Lecture (6) "Introduction to Data Base"

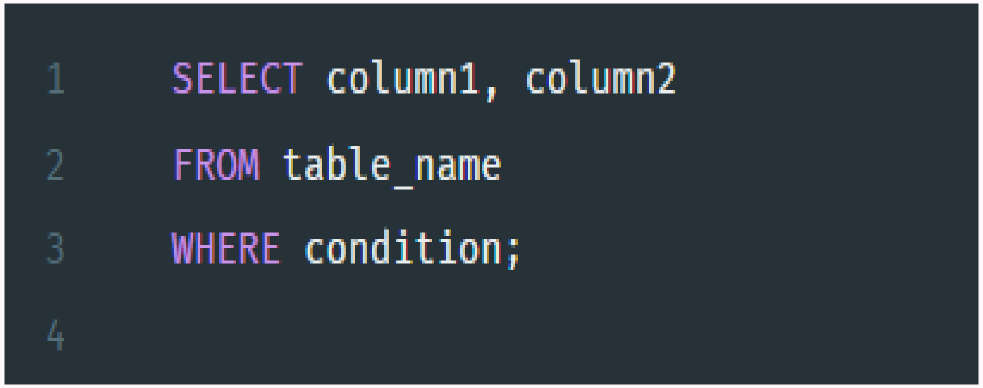
Presented by : Dr. Nehal El Azaly

Introduction to SQL for Beginners

1. Data Query Language (DQL) :

1. **SELECT :** The SELECT statement is used to select data from a database.

Syntax :



```
SELECT column1, column2
FROM table_name
WHERE condition;
```

column1, column2 are the *field names* of the table
you want to select data from.

The table_name represents the name of the table you want to select data from.

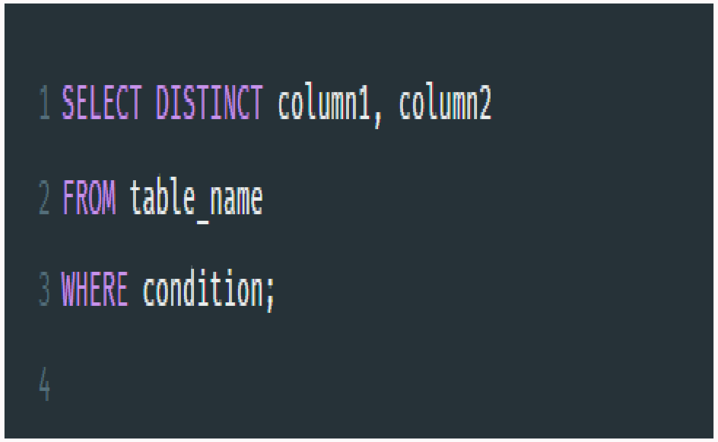
" WHERE condition " is optional and allows you to specify a condition for filtering the results.

