a SELECT query (same as DISTINCT). A UNIQUE constraint can be used as a primary key.
Performance	No significant difference in performance	No significant difference in performance

Typical Usage	SELECT DISTINCT column_name FROM table;	SELECT UNIQUE column_name FROM table;
		CREATE TABLE table_name (column_name data_type UNIQUE);

Data Integrity	Does not enforce any constraints on the table.	Ensures no duplicate values are inserted into the column(s).
NULL Values	Allows NULL values and treats them as distinct.	Allows NULL values; NULL values are considered unique and can be inserted multiple times.
Error Handling	No error if duplicates are present in a query result.	Error occurs if duplicate values are inserted into a column with a UNIQUE constraint.

Example	SELECT DISTINCT first_name FROM employees;	SELECT UNIQUE first_name FROM employees;
		CREATE TABLE employees (first_name VARCHAR2(50) UNIQUE);

Example:

DISTINCT Example: Suppose you have a table employees:

first_name	**last_name**
John	Doe
Alice	Smith

first_name	**last_name**
John	Doe
Bob	Brown

Using DISTINCT:

```
SELECT DISTINCT first_name FROM employees;
```

Output:

```
first_name
-----------
John
Alice
Bob
```

The DISTINCT keyword removes the duplicate John entries from the query result.

UNIQUE Constraint Example: To ensure that no two employees have the same first_name, you would define the column as UNIQUE:

```
CREATE TABLE employees (
   first_name VARCHAR2(50) UNIQUE,
   last_name  VARCHAR2(50)
);
```

Now, if you try to insert two rows with the same `first_name`, the database will prevent it:

```
INSERT INTO employees (first_name, last_name) VALUES ('John', 'Doe');  -- Success

INSERT INTO employees (first_name, last_name) VALUES ('John', 'Smith');  -- Error: Duplicate entry for 'John'
```

Conclusion:

- **`DISTINCT`** is used for filtering duplicate rows in the result set of a query.
- **`UNIQUE`** (as a constraint) is used for ensuring that no duplicate values exist in a table column, ensuring data integrity at the schema level.