Constraints

Constraints provide integrity to the data that is being used by an application by applying some rules or conditions. In other words, Constraints enforce limits to the data or type of data that can be inserted/updated/deleted from a table. Constraints can be added to the table while creating the table or after it is created. Constraints can be enabled and disabled whenever needed.

Levels of Constraints

A constraint can be one of the following levels.

- Column-level constraint: These refer to a single column in the table and do not specify a column name (except check constraints). They refer to the column that they follow. These constraints are declared while creating the table.

- Table-level constraint: Table-level constraints refer to one or more columns in the table. Table-level constraints specify the names of the columns to which they apply. These constraints cahn be declared wile creating the table or after creating the table.

Types of Constraints

There are six types of constraints

a. PRIMARY KEY: It uniquely identifies each value of a column or a combination of columns. It also ensures that the column has a value. Table with a PRIMARY KEY is called a master table or parent table. A table can have only one PRIMARY KEY.
b. FOREIGN KEY: It establishes a link between two tables. It refers to the PRIMARY KEY values. A table with foreign key is called a Child table. One or

more foreign keys can be declared for a table and more than one FOREIGN KEY can refer to a single primary key.

c. UNIQUE: enforces the Uniqueness of the values (i.e. avoids duplicate values) on the column(s) on which it is defined. Also these key's can Uniquely identify each row in the database table. It allows null values.

d. NOT NULL: enforces that a column does not hold NULL value which ensures a field to always contain a value. This can be declared only at column level.

e. CHECK: allows you to specify a condition on each row in a table.

f. DEFAULT: Used to set a default value for a column. Default values are added when no other value is specified.

Note:

- PRIMARY KEY and FOREIGN KEY constraints are also called as key constraints.
- When a PRIMARY KEY or a UNIQUE KEY is created , automatically a unique index will be created. The uniqueness of the values are enforced using this index.

Declaring column level constraints.

Syntax:

```
Column_name data type(size) [CONSTRAINT constraint_name ]constraint type;
```

Note: Keywords given within [ ] are optional

Example:

```
CREATE TABLE student_profile(
regno INT  PRIMARY KEY,
name VARCHAR(10) NOT NULL,
age INT,
mailid VARCHAR(30) UNIQUE);
```

Note: If the programmer does not explicitly name constraints, then the server will automatically generate a name.

Declaring column level with user defined names constraints.

Syntax:

```
Column_name data type(size) [CONSTRAINT  constraint_name] constraint type
```

Example:

```
CREATE TABLE student_profile(
regno INT  CONSTRAINT regno_pk PRIMARY KEY,
name VARCHAR(10) NOT NULL,
age INT,
mailid VARCHAR(30) CONSTRAINT  mail_un UNIQUE);
```

Declaring a foreign key constraint.

syntax:

```
CONSTRAINT constraintName    FOREIGN KEY (referringColumnName)
   REFERENCES referredTable  (referredColumn)
   ON UPDATE reference-option
  ON DELETE reference-option
```

Example:

```
CREATE TABLE sem1(
regno INT ,
mark1 NUMBER CHECK  (mark1 between 0 and 100),
mark2 NUMBER CHECK  (mark2 between 0 and 100),
FOREIGN KEY(regno) REFERENCES student_profile(regno) on delete set null
);
```

Reference options.

5 reference options are available.

- CASCADE : If a row from the parent table is deleted or updated, the values of the matching rows in the child table automatically deleted or updated.
- SET NULL: if a row from the parent table is deleted or updated, the values of the foreign key column (or columns) in the child table are set to NULL.
- RESTRICT: if a row from the parent table has a matching row in the child table, MySQL rejects deleting or updating rows in the parent table.
- NO ACTION: is the same as RESTRICT.

Declaring Table level constraint.

Example:

```
CREATE TABLE student_profile(

regno NUMBER,

name VARCHAR(10) NOT NULL,

age NUMBER CONSTRAINT chk_name CHECK(age>18),

mailid VARCHAR(30),

CONSTRAINT rg_pk PRIMARY KEY(regno),

UNIQUE(mailid);
```

Declaring constraints (Except not null) after creating the table

The ALTER TABLE command is used to add a constraint after creating the table. Since all constraints are enabled after creation by default, the values in the column must be compatible.

Syntax:

```
ALTER TABLE table_name ADD [CONSTRAINT constraint_name]
constraint_type(column_name)
```

Example:

```
ALTER TABLE customers ADD CONSTRAINT id_pk PRIMARY KEY(cust_id);
```

Adding a NOT NULL constraint

Syntax:

```
ALTER TABLE table_name MODIFY column_name NOT NULL;
```

Example:

```
ALTER TABLE customers MODIFY cust_name NOT NULL;
```

Default constraint

It is used to assign a default value for a column and it will be added when no other value is specified.

Syntax:

```
column_name data_type DEFAULT value;
ALTER TABLE table_name ALTER column_name DEFAULT value;
```

It can be set while creating the table or can be altered after creating the table. It must be a literal constant i.e. a string,number or date. It cannot be a function or expression.

Dropping a constraint

The DROP CONSTRAINT option in ALTER TABLE command is used to drop a constraint from the table.