1. Data Query Language (DQL) :

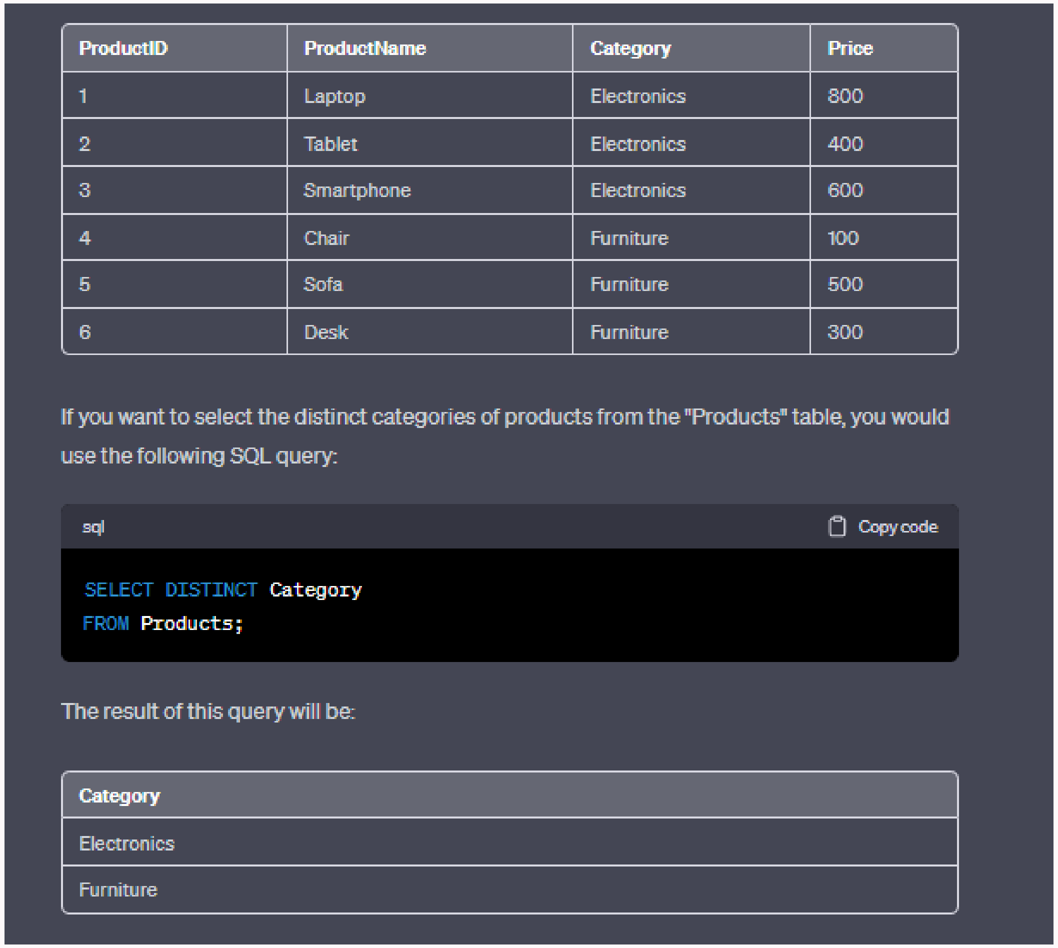
1. SELECT DISTINCT :

The SELECT DISTINCT “ Unique Values ” statement is used to return only distinct (Unique) values.

Syntax :



```sql
SELECT DISTINCT column1, column2
FROM table_name
WHERE condition;
```



ProductID	ProductName	Category	Price
1	Laptop	Electronics	800
2	Tablet	Electronics	400
3	Smartphone	Electronics	600
4	Chair	Furniture	100
5	Sofa	Furniture	500
6	Desk	Furniture	300

If you want to select the distinct categories of products from the "Products" table, you would use the following SQL query:

```sql
SELECT DISTINCT Category
FROM Products;
```

The result of this query will be:

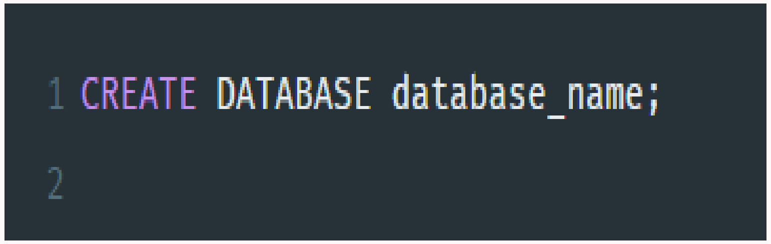
Category
Electronics
Furniture

2. Data Definition Language (DDL) :

1. CREATE DATABASE :

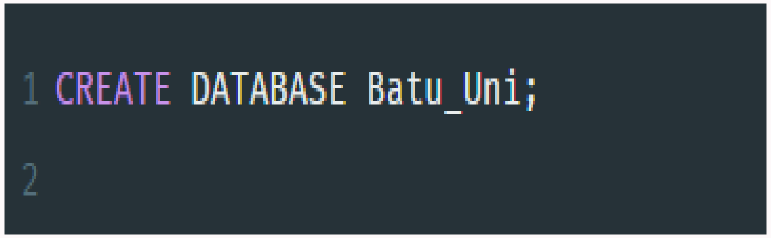
The CREATE DATABASE statement is used to create a new SQL database.

