

DBMS Lab-3

INTEGRITY CONSTRAINTS

- Constraints are the rules enforced on columns of a table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.
- Constraints could be column level or table level.
 - Column level constraints (constraints on single column)
 - Table level constraints (constraints on composite columns), applied to the whole table
- Commonly Used constraint available in SQL are:
 - PRIMARY KEY
 - UNIQUE
 - NOT NULL
 - CHECK
 - FOREIGN KEY (REFERENTIAL INTEGRITY)

Note: Except not null, every constraint can be either table level or column level. Not Null can be enforced only at column level

Defining Table Level Constraints :

Syntax:

```
CREATE TABLE <table-name>
(column-1 datatype,...,column-n datatype
,[CONSTRAINT <constraint-name1>] <INTEGRITY_CONSTRAINT1>(column_name(s)),
[CONSTRAINT <constraint-name2>] <INTEGRITY_CONSTRAINT2>(column_name(s)...)
);
```

Defining Column Level Constraint:

Syntax:

```
CREATE TABLE <table-name>
(column-1 datatype [CONSTRAINT <constraint-name1>] <INTEGRITY_CONSTRAINT1>,
column-2 datatype [CONSTRAINT <constraint-name2>] <INTEGRITY_CONSTRAINT2>,...,
column-n datatype);
```

Defining Integrity constraints using the ALTER command:

Syntax:

```
ALTER TABLE <table-name>
ADD [CONSTRAINT <constraint-name>]
<INTEGRITY_CONSTRAINT>(column_name(s));
```

Note: constraint name should be unique in a schema, by default the system will create the constraint name

USER_CONSTRAINTS Table:

This table provides the details about the constraints put on the columns of a table.

```
SELECT constraint_name, constraint_type FROM user_constraints WHERE table_name='SAILORS';
```

Note: When enforcing an integrity constraint by using alter command on tables that already have data in them, we need to make sure that the data existing in the column should comply with the integrity constraint that is being enforced on the column.

USER_CONSTRAINTS Table:

This table has the following columns

Column Name	Description
OWNER	The owner of the constraint
CONSTRAINT_NAME	The name of the constraint
TABLE_NAME	The name of the table associated with the constraint
CONSTRAINT_TYPE	The type of the constraint: P : primary key R: Foreign Key C: Check/Not Null U: Unique
SEARCH_CONDITION	The search condition used for check constraints
R_OWNER	The owner of the table referenced by the Foreign key constraints
R_CONSTRAINT_NAME	The name of the constraint referenced by the Foreign key constraints

Dropping Integrity constraints using the ALTER command:

Syntax:

```
ALTER TABLE <table-name> DROP <INTEGRITY _CONSTRAINT>;
```

(OR)

```
ALTER TABLE <table-name> DROP CONSTRAINT <constraint-name>;
```

Note: check constraint and foreign key constraint can be dropped with constraint name only

PRIMARY KEY CONSTRAINT

- Primary Key is a column or set of columns that uniquely Identifies a row
- Primary key will not allow duplicate values (unique)
- Primary key will not allow null values (not null)
- Only one primary key is allowed per table

Example: (Table level)

```
CREATE TABLE employees(empid number, ssn number, PRIMARY KEY (empid, ssn));  
      (OR)
```

```
CREATE TABLE employees(empid number,ssn number, CONSTRAINT pk_empid_ssn PRIMARY KEY(empid,ssn));
```

Example: (Column level)

```
CREATE TABLE employees(empid number PRIMARY KEY, ename varchar2(20));  
      (OR)
```

```
CREATE TABLE employees(empid number CONSTRAINT pk_empid PRIMARY KEY, ename varchar2(20));
```

Example (alter):

```
CREATE TABLE sample (sid NUMBER, sname varchar2(20));  
ALTER TABLE sample ADD PRIMARY KEY(sid);  
      (OR)
```

```
ALTER TABLE sample ADD CONSTRAINT pk_sid PRIMARY KEY(sid);
```

Example (drop):

```
ALTER TABLE employees DROP PRIMARY KEY;  
      (OR)
```

```
ALTER TABLE employees DROP CONSTRAINT pk_empid
```

CHECK CONSTRAINT

- Business Rule Validations can be applied to a table column by using CHECK constraint.
- CHECK constraints must be specified as a logical expression that either evaluates to TRUE or FALSE
- The condition must be a Boolean expression that can be evaluated using the values in the row being inserted or updated

Example: (Table level)

```
CREATE TABLE items (item_id number, qty_hand number, qty_sold number, CONSTRAINT ck_qty CHECK(qty_hand>qty_sold));
```

Example: (Column level)

```
CREATE TABLE students(sid number, grade number CONSTRAINT ck_grade CHECK (grade between 1 AND 10));
```

Example (alter):

```
CREATE TABLE students(sid number, grade number);  
ALTER TABLE students ADD CONSTRAINT ck_grade CHECK(grade between 1 AND 10);
```

Example (drop):

```
ALTER TABLE students DROP CONSTRAINT ck_grade;
```

UNIQUE CONSTRAINT

- Unique key will not allow duplicate values
- It permits multiple entries of null values
- A Table can have more than one Unique key
- Unique Key can combine upto 16 columns in a composite Unique Key

Example: (Table level)

```
CREATE TABLE emp_demo(empid number, ssn number, UNIQUE(empid,ssn));
```

Example: (Column level)

```
CREATE TABLE students(sid number, email varchar2(20) UNIQUE);
```

