

Aim 6: Write queries to use 'DEFAULT' constraint?

Query: create table sample (ID number(3) default 0, name varchar2(20), Bdate date Default SYSDATE);

Output: Table created.

Query: insert into sample (id, name) values (100, 'A');

Output: 1 row inserted.

Query: Select * from sample;

ID	NAME	BDATE
100	A	08-11-23

 13/9/23

Aim1: Write a query to use 'NOT NULL' constraint?

Syntax: create table tablename(col1 datatype NOT NULL, ...);

Query: insert into sample1 values(NULL, 'A');

Output: cannot insert NULL into ("R21BQ1A4285", "SAMPLE1", "ID")

Aim2: Write a query to use 'CHECK' constraint?

Query: insert into sample2 values(150, 'B');

Output: check constraint (R21BQ1A4285.SYS_C0051789) violated.

Aim3: Write queries to use 'UNIQUE' constraint?

Query: insert into sample3 values(100, 'A');

Output: 1 row inserted

Query: insert into sample3 values(100, 'B');

Output: unique constraint (R21BQ1A4285.SYS_C0051794) violated.

Aim4: Write queries to use 'PRIMARY KEY' constraint?

Query: insert into sample5 values(100, 'A');

Output: 1 row inserted.

Query: insert into sample5 values(100, 'B');

Output: unique constraint (R21BQ1A4285.SYS_C0051961) violated.

Aim5: Write queries to use 'FOREIGN KEY' constraint?

Query: update bills set custcode=104 where custcode=100;

Output: integrity constraint (R21BQ1A4285.SYS_C0051991) violated - parent key not found.

Query: delete customer1 where custcode=100;

Output: integrity constraint (R21BQ1A4285.SYS_C0051991) violated - child record found.

table, which means more than one unique constraint can exist on a single table.

- Using the UNIQUE constraint, you can also modify the already created tables.
- Syntax

```
CREATE TABLE Tablename (col1 datatype UNIQUE, col2 datatype, ...);
```

PRIMARY KEY:

- PRIMARY KEY Constraint is a combination of NOT NULL and unique constraints.
- NOT NULL Constraint and a unique constraint together forms a PRIMARY constraint.
- The column to which we have applied the primary constraint will always contain a unique value and will not allow null values.

FOREIGN KEY:

- A foreign Key is used for referential integrity.
- When we have two tables, and one table takes reference from another table, i.e., the same column is present in both the tables and that column acts as a primary Key in one table.
- That particular column will act as a foreign Key in another table.

CHECK:

- Whenever a check constraint is applied to the table's column and the user wants to insert the value in it, then the value will first be checked for certain conditions before inserting the value into that column.

DEFAULT:

- Whenever a default constraint is applied to the table's column, and the user has not specified the value to be inserted in it, then the default value which was specified while applying the default constraint will be inserted into that particular column.

Experiment -

Aim: Queries on Constraints in SQL.

Description:

Constraints in SQL means we are applying certain conditions (or) restrictions on the database. This further means that before inserting data into the database, we are checking for some conditions. If the condition we have applied to the database holds true for the data which is to be inserted, then only the data will be inserted into the database tables.

Constraints available in SQL are

1. NOT NULL
2. UNIQUE
3. PRIMARY KEY
4. FOREIGN KEY
5. CHECK
6. DEFAULT
7. CREATE INDEX

NOT NULL:

- NULL means empty, i.e., the value is not available.
- Whenever a table's column is declared as NOT NULL, then the value for that column cannot be empty for any of the table's records.
- There must exist a value in the column to which the NOT NULL constraint is applied.
- Syntax

```
CREATE TABLE TableName(col1 datatype NOT NULL, col2 datatype, ---);
```

UNIQUE:

- Duplicate values are not allowed in the columns to which the UNIQUE constraint is applied.
- The column with the unique constraint will always contain a unique value.
- This constraint can be applied to one (or) more than one column of a