Syntax :



```
CREATE DATABASE database_name;
```

Example :



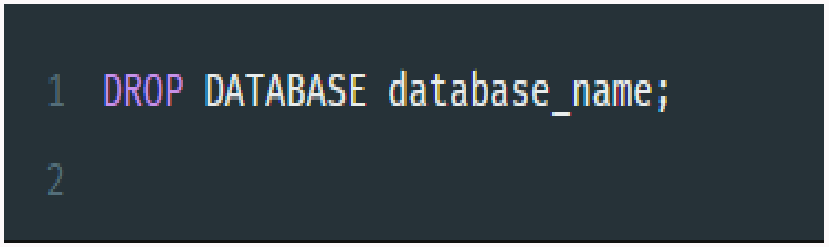
```
CREATE DATABASE Batu_Uni;
```

2. Data Definition Language (DDL) :

2. DROP DATABASE :

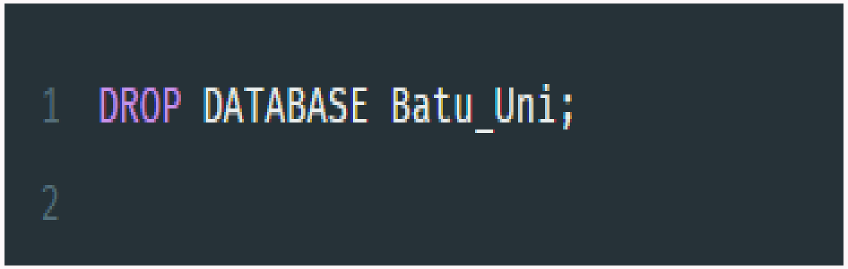
The DROP DATABASE statement is used to DELETE a SQL database.

Syntax :



```
DROP DATABASE database_name;
```

Example :



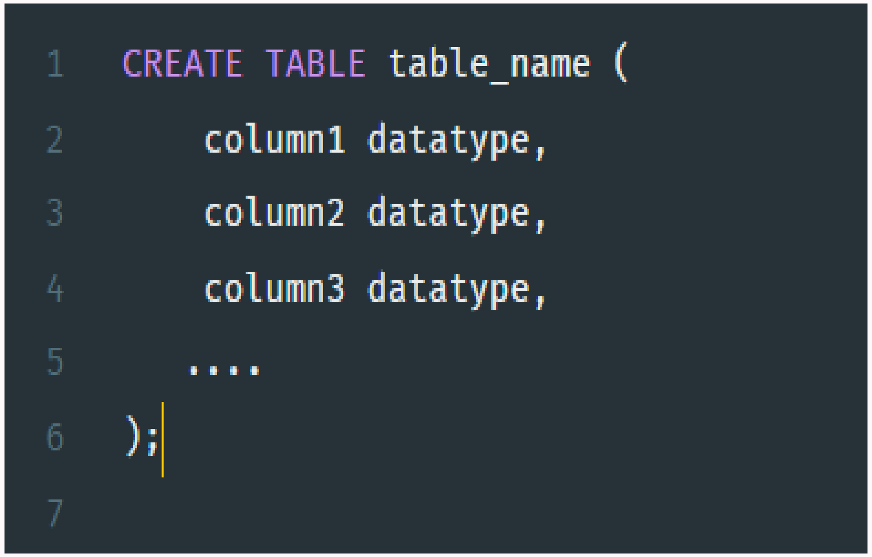
```
DROP DATABASE Batu_Uni;
```

2. Data Definition Language (DDL) :

3. CREATE TABLE :

CREATE TABLE statement is used to create a new table in a database.

Syntax :



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

The table_name represents the name of the table you want to create.

column1, column2 are the *field names* of the table
you want to create in the table.

The datatype parameter specifies the type of data the column can hold (varchar, integer, date, etc.).

2. Data Definition Language (DDL) :

3. CREATE TABLE :

Commonly used SQL data types:

INTEGER (INT): Used for whole numbers.

VARCHAR: Used for variable-length character strings.

DATE: Used for date values.