Example (alter):

```
CREATE TABLE students(sid number, contact number);  
ALTER TABLE students ADD CONSTRAINT un_sid_con UNIQUE(sid,contact);
```

Example (drop):

```
ALTER TABLE emp_demo DROP UNIQUE(empid,ssn);  
  
ALTER TABLE students DROP CONSTRAINT un_sid_con;
```

NOT NULL CONSTRAINT

- A Not Null constraint can be enforced only at column level
- A Null value is not equivalent to **zero** or **spaces**
- A Null value will evaluate to NULL in any expression

Example: (Column level)

```
CREATE TABLE sample(id number, name varchar2(20) NOT NULL);
```

Example (alter):

```
CREATE TABLE person(pid number PRIMARY KEY, pname varchar2(20));  
ALTER TABLE person MODIFY (pname varchar2(20) CONSTRAINT nn_pn NOT NULL);
```

Example (drop):

```
ALTER TABLE person MODIFY (pname varchar2(20) null);  
(OR )  
ALTER TABLE person DROP CONSTRAINT nn_pn;
```

Note: NOT NULL constraint can be dropped either using DROP or MODIFY, but can be added only using MODIFY option of ALTER

FOREIGN KEY/REFERENTIAL INTEGRITY CONSTRAINT

- Foreign Key is a column(s) that references a column(s) of a table, it can be the same table also
- It rejects an INSERT or UPDATE of a value, if a corresponding value does not currently exist in the master table
- Parent that is being referenced has to be unique or Primary key
- Child can have duplicates and nulls, unless specified explicitly
- Parent record can be deleted only when no corresponding child records exist in the child table
- Master table cannot be updated if child record exists
- If the **ON DELETE CASCADE** option is set, a DELETE operation in the master table will trigger a DELETE operation for corresponding records in all the child tables
- If the **ON DELETE SET NULL** option is set, a DELETE operation in the master table will set the value held by the foreign key of the child table to NULL

Example: (Table level)

```
CREATE TABLE score(sid number, score number, FOREIGN KEY(sid) REFERENCES students));
```

Example: (Column level)

```
CREATE TABLE score(sid number REFERENCES students(stud_id), score number);
```

Example (alter):

```
ALTER TABLE score ADD FOREIGN KEY(sid) REFERENCES students;
```

```
ALTER TABLE score ADD CONSTRAINT fk_sid FOREIGN KEY(sid) REFERENCES students ON DELETE CASCADE;
```

//will delete child values when parent is deleted

Example (drop):

```
ALTER TABLE students DROP CONSTRAINT fk_sid;
```

Enforcing more than one constraints on a single column of a table:

Syntax:

```
CREATE TABLE <table-name>
(column-1 datatype [CONSTRAINT <constraint-name1>] <INTEGRITY_CONSTRAINT1>
 [CONSTRAINT <constraint-name2>] <INTEGRITY_CONSTRAINT2> . . ., column-2 datatype
);
```

Example:

```
CREATE TABLE items (item_id number CONSTRAINT pk_item PRIMARY KEY CONSTRAINT chk_item check(item_id>0), qty_taken number,
qty_at_hand number, CHECK(qty_taken<qty_at_hand));
```

Adding more than one constraints on a table using the ALTER command:

Syntax:

```
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name> INTEGRITY CONSTRAINT(column)
ADD CONSTRAINT <constraint_name> INTEGRITY CONSTRAINT(column);
```

Example:

- CREATE TABLE sample(sid number, name varchar2(20));
- ALTER TABLE sample ADD CONSTRAINT pk_sid PRIMARY KEY(SID) ADD CONSTRAINT un_name UNIQUE(name);

Dropping more than one constraints from a table using the ALTER command:

Syntax:

```
ALTER TABLE <table_name>
DROP CONSTRAINT <constraint_name>
DROP CONSTRAINT <constraint_name>;
```

Example:

```
ALTER TABLE items DROP CONSTRAINT chk_item DROP CONSTRAINT pk_item ;
```

```
DROP TABLE <table-name> CASCADE constraint;
```

```
// will delete the table and the foreign key constraints of the primary key on other tables
```

```
// when we drop the primary key constraint alone, the FK constraints in the other tables are dropped as well
```

Exercise

1. Alter the table sailors and enforce the following constraints on its columns:
 - sid: primary key
 - rating: should accept only values between 1 and 10
2. Alter the table boats and enforce the following constraints on its columns:
 - bid: primary key
3. Alter the table reserves and enforce the following constraints on its columns:
 - Sid: Foreign Key
 - bid: Foreign key
 - (sid,bid): Primary Key
 - day : Not null

Viva Questions

1. What is a savepoint
2. What is integrity constraint?
3. Differentiate grant and revoke?
4. Write the syntax of enforcing a primary key constraint on a table?
5. Differentiate drop and truncate?
6. Differentiate check and unique constraint?
7. What do you mean by table and field in SQL?
8. What is the difference between CHAR and VARCHAR2 datatype in SQL?
9. What are Entities and Relationships?
10. What are the different subsets of SQL?

References

- https://www.youtube.com/watch?v=8lRc-Pt_krk
- [https://www.youtube.com/watch?v= 4EAg8lf04Q](https://www.youtube.com/watch?v=4EAg8lf04Q)
- <http://beginner-sql-tutorial.com/sql-integrity-constraints.htm>
- <https://www.javatpoint.com/dbms-integrity-constraints>