

Database Management System (DBMS)

Lecture-5

Dharmendra Kumar

August 9, 2020

Database languages

- A database system provides a data definition language to specify the database schema and a data manipulation language to express database queries and updates.
- the data definition and data manipulation languages are not two separate languages; instead they simply form parts of a single database language, such as the widely used SQL language.

Data-Definition Language

- We specify a database schema by a set of definitions expressed by a special language called a data-definition language (DDL).
- It is a set of SQL commands used to create, modify, and delete database objects such as tables, views, indices etc. .
- It is normally used by DBA and Database engineers.

It provides command like -

Command	Description
CREATE	to create objects in a database
ALTER	to modify existing data in a table
DROP	to delete objects from the database
TRUNCATE	to remove all records from the table

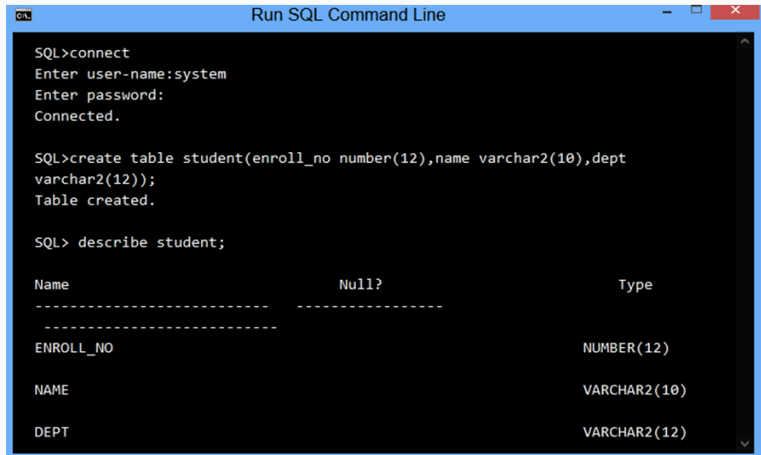
1. CREATE :

- The create command is used to create a table.
- A table name should be unique, i.e., it must not match with existing tables.
- A table name and column name must start with alphabet, must not match with reserved keywords, and should be combination of A-Z, a-z, 0-9, and '_' (underscore) having maximum length up to 30 characters.
- Each column definition requires name, data type and size for that column.
- Table name and column name are not case sensitive generally . But if they are enclosed within double quotes, then they are case sensitive.
- Each column definition is separated from other by a ',' (comma).
- The entire SQL statement is terminated with ';' (semi colon)

Database languages

Syntax: CREATE TABLE tablename(columnName1 datatype(size),column
datatype(size),....., columnNameN datatype(size));

Example: Table creation is shown in the following figure:-



```
SQL>connect
Enter user-name:system
Enter password:
Connected.

SQL>create table student(enroll_no number(12),name varchar2(10),dept
varchar2(12));
Table created.

SQL> describe student;
```

Name	Null?	Type
-----	-----	
ENROLL_NO		NUMBER(12)
NAME		VARCHAR2(10)
DEPT		VARCHAR2(12)

2. ALTER :

- Alter command used to modify structures of a table.
- Alter command can be used to add ,modify, or drop columns in a table.
- Alter command can be used for this purpose are described below:

A. Adding New Columns :

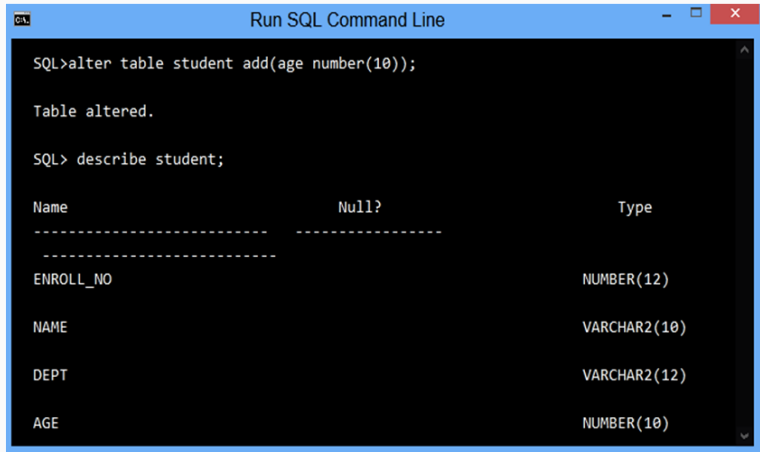
This command adds a new columns in an existing table. Initially, this column will not contain any data. If required, data can be filled for this column using UPDATE command.