DATETIME: Used for date and time values.

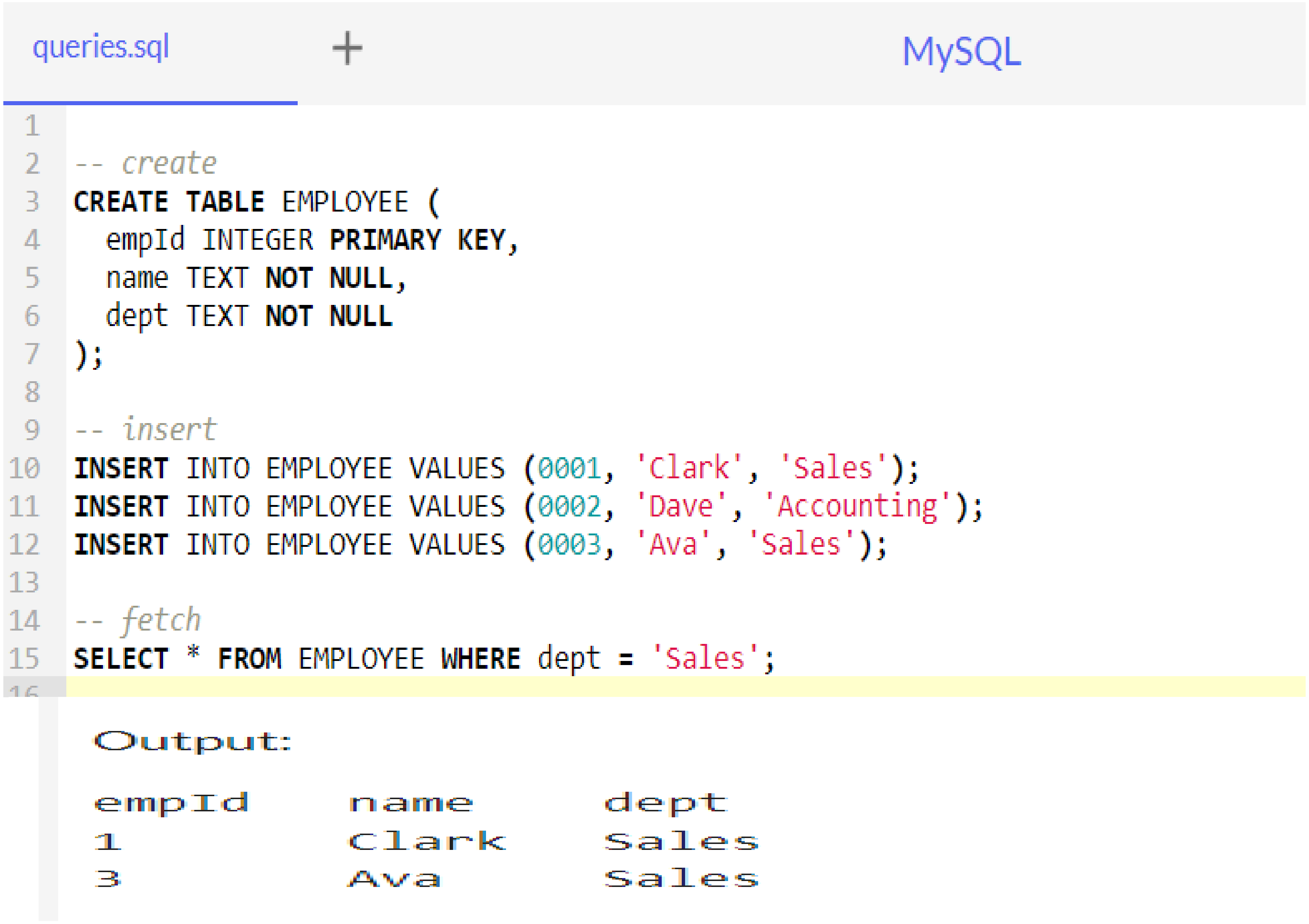
DECIMAL (NUMERIC): Used for fixed-point numbers with a specified precision and scale.

BOOLEAN (**TINYINT** or **BOOL**): Used for true or false values.