DROP a UNIQUE Constraint

Syntax

```
ALTER TABLE table_name DROP INDEX constraint_name;
```

DROP a PRIMARY KEY Constraint

Syntax

```
ALTER TABLE table_name DROP PRIMARY KEY;
```

Drop a FOREIGN KEY Constraint

Syntax

```
ALTER TABLE table_name DROP FOREIGN KEY constraint_name;
```

DROP a CHECK Constraint

Syntax

```
ALTER TABLE table_name DROP CHECK constraint_name;
```

Check the name of the Constraints declared for a table

```
SELECT TABLE_NAME, CONSTRAINT_TYPE, CONSTRAINT_NAME
FROM information_schema.table_constraints
WHERE table_name='student';
```

Auto Increment

Auto Increment is a function that operates on numeric data types. It automatically generates sequential numeric values every time that a record is inserted into a table for a field defined as auto increment. It is used to generate primary key values. By default, the starting value for AUTO_INCREMENT is 1, and it will increment by 1 for each new record.

Example

```
CREATE TABLE `categories` (
  `category_id` int(11) AUTO_INCREMENT,
  `category_name` varchar(150) DEFAULT NULL,
  `remarks` varchar(500) DEFAULT NULL,
  PRIMARY KEY (`category_id`)
);
```

To let the AUTO_INCREMENT sequence start with another value, use the following SQL statement

Syntax

```
ALTER TABLE table_name AUTO_INCREMENT=value;
```

Example

```
ALTER TABLE categories AUTO_INCREMENT=100;
```

Exercises

Answer the following

1. What is the difference between column level and table level constraints
    a. Constraints are applied to a single row using Column Level Constraints whereas Multiple rows can be constrained using a Table Level Constraint.
    b. Constraints are applied to multiple rows using Column Level Constraints whereas a single row can be constrained using a Table Level Constraint.
    c. Constraints are applied to a single column using Column Level Constraints whereas Multiple columns can be constrained using a Table Level Constraint.
    d. Constraints are applied to multiple columns using Column Level Constraints whereas only a single column can be constrained using a Table Level Constraint.

2. What is true about not null constraints.

    a. In columns that are subject to the NOT NULL constraint, duplicate values are not allowed.
    b. When a table's column is declared as NOT NULL, no record in the table can have an empty value for that column.
    c. By applying the NOT NULL constraint, we will always ensure that the column contains a unique value and won't allow nulls.
    d. The value will first be checked for certain conditions before inserting it into the column when a NOT NULL constraint applies to a column in the table.

3. Examine the following SQL statement

```
SQL> ALTER TABLE books_transactions ADD CONSTRAINT fk_book_id
FOREIGN KEY (book_id) REFERENCES books (book_id) ON DELETE CASCADE;
```

What does ON DELETE CASCADE imply?

a. When the BOOKS table is dropped, the BOOK_TRANSACTIONS table is dropped.

b. When the BOOKS table is dropped, all the rows in the BOOK_TRANSACTIONS table are deleted but the table structure is retained.

c. When a row in the BOOKS table is deleted, the rows in the BOOK_TRANSACTIONS table whose BOOK_ID matches that of the deleted row in the BOOKS table are also deleted.

d. When a value in the BOOKS.BOOK_ID column is deleted, the corresponding value is updated in the BOOKS_TRANSACTIONS.BOOK_ID column.

4. What is a Foreign key constraint?

a. Will not allow duplicate values but allow null value

b. Used identify the records uniquely

c. Will allow null values but not duplicate values

d. Refers the values in the primary key column.

5. What is TRUE about DEFAULT constraints?

a. The value will first be checked for certain conditions before inserting it into the column when a DEFAULT constraint applies to a column in the table.

b. In the event of a DEFAULT constraint being applied to a table's column without a user specifying the value to be inserted when that constraint was applied, the default value that was specified when the constraint was applied will be put into that column.

c. An index can be created on the table using the DEFAULT constraint.

d. None of the above

6. View the exhibit and examine the ORDERS table.

ORDERS

NAME	Null?	Type

ORDER ID	NOT NULL	INT(4)
ORDATE		DATE
CUSTOMER ID		INT(3)
ORDER TOTAL		FLOAT(7,2)

The ORDERS table contains data and all orders have been assigned a customer ID. Which statement would add a NOT NULL constraint to the CUSTOMER_ID column?

a. ALTER TABLE orders MODIFY CONSTRAINT orders_cust_id_nn NOT NULL (customer_id);
b. ALTER TABLE orders ADD CONSTRAINT orders_cust_id_nn NOT NULL (customer_id};
c. ALTER TABLE orders MODIFY customer_id CONSTRAINT orders_cust_nn NOT NULL ;
d. ALTER TABLE orders ADD customer_id NUMBER(6)CONSTRAINT orders_cust_id_nn NOT NULL;

7. _____________ is the preferred method to maintain data integrity.

a. Triggers
b. Stored procedures
c. Application Logic
d. Constraints

8. Which type of constraint in mysql allows you to define custom data validation rules?

a. Primary key
b. Check
c. Not Null
d. Foreign key
e. Unique

9. Which type of constraint automatically assigns a predefined value to a column if no value is provided during insertion

a. Primary key

b. Check
c. Default
d. Foreign key
e. Unique

10. Which of the following constraints does not enforce Uniqueness (choose two
    a. Unique
    b. Primary key
    c. Foreign key
    d. Not null