Syntax:

```
Alter table TableName Add (NewColumnName Datatype(size), New-  
ColumnName Datatype(size). . . );
```

Database languages

Example: Addition of new column is shown in the following figure:-



```
SQL>alter table student add(age number(10));

Table altered.

SQL> describe student;
```

Name	Null?	Type
ENROLL_NO		NUMBER(12)
NAME		VARCHAR2(10)
DEPT		VARCHAR2(12)
AGE		NUMBER(10)

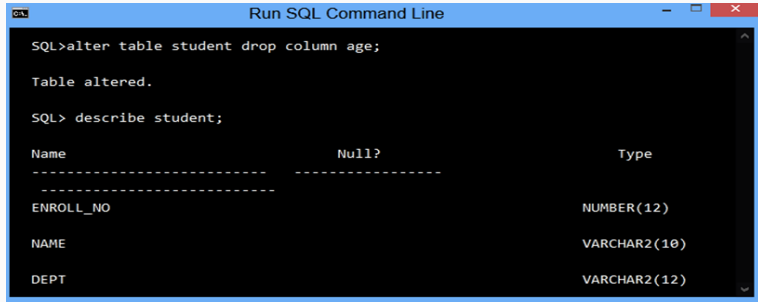
B. Dropping Columns :

This command deletes an existing column from the table along with the data held by that column.

Syntax:

Alter table TableName Drop column columnName;

Example: Deletion of a column from a table is shown in the following figure:-



The screenshot shows a window titled "Run SQL Command Line" with a black background and white text. The command prompt shows the following sequence of commands and output:

```
SQL>alter table student drop column age;

Table altered.

SQL> describe student;
```

Name	Null?	Type
ENROLL_NO		NUMBER(12)
NAME		VARCHAR2(10)
DEPT		VARCHAR2(12)

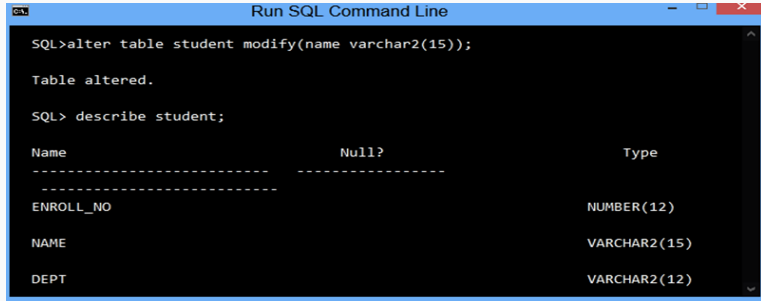
C. Modifying Columns :

This command sets newDatatype and newSize as datatype and size for specified column respectively. The main aim of this command is to modify or change the datatype and size of the column.

Syntax :

Alter table TableName Modify(columnName newDatatype(newSize));

Example:



```
SQL>alter table student modify(name varchar2(15));

Table altered.

SQL> describe student;
```

Name	Null?	Type
ENROLL_NO		NUMBER(12)
NAME		VARCHAR2(15)
DEPT		VARCHAR2(12)