TEXT: Used for longer text strings with no predefined length.

CHAR: Used for fixed-length character strings.

FLOAT: Used for approximate numeric values.


queries.sql + MySQL

```sql

-- create
CREATE TABLE EMPLOYEE (
  empId INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  dept TEXT NOT NULL
);

-- insert
INSERT INTO EMPLOYEE VALUES (0001, 'Clark', 'Sales');
INSERT INTO EMPLOYEE VALUES (0002, 'Dave', 'Accounting');
INSERT INTO EMPLOYEE VALUES (0003, 'Ava', 'Sales');

-- fetch
SELECT * FROM EMPLOYEE WHERE dept = 'Sales';
```

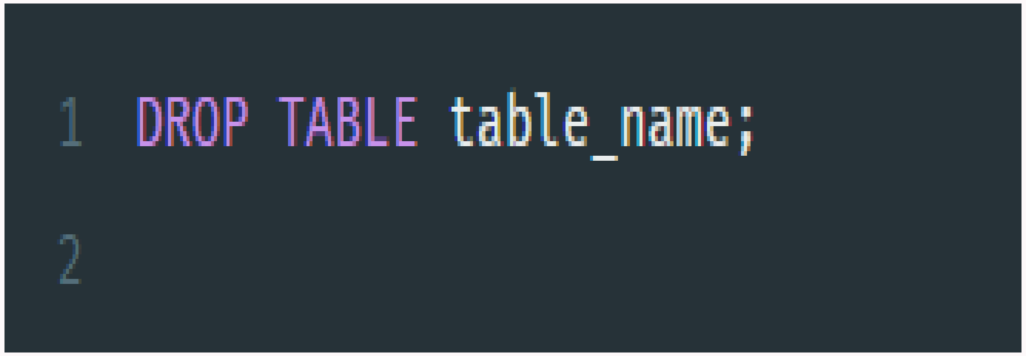
```
Output:

empId     name      dept
1         Clark     Sales
3         Ava       Sales
```

2. Data Definition Language (DDL) :

4. DROP TABLE :

DROP TABLE statement is used to delete a table in a database.

Syntax :



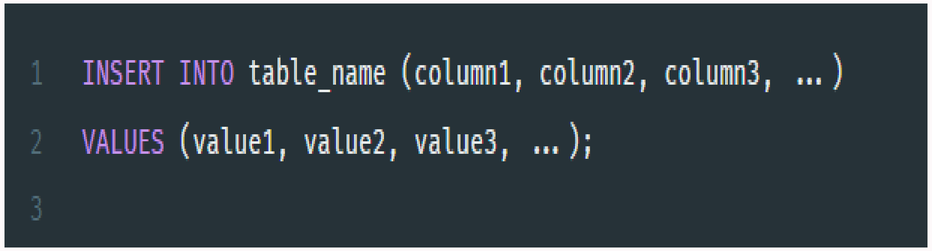
```sql
DROP TABLE table_name;
```

3. Data Manipulation Language (DML) :

Data Manipulation Language is used to manipulate data stored in the database.

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

Syntax :



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

table_name: Replace this with the name of the table where you want to insert data.

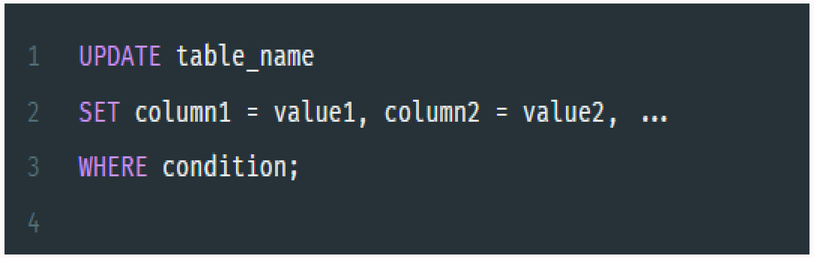
(column1, column2, column3, ...): you list the columns into which you want to insert data.

VALUES (value1, value2, value3, ...): the values that you want to insert into the specified columns. The values should be in the same order as the columns.

