SQL Basics Part 3

* This PDF about Data Manipulation Language [DML]

/\*

=========================

== 13 INSERT Statement ==

=========================

- used to add a new records inte the table

- string values must be written into single quotes

- use "DEFAULT" if you want to insert a column's default name

- the number of columns must matches the number of values

- syntax

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

\*/

-- insert a new record

INSERT INTO customers (id, first\_name, country, score)

VALUES (9, 'Laila', 'France', 700);

-- Insert into Specific Columns

INSERT INTO customers (id, first\_name, country)

VALUES (10, 'Omar', 'Canada');

-- Insert Multiple Students at Once

INSERT INTO customers (id, first\_name, country, score)

VALUES

(11, 'Mina', 'Egypt', 620),

(12, 'Sam', 'USA', 800),

(13, 'Nour', 'Germany', 450);

/\*

----------

-- NOTE --

----------

- the SQL execute the command based on -> matching data types, column count, and constraints

- the SQL will insert data as you will write on the statement

- the columns that does not written will be NULL

\*/

INSERT INTO customers (id, first\_name, country, score)

VALUES (14, 'USA', 'Anne', 530);

FROM customers;

SELECT \*

------------------------------------------------

-- write data from one table to another table --

------------------------------------------------

-- copy data from 'customers' table into 'persons' table

-- first: select matches value from the source

-- second: insert the data from the source table to the destination table

INSERT INTO persons (id, person\_name, birth\_date, phone)

SELECT

id,

first\_name,

NULL,

'Unknwon'

FROM customers;

-- select all records from the table persons after insert

SELECT \*

FROM persons;

/\*

=========================

== 14 UPDATE Statement ==

=========================

- UPDATE used to modify the content of existing records

- don't forget the WHERE condition, else all rows will be changed

- always use WHERE clause to avoid UPDATE all rows

- syntax

UPDATE table\_name

SET column1 = value1,

column2 = value2

WHERE condition

\*/

use MyDatabase

-- modify all rows of the Egyption people with score = 600

UPDATE customers

SET score = 600

WHERE country LIKE 'Egypt';

-- change the score of customer 6 to 0

UPDATE customers

SET score = 0

WHERE id = 6;

-- change the score of the customer with ID 10 to 0 and update the country to 'UK'

UPDATE customers

SET score = 0,

country = 'UK'

WHERE id = 10;

-- insert record with NULL value

INSERT INTO customers (id, first\_name, country) VALUES (28, 'Ahmed', 'Egypt');

-- update all customers with a NULL score by setting their score to 0

UPDATE customers

SET score = 0

WHERE score IS NULL;

SELECT \*

FROM customers;

/\*

=========================

== 15 DELETE Statement ==

=========================

- DELETE Statement used to delete specific record on the table

- be careful when delete records, it is very risky

- always use WHERE clause when delete records to determine the specific row to be deleted

- syntax

DELETE FROM table\_name

WHERE condition;

- DELETE all rows

DELTE FROM table\_name;

\*/

use MyDatabase;

-- remove all customers will score = 0

DELETE FROM customers

WHERE score = 0;

-- remove all customers with NULL values on its score

DELETE FROM customers

WHERE score IS NULL;

-- delete all customers with an ID greater than 15

DELETE FROM customers

WHERE id > 15;

-- delete all customers

DELETE FROM customers;

--------------

-- TRUNCATE --

--------------

-- another way to delete all records from the table without checking or logging

-- TRUNCATE faster than DELETE

TRUNCATE TABLE persons;