3. DROP :

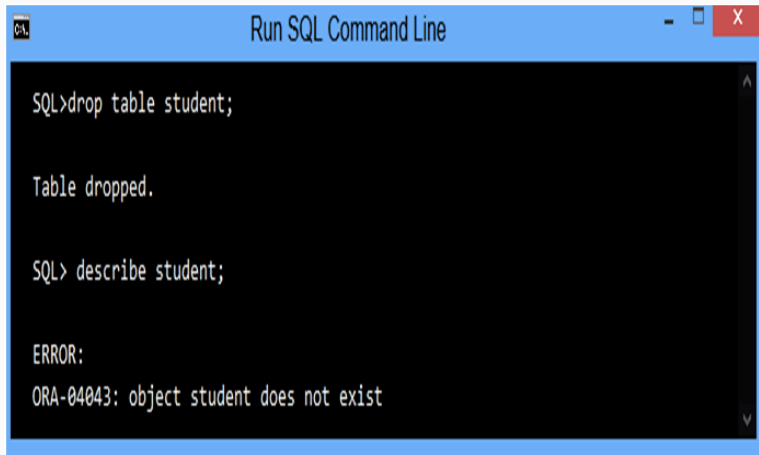
- DROP TABLE command is used to delete or destroy table from a database.
- The DROP TABLE command drops the specified table. This means, all records along with structure of the table will be destroyed.
- Care must be taken while using this command, as all records held within the table are lost and cannot be recovered.

Syntax :

drop table tablename;

Database languages

Example:



A screenshot of a Windows-style application window titled "Run SQL Command Line". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is black with white text, displaying the following SQL commands and their outputs:

```
SQL>drop table student;  
  
Table dropped.  
  
SQL> describe student;  
  
ERROR:  
ORA-04043: object student does not exist
```

4. TRUNCATE :

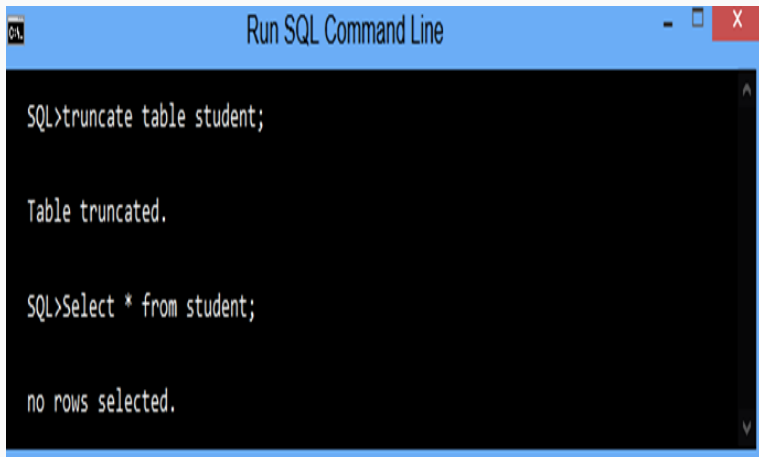
- TRUNCATE TABLE is used to delete all data from a table.
- Logically, this is equivalent to DELETE statement that deletes all rows without using WHERE clause.
- TRUNCATE operation drops and re-creates the table. This is much faster than deleting all rows one by one.
- The deleted records cannot be recovered in truncate operation. while in delete operation, deleted records can be recovered using ROLLBACK statement.

Syntax :

truncate table tablename;

Database languages

Example:



A screenshot of a Windows-style application window titled "Run SQL Command Line". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is black with white text, displaying the following SQL commands and their outputs:

```
SQL>truncate table student;  
  
Table truncated.  
  
SQL>Select * from student;  
  
no rows selected.
```

Data Manipulation Language

Data manipulation is

- The retrieval of information stored in the database
- The insertion of new information into the database
- The deletion of information from the database
- The modification of information stored in the database

Data Manipulation Language(cont.)

A data manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model. There are basically two types:

- Procedural DMLs require a user to specify what data are needed and how to get those data.
- Declarative DMLs (also referred to as non-procedural DMLs) require a user to specify what data are needed without specifying how to get those data.

Database languages

A **query** is a statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called a query language.

Following commands in SQL are used to manipulate database.

Command	Description
INSERT	insert data into table
SELECT	retrieve data from the table
UPDATE	modify existing data in the table
DELETE	delete records from the table

These all Commands are shown in detail with their syntax and example:

1. INSERT :

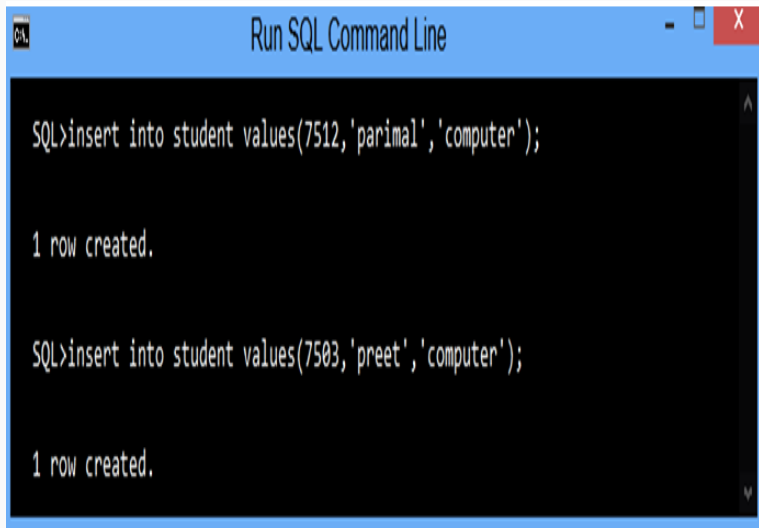
- The INSERT command is used insert the data into a table or create a new row in table.
- To insert user data into tables, "INSERT INTO ..." SQL statement is used. and stores the inserted values into respective columns.

Syntax :

insert into tablename (column1, column2, columnN) Values (expression1, expression2, ... , expressionN);

Database languages

Example:



```
Run SQL Command Line

SQL>insert into student values(7512,'parimal','computer');

1 row created.

SQL>insert into student values(7503,'preet','computer');

1 row created.
```

2. SELECT :

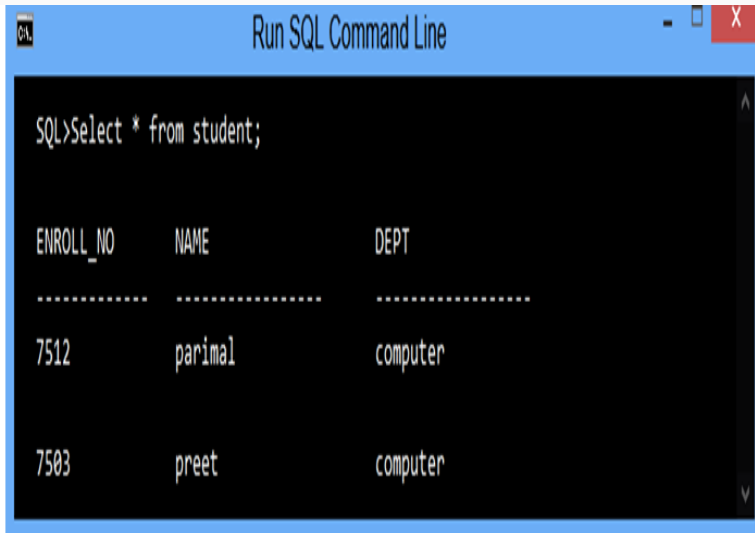
- The SELECT command is used to retrieve selected rows from one or more tables and displays on the screen. It is most widely used and required statement among all others in SQL.
- Once a table is loaded with user data, these data can be retrieved in number of different manners.
- Here, an asterisk (' * ') is used as the meta character, and it indicates all the columns of a given table.
- This statement retrieves all the columns and all the rows of the table.

Syntax :

```
select * from tablename;
```

Database languages

Example:



```
SQL>Select * from student;
```

ENROLL_NO	NAME	DEPT
7512	parimal	computer
7503	preet	computer

There are three ways of table data filtering as given below:

(A) Selected columns, All Rows

(B) Selected Rows, All Columns

(C) Selected columns, Selected Rows

Variations of the basic SELECT statement can be used to retrieve selected data as described below:

A. SELECTED COLUMNS, ALL ROWS

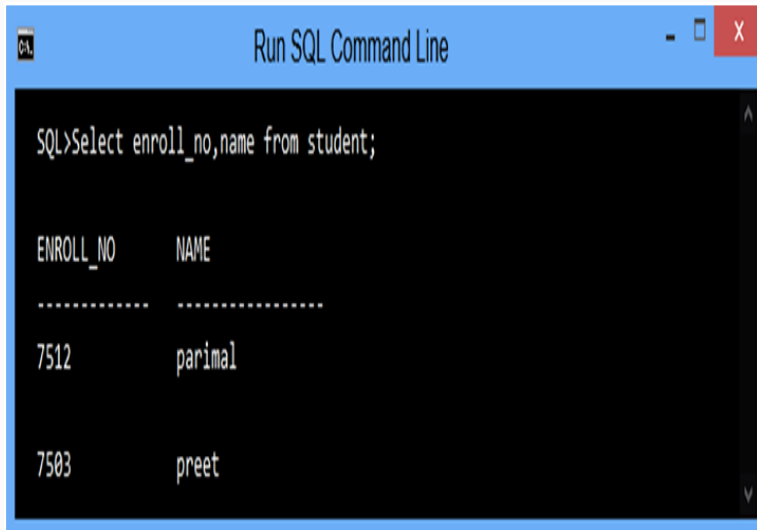
- This Statement retrieves only selected columns as specified with SELECT clause.
- This Statement retrieves all the row of the table.

Syntax :

```
select column1, column2, ..., columnN from tablename;
```

Database languages

Example:



A screenshot of a Windows-style application window titled "Run SQL Command Line". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is black with white text. It shows a SQL query being executed: "SQL>Select enroll_no,name from student;". Below the query, the results are displayed in a table format with two columns: "ENROLL_NO" and "NAME". The first row of data shows "7512" and "parimal". The second row shows "7503" and "preet". The text is in a monospaced font.

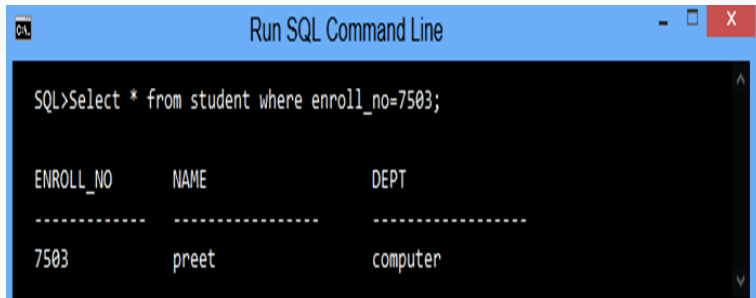
```
SQL>Select enroll_no,name from student;
```

ENROLL_NO	NAME
7512	parimal
7503	preet

B. SELECTED ROWS, ALL COLUMNS

- This Statement retrieves all the columns of the table.
- This statement retrieves only specific rows that specify the condition given with WHERE clause.
- Multiple conditions can be combined with logical operators such as AND and OR.

Syntax : `select * from tablename WHERE condition;`



The screenshot shows a window titled "Run SQL Command Line" with a black background and white text. The command entered is "SQL>Select * from student where enroll_no=7503;". Below the command, the result is displayed as a table with three columns: ENROLL_NO, NAME, and DEPT. The first row of data shows the value 7503 for ENROLL_NO, preet for NAME, and computer for DEPT. The table headers are separated by dashed lines.

```
SQL>Select * from student where enroll_no=7503;
```

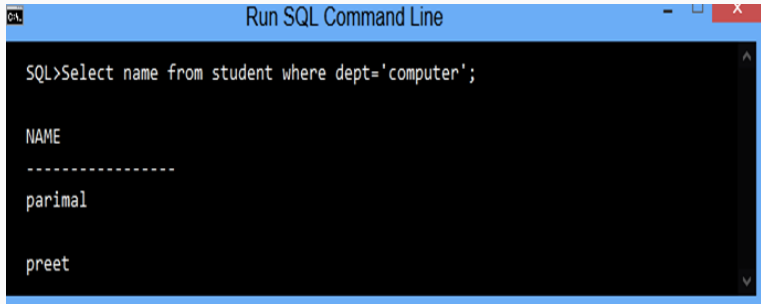
ENROLL_NO	NAME	DEPT
7503	preet	computer

C. SELECTED ROWS, SELECTED COLUMNS

- This Statement retrieves only Selected columns as specified with SELECT clause.
- Also, retrieves only specific rows that specify the condition given with WHERE clause.