3. Data Manipulation Language (DML) :

2. The UPDATE statement is used to insert new records in a table.

Syntax :



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

table_name: Replace this with the name of the table where you want to update data.

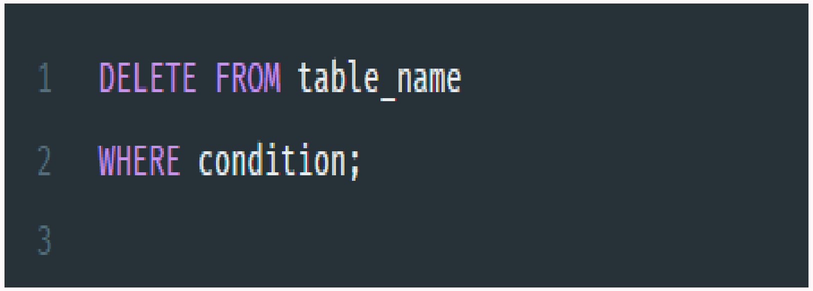
SET column1 = value1, column2 = value2, ...: you specify the columns you want to update and
the new values you want to set. Each column is paired with its new value. You can update one or multiple columns in a single UPDATE statement.

WHERE condition: This part is optional but recommended. It allows you to specify a condition that determines
which rows in the table should be updated. If you omit the WHERE clause, all rows in the table will be updated.

3. Data Manipulation Language (DML) :

3. The DELETE statement is used to delete records from table.

Syntax :



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

table_name: Replace this with the name of the table where you want to delete data.

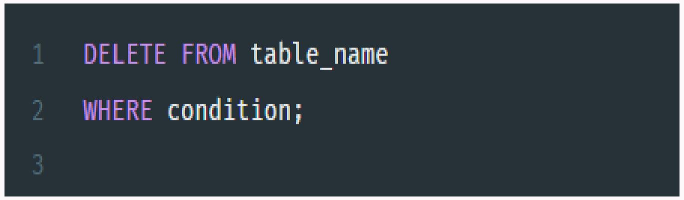
SET column1 = value1, column2 = value2, ...:
you specify the columns you want to update and
the new values you want to set. Each column is paired with its new value. You can update one or multiple columns in a single UPDATE statement.

WHERE condition: This part is optional but recommended. It allows you to specify a condition that determines
which rows in the table should be deleted.
If you omit the WHERE clause, all rows in the table will be deleted.

3. Data Manipulation Language (DML) :

3. The DELETE statement is used to delete records from table.

Syntax :



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

table_name: Replace this with the name of the table where you want to delete data.

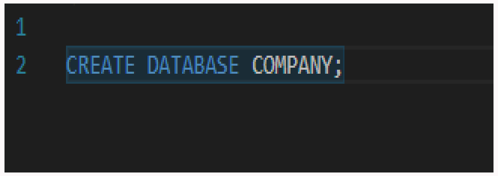
SET column1 = value1, column2 = value2, ...: you specify the columns you want to update and
the new values you want to set. Each column is paired with its new value. You can update one or multiple columns in a single UPDATE statement.

WHERE condition: This part is optional but recommended. It allows you to specify a condition that determines
which rows in the table should be deleted. If you omit the WHERE clause, all rows in the table will be deleted.

Examples :

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:

