

Created by Ahsan Arif

Unique Constraints

A **unique constraint** is a single field or combination of fields that uniquely defines a record. Some of the fields can contain null values as long as the combination of values is unique.

Note:

In Oracle, a unique constraint can not contain more than 32 columns.

A unique constraint can be defined in either a CREATE TABLE statement or an ALTER TABLE statement.

What is the difference between a unique constraint and a primary key?

Primary Key	Unique Constraint
None of the fields that are part of the primary key can contain a null value.	Some of the fields that are part of the unique constraint can contain null values as long as the combination of values is unique.

Oracle does not permit you to create both a primary key and unique constraint with the same columns.

Using a CREATE TABLE statement

The syntax for creating a unique constraint using a CREATE TABLE statement is:

```
CREATE TABLE table_name
(column1 datatype null/not null,
column2 datatype null/not null,
...
CONSTRAINT constraint_name UNIQUE (column1, column2, . . . column_n)
);
```

For example:

```
CREATE TABLE supplier
( supplier_id    numeric(10) not null,
  supplier_name varchar2(50) not null,
  contact_name  varchar2(50),
  CONSTRAINT supplier_unique UNIQUE (supplier_id)
);
```

In this example, we've created a unique constraint on the supplier table called `supplier_unique`. It consists of only one field - the `supplier_id` field.

We could also create a unique constraint with more than one field as in the example below:

```
CREATE TABLE supplier
( supplier_id    numeric(10) not null,
  supplier_name varchar2(50) not null,
  contact_name  varchar2(50),
  CONSTRAINT supplier_unique UNIQUE (supplier_id,
  supplier_name)
);
```

Using an ALTER TABLE statement

The syntax for creating a unique constraint in an ALTER TABLE statement is:

```
ALTER TABLE table_name
add CONSTRAINT constraint_name UNIQUE (column1, column2, ... column_n);
```

For example:

```
ALTER TABLE supplier
add CONSTRAINT supplier_unique UNIQUE (supplier_id);
```

In this example, we've created a unique constraint on the existing supplier table called `supplier_unique`. It consists of the field called `supplier_id`.

We could also create a unique constraint with more than one field as in the example below:

```
ALTER TABLE supplier
add CONSTRAINT supplier_unique UNIQUE (supplier_id, supplier_name);
```

Drop a Unique Constraint

The syntax for dropping a unique constraint is:

```
ALTER TABLE table_name  
drop CONSTRAINT constraint_name;
```

For example:

```
ALTER TABLE supplier  
drop CONSTRAINT supplier_unique;
```

In this example, we're dropping a unique constraint on the supplier table called `supplier_unique`.

Disable a Unique Constraint

The syntax for disabling a unique constraint is:

```
ALTER TABLE table_name  
disable CONSTRAINT constraint_name;
```

For example:

```
ALTER TABLE supplier  
disable CONSTRAINT supplier_unique;
```

In this example, we're disabling a unique constraint on the supplier table called `supplier_unique`.

Enable a Unique Constraint

The syntax for enabling a unique constraint is:

```
ALTER TABLE table_name  
enable CONSTRAINT constraint_name;
```

For example:

```
ALTER TABLE supplier  
enable CONSTRAINT supplier_unique;
```

In this example, we're enabling a unique constraint on the supplier table called `supplier_unique`.