Syntax :

select column1, column2, ..., columnN from tablename WHERE condition;



The screenshot shows a window titled "Run SQL Command Line" with a black background and white text. The command prompt shows the SQL query: `SQL>Select name from student where dept='computer';`. Below the query, the results are displayed as a table with a header row "NAME" and a separator row of dashes. The data rows are "parimal" and "preet".

```
SQL>Select name from student where dept='computer';

NAME
-----
parimal
preet
```


3. UPDATE :

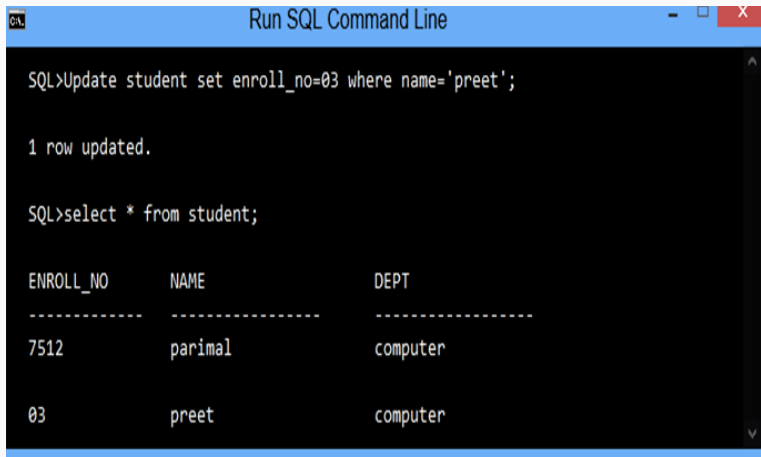
- The UPDATE command can be used to change or modify the data values in a table.
- It can be used to update either all rows or a set of rows from a table.
- This update command updates all rows from the table, and displays message regarding how many rows have been updated.
- The SET clause specifies which column data to modify.
- An expression can be a constant value, a variable, or some expression and it specifies the new value for related columns.
- You can update specific rows by the WHERE clause, and displays message regarding how many rows have been updated.

Syntax :

update tablename set column1=expression1,column2=expression2 where condition;

Database languages

Example:



```
SQL>Update student set enroll_no=03 where name='preet';

1 row updated.

SQL>select * from student;
```

ENROLL_NO	NAME	DEPT
7512	parimal	computer
03	preet	computer

4. DELETE :

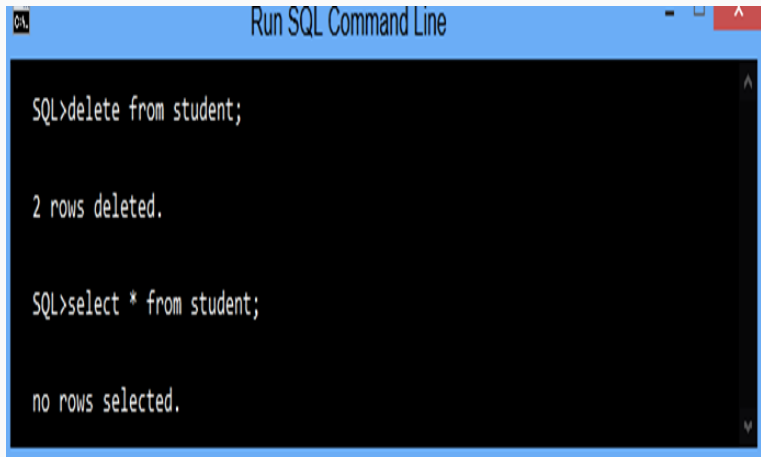
- The DELETE command can be used to remove either all rows of a table, or a set of rows from a table.
- The DELETE command deletes all rows from the table, and displays message regarding how many rows have been deleted.
- The DELETE command deletes rows from the table that satisfy the condition provided by WHERE clause. It also displays message regarding how many rows have been deleted.

Syntax :

delete from tablename;

Database languages

Example:



```
SQL>delete from student;

2 rows deleted.

SQL>select * from student;

no rows selected.
```

The image shows a screenshot of a command-line window titled "Run SQL Command Line". The window has a blue title bar with standard Windows window controls (minimize, maximize, close) on the right. The main area is black with white text. It shows two SQL commands being executed. The first command is "SQL>delete from student;", followed by the output "2 rows deleted.". The second command is "SQL>select * from student;", followed by the output "no rows selected.". A vertical scrollbar is visible on the right side of the command area.