

Data Manipulation Language

Insert, Update, Delete.

Insert Data

The INSERT INTO statement is used to insert new records in a table.

The first way specifies both the column names and the values to be inserted:

Syntax

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Exm:

```
Insert into tblPerson(EmpID,EmpName,Email,GenderID) Values(1,'John','j@n.com',1)
```

If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. The INSERT INTO syntax would be as follows:

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

Exmpl

```
Insert into tblPerson Values(1,'John','j@n.com',1)
```

Update Data

The UPDATE statement is used to modify the existing records in a table.

Syntax

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

```
Update tblPerson set age=22  
where empid=1
```

```
Update tblPerson set Genderid=2  
where empid>=7 and empid<=9
```

OR

```
Update tblPerson set Genderid=2  
where empid between 7 and 8
```

Update Warning!

if we will not use where than all data will be updated

```
-- update in column all rows
```

```
update tblGender set temp='temp'
```

SQL Delete Statement

The DELETE Statement is used to delete rows from a table.

Syntax of a SQL DELETE Statement

```
DELETE FROM table_name [WHERE condition];
```

- table_name -- the table name which has to be updated.
- DELETE FROM employee where id =100

NOTE: The *WHERE* clause in the SQL delete command is optional and it identifies the rows in the column that gets deleted. If you do not include the WHERE clause all the rows in the table is deleted, so be careful while writing a DELETE query without WHERE clause.

SQL TRUNCATE Statement

The SQL TRUNCATE command is used to delete all the rows from the table and free the space containing the table.

Syntax to TRUNCATE a table:

```
TRUNCATE TABLE table_name;
```

SQL TRUNCATE Statement Example

To delete all the rows from employee table, the query would be like,

```
TRUNCATE TABLE employee;
```

Difference between DELETE and TRUNCATE Statements:

DELETE Statement: This command deletes only the rows from the table based on the condition given in the where clause or deletes all the rows from the table if no condition is specified. But it does not free the space containing the table.

TRUNCATE statement: This command is used to delete all the rows from the table and free the space containing the table.

Difference between DROP and TRUNCATE Statement:

If a table is dropped, all the relationships with other tables will no longer be valid, the integrity constraints will be dropped, grant or access privileges on the table will also be dropped, if you want use the table again it has to be recreated with the integrity constraints, access privileges and the relationships with other tables should be established again. But, if a table is truncated, the table structure remains the same, therefore any of the above problems will not exist.

SQL DELETE Statement

Syntax

```
DELETE FROM table_name WHERE condition;
```

```
delete from tblPerson where empid =9
```

```
delete from tblPerson where empid >= 7
```

```
delete from tblPerson where empid >= 7 and empid<=9
```

How to check the table properties

Using Command

Alt + F1

Identity Column

AUTO INCREMENT Field

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.

Often this is the primary key field that we would like to be created automatically every time a new record is inserted.

```
CREATE TABLE Persons
```

```
(  
Column1 data type identity (seed, increment),  
Column2 data type .....  
Columnn data type  
)
```

Example:

Create table tblPerson1

```
(  
EmpID int IDENTITY (1, 1) Primary key,  
Name varchar(20) not null,  
)
```

Set Identity_Insert On/OFF

By default **Identity_Insert** would be OFF. To insert the same value which has been deleted need to ON Identity_Insert ON

Example:

Step 1

Insert Value

Command

Insert into tblPerson1 values ('Ramesh')

Step 2

Delete value

Delete from tblPerson1 Where ID=1

Again update the same Value.

Step 3

ON Identity_Insert

Command

Set Identity_Insert tblperson1 ON

Step 4

Insert value again

Insert into tblPerson1 (ID,Name)values(1,'Vikrant')

DBCC CHECKIDENT

If you want to delete all the rows from the table and you want to reset the identity column value, use DBCC CHECKIDENT command.

DBCC CHECKIDENT('tblPerson1',RESEED,0)