UNIQUE CONSTRAINT

SQL Constraints Unique constraints in SQL is used to check whether the sub-query has duplicate tuples in its result. It returns a boolean value indicating the presence/absence of duplicate tuples. Unique constraint returns true only if the subquery has no duplicate tuples, else it returns false.

Important Points

- Evaluates to true on an empty subquery.
- Returns true only if there are unique tuples present as the output of the sub-query (two tuples are unique if the value of any attribute of the two tuples differs).
- Returns true if the sub-query has two duplicate rows with at least one attribute as NULL.

Syntax

```
CREATE TABLE table_name (
  column1 datatype UNIQUE,
  column2 datatype,
  ...
);
```

Parameter Explanation

UNIQUE: It means that in that particular column, the data should be unique.

We can directly calculate the unique data in the column without using unique data words in SQL but the UNIQUE keyword ease the complexity.Suppose that we have one table in which we have to calculate the unique data in the ID column.

Query

```
SELECT table.ID FROM table
WHERE UNIQUE (SELECT table2.ID FROM table2 WHERE table.ID =
table2.ID );
```

Note: During the execution, first the outer query is evaluated to obtain the table. ID. Following this, the inner subquery is processed which produces a new relation that contains the output of the inner query such as the table.ID == table2.ID. If every row in the new relation is unique then unique returns true and the corresponding table.ID is added as a tuple in the output relation produced. However, if every row in the new relationship is not unique then unique evaluates to false and the corresponding table.ID is not added to the output relation. Unique applied to a subquery return false if and only if there are two tuples t1 and t2 such that t1 = t2. It considers t1 and t2 to be two different tuples when unique is applied to a subquery that contains t1 and t2 such that t1 = t2 and at least one of the attributes of these tuples contains a NULL value. The unique predicate in this case evaluates to true. This is because, a NULL value in SQL is treated as an unknown value therefore, two NULL values are considered to be distinct.

The SQL statement without a UNIQUE clause can also be written as:

Query

```
SELECT table.ID FROM table
WHERE 1 <= (SELECT COUNT(table2.ID) FROM table2
WHERE table.ID = table2.ID );
```

Rules for the data in a table can be specified using SQL constraints.

The kinds of data that can be entered into a table are restricted by constraints. This guarantees the reliability and accuracy of the data in the table. The action is stopped if there is a violation between the constraint and the data action column-level or table-level constraints are both possible. Table-level constraints apply to the entire table, while column-level constraints only affect the specified column.

In SQL, the following restrictions are frequently applied:

- **NULL VALUE**: A column cannot have a NULL value by using the NOT NULL flag.
- **UNIQUE KEY**: A unique value makes sure that each value in a column is distinct.
- **PRIMARY KEY**: A NOT NULL and UNIQUE combination. Identifies each row in a table individually.
- **A FOREIGN KEY**: Guards against actions that would break links between tables.
- **CHECK**: Verifies that the values in a column meet a particular requirement.

SQL Unique Constraint on ALTER Table

We can add a unique column in a table using ALTER.Suppose that we have one table with the named instructor and we want to insert one unique column in the instructor.

Syntax

```
ALTER TABLE Instructor
ADD UNIQUE(ID);
```

To DROP a Unique Constraint

Suppose we have to DROP that column, particularly in the table.

Syntax

```
ALTER TABLE Instructor
DROP INDEX ID;
```

Queries

Find all the instructors that taught at most one course in the year 2017.

Instructor relation

EmployeeID	Name	CourseID	Year

77505	Alan	SC110	2017
77815	Will	CSE774	2017
85019	Smith	EE457	2017
92701	Sam	PYS504	2017
60215	Harold	HSS103	2016
77505	Alan	BIO775	2017
92701	Sam	ECO980	2017

Without Using Unique Constraint

Query

```
SELECT I.EMPLOYEEID, I.NAME
FROM Instructor AS I
WHERE (
  SELECT DISTINCT Inst.EMPLOYEEID
  FROM Instructor AS Inst
  WHERE I.EMPLOYEEID = Inst.EMPLOYEEID
    AND Inst.YEAR = 2017
);
```

By Using Unique Constraint

Query

```
SELECT Name
FROM Instructor
WHERE Year = 2017
GROUP BY Name
HAVING COUNT(DISTINCT CourseID) = 1;
```

Output

EmployeeID	Name
77815	Will
85019	Smith

Explanation: In the Instructor relation, only instructors Will and Smith teach a single course during the year 2017. The sub-query corresponding to these instructors contains only a single tuple and therefore the unique clause corresponding to these instructors evaluates to true thereby producing these two instructors in the output relation.

Example

Find all the courses in the Computer Science department that has only a single instructor allotted to that course.

Course Relation

CourseID	Name	Department	InstructorID
CSE505	Computer Network	Computer Science	11071
CSE245	Operating System	Computer Science	74505
CSE101	Programming	Computer Science	12715
HSS505	Psychology	Social Science	85017
EE475	Signals & Systems	Electrical	22150
CSE314	DBMS	Computer Science	44704
CSE505	Computer Network	Computer Science	11747
CSE314	DBMS	Computer Science	44715

Without Unique Constraint

Query

```
SELECT C.COURSEID, C.NAME
FROM Course AS C
WHERE EXISTS (
  SELECT T.INSTRUCTORID
  FROM Course AS T
  WHERE T.COURSEID = C.COURSEID
    AND C.DEPARTMENT = 'Computer Science'
);
```

By Using UNIQUE Constraint

Query

```
SELECT CourseID
FROM Instructor
WHERE CourseID LIKE 'CSE%'
GROUP BY CourseID
HAVING COUNT(DISTINCT Name) = 1;
```

Output

COURSE_ID	Name
CSE245	Operating System
CSE101	Programming

Explanation: In the course relation, the only courses in the computer science department that have a single instructor allotted are Operating Systems and Programming. The unique constraint corresponding to these two courses has only a single tuple consisting of the corresponding instructors. So, the unique clause for these two courses evaluates to true and these courses are displayed in output relation. Other courses in the Course relation either have two or more instructors or they do not belong to the computer science department and therefore, those courses aren't displayed in the output relation.

Frequently Asked Questions on Unique Constraint – FAQs

What is a unique constraint in SQL?

A UNIQUE constraint ensures that all values in a column are unique. UNIQUE and PRIMARY KEY constraints provide uniqueness guarantees for a column or set of columns. PRIMARY KEY constraints automatically have UNIQUE constraints.