Sukkur IBA University
Merit - Quality - Excellence

CSC-251: Database Management System

Semester IV (CS, SE) Section(A, B) (Fall 2020)
Course Instructor(s): Khalid Hussain

Lab 07: DDL Queries

Objective(s):

1. To learn Data Definition Language (DDL)

1: Introduction of Data Definition Language

DDL is short name of Data Definition Language, defines the database structure or database schema. DDL also defines additional properties of the data defined in the database, as the domain of the attributes. The Data Definition Language also provide the facility to specify some constraints that would maintain the data consistency.

Following are the common SQL commands:

- **CREATE** - to create a database and its objects like (table, index, views, store procedure, function, and triggers)
- **ALTER** - alters the structure of the existing database
- **DROP** - delete objects from the database
- **TRUNCATE** - remove all records from a table, including all spaces allocated for the records are removed
- **RENAME** - rename an object

DDL statements automatically commit the current transaction; they cannot be rolled back.

CREATE COMMAND:

This command is used to create database & its objects. There are two common CREATE statements available in SQL:

1. CREATE DATABASE
2. CREATE TABLE

1. CREATE DATABASE

A Database is defined as a structured set of data. So, in SQL the very first step to store the data in a well-structured manner is to create a database. The CREATE DATABASE statement is used to create a new database in SQL.

Syntax:

```
CREATE DATABASE database_name;
```

2. CREATE TABLE

The CREATE TABLE statement is used to create a table in SQL. We know that a table comprises of rows and columns. So while creating tables we have to provide all the information to SQL about the names of the columns, type of data to be stored in columns, size of the data etc. Let us now dive into details on how to use CREATE TABLE statement to create tables in SQL.

Syntax:

```
CREATE TABLE table_name
(
column1 data_type(size),
column2 data_type(size),
column3 data_type(size),
. . .
);
```

Example:

```
CREATE TABLE patient
(
ID INT(11),
name VARCHAR(30),
phone VARCHAR(15)
);
```

ALTER COMMAND:

`alter` command is used for altering the table structure, such as,

- to add a column to existing table
- to rename any existing column
- to change datatype of any column or to modify its size.
- to drop a column from the table.

`ALTER` Command: Add a new Column

Using `ALTER` command we can add a column to any existing table. Following is the syntax,

```
ALTER TABLE table_name ADD(
    column_name datatype);
```

Here is an Example for this,

```
ALTER TABLE student ADD(
    address VARCHAR(200)
);
```

The above command will add a new column `address` to the table **student**, which will hold data of type `varchar` which is nothing but string, of length 200.

`ALTER` Command: Add multiple new Columns

Using `ALTER` command we can even add multiple new columns to any existing table. Following is the syntax,

```
ALTER TABLE table_name ADD(
    column_name1 datatype1,
    column-name2 datatype2,
    column-name3 datatype3);
```

Here is an Example for this,

```
ALTER TABLE student ADD(
    father_name VARCHAR(60),
    mother_name VARCHAR(60),
    dob DATE);
```

The above command will add three new columns to the **student** table

`ALTER` Command: Add Column with default value

`ALTER` command can add a new column to an existing table with a default value too. The default value is used when no value is inserted in the column. Following is the syntax,

```
ALTER TABLE table_name ADD(
    column-name1 datatype1 DEFAULT some_value
);
```

Here is an Example for this,

```
ALTER TABLE student ADD(
    dob DATE DEFAULT '01-Jan-99'
);
```

The above command will add a new column with a preset default value to the table **student**.

ALTER Command: Modify an existing Column

ALTER command can also be used to modify data type of any existing column. Following is the syntax,

```
ALTER TABLE table_name modify(
    column_name datatype
);
```

Here is an Example for this,

```
ALTER TABLE student MODIFY
    address varchar(300);
```

Remember we added a new column address in the beginning? The above command will modify the address column of the **student** table, to now hold upto 300 characters.

