

**Lecture (8)**  
**"Introduction to  
Database"**  
**SQL Database**

**Presented by : Dr. Nehal El Azaly  
Dr. Dina Abdel hafez**

# Introduction to SQL for Beginners



# SQL basic commands

After connecting to SQL server if no database exist create one

Every online SQL server will create it for you

In Local SQL you must create new one

To Create Database:

```
1  
2 CREATE DATABASE COMPANY;
```

To use Database:

You must type this command to select which Database you'll create you tables into SQL

```
1 USE COMPANY;  
2  
3
```

S  
Q

The screenshot shows a SQL IDE with a dark theme. The editor contains two lines of SQL code: '1' and '2 CREATE DATABASE COMPANY;'. The second line is selected. A right-click context menu is open over the selected text. The menu items are: 'Run MySQL Query', 'Change All Occurrences Ctrl+F2', 'Format Document Alt+F Alt+D', 'Format Document With...', 'Refactor... Ctrl+Shift+R', 'Open Changes >', 'Cut Ctrl+X', 'Copy Ctrl+C', 'Paste Ctrl+V', 'Spelling >', 'Attach Connection To This File', and 'Command Palette... Ctrl+Shift+P'. The text 'Right Click & Select Run MySQL Query' is overlaid on the editor in yellow.

## Examples :

After creating our company database we need to create the tables

### Employee Table:

```
1 USE COMPANY;  
2  
3  
4 CREATE TABLE EMPLOYEE  
5 (  
6     EMPLOYEE_ID INT AUTO_INCREMENT PRIMARY KEY,  
7     NAME VARCHAR(255) NOT NULL,  
8     ADDRESS VARCHAR(500),  
9     CITY VARCHAR(255),  
10    STATE VARCHAR(255)  
11 );  
12
```

**USE COMPANY ;**

This SQL statement is used to select the "COMPANY" database

**EMPLOYEE\_ID INT AUTO\_INCREMENT PRIMARY KEY :**

This column is named "**EMPLOYEE\_ID.**" It is of the **integer** data type (**INT**) and is defined as an **auto-incrementing primary key.**

**NAME VARCHAR(255) NOT NULL:**

This column is named "**NAME**" with data type (**VARCHAR**). It can store text values with a maximum length of 255 characters. The "**NOT NULL**" constraint specifies that this column must always have a value; it cannot be left empty (null).

# One-to-One Example using SQL

queries.sql



3zvgp366v

NEW

MYSQL

RUN



```
1
2 -- create
3 CREATE TABLE EMPLOYEE (
4   EMPLOYEE_id INTEGER PRIMARY KEY,
5   name varchar(255) NOT NULL,
6   ADDRESS varchar(500) NOT NULL,
7   CITY VARCHAR(255)
8 );
9 CREATE TABLE PAYROLL (
10  PAYROLL_ID INT PRIMARY KEY,
11  SALARY INT NOT NULL,
12  POSITION VARCHAR(255) NOT NULL,
13  EMPLOYEE_ID INT UNIQUE REFERENCES EMPLOYEE(EMPLOYEE_id)
14 );
15
```

**1 to 1 Relation**



STDIN

Input for the program ( Optional)



Output:

Program did not output anything!

# On update & on delete using SQL

queries.sql



3zvqp366v

NEW

MYSQL ▾

RUN ▶



```
1
2 -- create
3 CREATE TABLE EMPLOYEE (
4   EMPLOYEE_id INTEGER PRIMARY KEY,
5   name varchar(255) NOT NULL,
6   ADDRESS varchar(500) NOT NULL,
7   CITY VARCHAR(255)
8 );
9 CREATE TABLE PAYROLL (
10  PAYROLL_ID INT PRIMARY KEY,
11  SALARY INT NOT NULL,
12  POSITION VARCHAR(255) NOT NULL,
13  EMPLOYEE_ID INT UNIQUE REFERENCES EMPLOYEE(EMPLOYEE_id)
14  ON UPDATE CASCADE
15  ON DELETE CASCADE
16 );
17
```

STDIN

Input for the program (Optional)



Output:

Program did not output anything!

# Example using SQL

queries.sql



3zvgp366v

NEW

MYSQL

RUN



```
1
2 -- create
3 CREATE TABLE EMPLOYEE (
4   EMPLOYEE_id INTEGER PRIMARY KEY,
5   name varchar(255) NOT NULL,
6   ADDRESS varchar(500) NOT NULL,
7   CITY VARCHAR(255)
8 );
9 CREATE TABLE PAYROLL (
10  PAYROLL_ID INT PRIMARY KEY,
11  SALARY INT NOT NULL,
12  POSITION VARCHAR(255) NOT NULL,
13  EMPLOYEE_ID INT UNIQUE REFERENCES EMPLOYEE(EMPLOYEE_id)
14  ON UPDATE CASCADE
15  ON DELETE CASCADE
16 );
17
18
19 insert into EMPLOYEE
20 (EMPLOYEE_id, name, ADDRESS)
21 values(1, "mohamed", "alexandria"),
22 (2, "ahmed", "alexandria");
23
24 select * from EMPLOYEE;
```

STDIN

Input for the program ( Optional )



Output:

EMPLOYEE_id	name	ADDRESS	CITY
1	mohamed	alexandria	NULL
2	ahmed	alexandria	NULL

# Example using SQL

queries.sql



3zvgp366v

NEW

MYSQL ▾

RUN ▶

```
1
2 -- create
3 CREATE TABLE EMPLOYEE (
4   EMPLOYEE_id INTEGER PRIMARY KEY,
5   name varchar(255) NOT NULL,
6   ADDRESS varchar(500) NOT NULL,
7   CITY VARCHAR(255)
8 );
9 CREATE TABLE PAYROLL (
10  PAYROLL_ID INT PRIMARY KEY,
11  SALARY INT NOT NULL,
12  POSITION VARCHAR(255) NOT NULL,
13  -- PersonID int FOREIGN KEY REFERENCES Persons(PersonID),
14  EMPLOYEE_ID INT UNIQUE REFERENCES EMPLOYEE(EMPLOYEE_id)
15  ON UPDATE CASCADE
16  ON DELETE CASCADE
17 );
18
19
20 insert into EMPLOYEE
21 (EMPLOYEE_id, name, ADDRESS)
22 values(1, "mohamed", "alexandria"),
23 (2, "ahmed", "alexandria");
24
25 select * from EMPLOYEE;
26
27
28 insert into payroll
29 (PAYROLL_ID, SALARY, POSITION, EMPLOYEE_ID)
30 values (1, 1000, "developer", 1), (2, 3000, "anthing", 2);
31
32 select * from payroll;
33
```

STDIN

Input for the program ( Optional)

Output:

EMPLOYEE_id	name	ADDRESS	CITY
1	mohamed	alexandria	NULL
2	ahmed	alexandria	NULL

  

PAYROLL_ID	SALARY	POSITION	EMPLOYEE_ID
1	1000	developer	1
2	3000	anthing	2

# Example using SQL

queries.sql + 3zvgp366v NEW MYSQL RUN

```
1
2 -- create
3 CREATE TABLE EMPLOYEE (
4   EMPLOYEE_id INTEGER AUTO_INCREMENT PRIMARY KEY,
5   name varchar(255) NOT NULL,
6   ADDRESS varchar(500) NOT NULL,
7   CITY VARCHAR(255)
8 );
9 CREATE TABLE PAYROLL (
0   PAYROLL_ID INT AUTO_INCREMENT PRIMARY KEY,
1   SALARY INT NOT NULL,
2   POSITION VARCHAR(255) NOT NULL,
3   EMPLOYEE_ID INT UNIQUE REFERENCES EMPLOYEE(EMPLOYEE_id)
4   ON UPDATE CASCADE
5   ON DELETE CASCADE
6 );
7
8
9 insert into EMPLOYEE
0 (name, ADDRESS)
1 values("mohamed", "alexandria"),
2 ("ahmed", "alexandria");
3
4 select * from EMPLOYEE;
5
6
7 insert into payroll
8 (SALARY, POSITION, EMPLOYEE_ID)
9 values (1000, "developer", 1), (3000, "anthing", 2);
0
1 select * from payroll;
```

STDIN

Input for the program (Optional)

---

Output:

EMPLOYEE_id	name	ADDRESS	CITY
1	mohamed	alexandria	NULL
2	ahmed	alexandria	NULL

  

PAYROLL_ID	SALARY	POSITION	EMPLOYEE_ID
1	1000	developer	1
2	3000	anthing	2

# One to many Example using SQL

queries.sql



3zvgp366v

NEW

MYSQL

RUN

```
1 CREATE TABLE CUSTOMERS
2 (
3   CUSTOMER_ID INT AUTO_INCREMENT PRIMARY KEY,
4   CUSTOMER_NAME VARCHAR(255) NOT NULL,
5   PHONE_NUMBER VARCHAR(255) NOT NULL,
6   ADDRESS VARCHAR(500)
7 );
8
9 CREATE TABLE ORDERS
10 (
11   ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
12   ORDER_DATE DATE,
13   COST INT NOT NULL,
14   CUSTOMER_ID INT NOT NULL,
15   foreign KEY (CUSTOMER_ID) references CUSTOMERS(CUSTOMER_ID)
16 );
```

**1 to many Relation**

STDIN

Input for the program (Optional)

Output:

Program did not output anything!

```
1 CREATE TABLE CUSTOMERS
2 (
3   CUSTOMER_ID INT AUTO_INCREMENT PRIMARY KEY,
4   CUSTOMER_NAME VARCHAR(255) NOT NULL,
5   PHONE_NUMBER VARCHAR(255) NOT NULL,
6   ADDRESS VARCHAR(500)
7 );
8
9 CREATE TABLE ORDERS
10 (
11   ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
12   ORDER_DATE DATE,
13   COST INT NOT NULL,
14   CUSTOMER_ID INT NOT NULL,
15   foreign KEY (CUSTOMER_ID) references CUSTOMERS(CUSTOMER_ID)
16 );
17
18 INSERT INTO CUSTOMERS
19 (CUSTOMER_NAME, PHONE_NUMBER, ADDRESS)
20 VALUES ("MOHAMED", "+2011111111", "ALEXANDRIA"),
21         ("AHMED", "+20123333333", "ALEXANDRIA");
22
23 INSERT INTO ORDERS
24 (ORDER_DATE, COST, CUSTOMER_ID)
25 VALUES ("2023-12-4", 4000, 1), ("2023-12-5", 5000, 2);
26
27
28 SELECT * FROM CUSTOMERS;
29 SELECT * FROM ORDERS;
```

STDIN

Input for the program (Optional)

Output:

CUSTOMER_ID	CUSTOMER_NAME	PHONE_NUMBER	ADDRESS
1	MOHAMED	+2011111111	ALEXANDRIA
2	AHMED	+20123333333	ALEXANDRIA

  

ORDER_ID	ORDER_DATE	COST	CUSTOMER_ID
1	2023-12-04	4000	1
2	2023-12-05	5000	2

*Thank  
you*