ALTER Command: Modify multiple Columns

Using ALTER command we can even modify multiple columns to any existing table. Following is the syntax,

```
ALTER TABLE table_name
    modify column_name datatype,
    modify column_name2 datatype;
```

Here is an Example for this,

```
ALTER TABLE student
    modify father_name VARCHAR(70),
    modify mother_name VARCHAR(70);
```

The above command will modify the father_name & mother_name column of the **student** table, to now hold upto 70 characters.

ALTER Command: Rename a Column

Using ALTER command you can rename an existing column. Following is the syntax,

```
ALTER TABLE table_name CHANGE COLUMN
    old_column_name new_column_name datatype;
```

Here is an example for this,

```
ALTER TABLE student CHANGE COLUMN
    address location VARCHAR(20);
```

The above command will rename `address` column to `location`.

`ALTER` Command: Drop a Column

`ALTER` command can also be used to drop or remove columns. Following is the syntax,

```
ALTER TABLE table_name DROP
    column_name;
```

Here is an example for this,

```
ALTER TABLE student DROP
    address;
```

The above command will drop the `address` column from the table **student**.

`ALTER` Command: Drop Multiple Columns

`ALTER` command can also be used to drop or remove multiple columns. Following is the syntax,

```
ALTER TABLE table_name
    DROP column_name,
    DROP column_name2;
```

Here is an example for this,

```
ALTER TABLE student
    DROP address,
    DROP phone;
```

The above command will drop `address & phone` column from the table **student**.

DROP COMMAND:

Drop command is used to delete an existing SQL database or its objects.

`DROP DATABASE` Command: Delete a Database

`DROP DATABASE` command is used to drop or remove SQL database. Following is the syntax,

```
DROP DATABASE database_name;
```

Note: Be careful before dropping a database. Deleting a database will result in loss of complete information stored in the database!

`DROP TABLE` Command: Delete an object (i.e. Table, Column) from a Database

`DROP TABLE` command is used to drop or remove table from database. Following is the syntax,

```
DROP TABLE table_name;
```

`Drop` column is already covered in alter command section.

TRUNCATE COMMAND:

The `TRUNCATE TABLE` command deletes the data inside a table, but not the table itself.

Following is the syntax,

```
TRUNCATE TABLE table_name;
```

RENAME COMMAND:

The rename command is used to change the name of an existing database object(like Table,Column) to a new name.

Renaming a table does not make it to lose any data is contained within it.

Following is the syntax,

```
RENAME TABLE current_name TO new_name;
```

You can also use command to rename a table name:

```
ALTER TABLE current_name RENAME new_name;
```

Note: RENAME TABLE, unlike ALTER TABLE, can rename multiple tables within a single statement:

RENAME TABLE does not work for TEMPORARY tables. However, you can use ALTER TABLE to rename temporary tables.

RENAME TABLE works for views, except that views cannot be renamed into a different database.

Lab Task(s):

Exercise

Using Employee table, solve the following queries (1-5).

1. Create a replica of Employee table with all the records in it.
2. Add a column ‘Address’ in it.
3. Drop column ‘Address’ from it.
4. Add columns ‘House No’ character ,’Street No’ numeric, ’Area’ character ,’City’ character in it with the respective data types.
5. Change the data type of ‘House No’ from character to numeric.
6. Create the Data Definitions for each of the relations shown below, using SQL DDL. Assume the following attributes and data types:

FACULTY:

```
FacultyID (integer, primary key)

FacultyName (25 characters)
```

COURSE:

```
CourseID (8 characters, primary key)

CourseName (15 characters)
```

CLASS:

```
ClassID (8 characters)

CourseID (8 characters foreign key)

SectionNo (integer)

Semester (10 characters)
```

STUDENT:

```
StudentID (integer, primary key)

StudentName (25 characters)

FacultyID (integer foreign key)
```

7. How would you add an attribute, CLASS, to the STUDENT table?

8. Write a SQL statement to rename the table department to dept (with both methods).

9. Write a SQL statement to add a column regionId to the table locations.

10. Write a SQL statement to change the name of the column state_province to state in locations table, keeping the data type and size same.

END