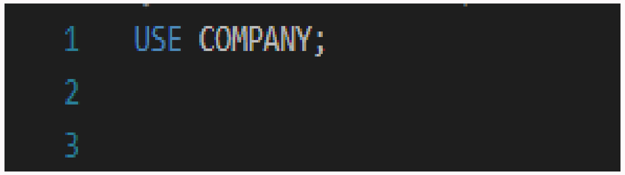


```
CREATE DATABASE COMPANY;
```

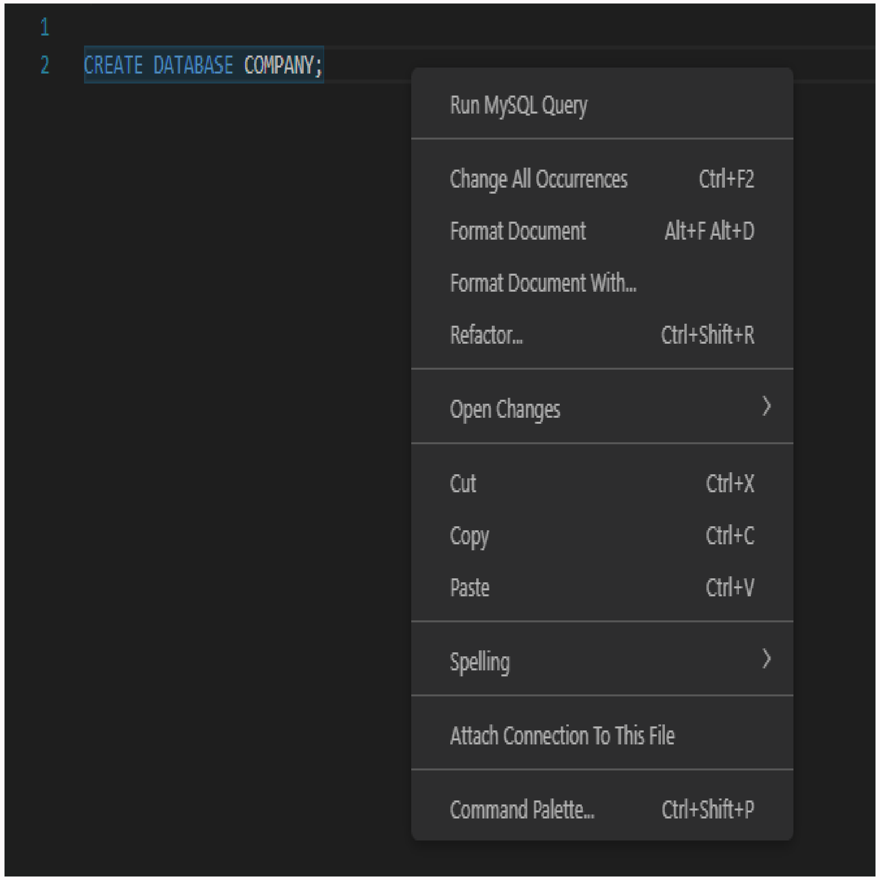
Right Click
&
Select Run MySQL Query

To use Database:

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



```
USE COMPANY;
```



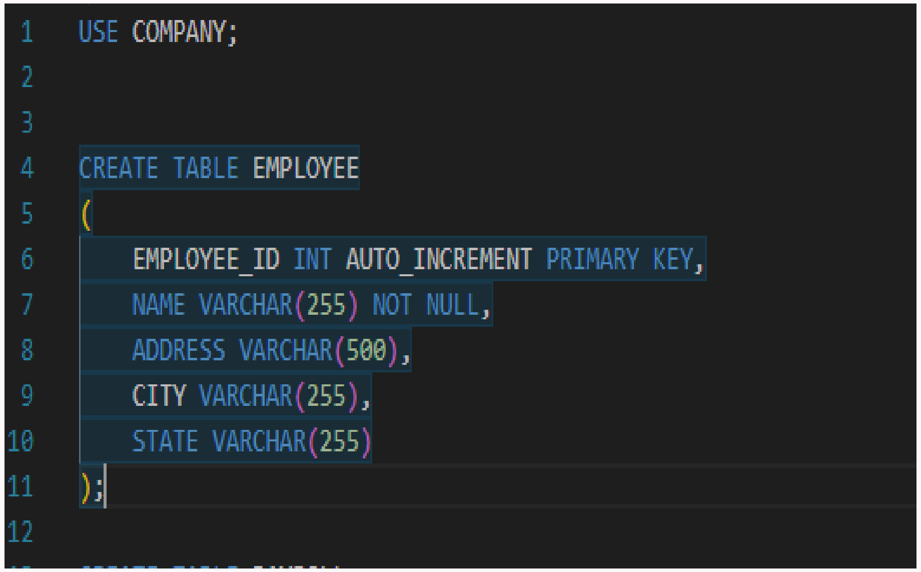
```
CREATE DATABASE COMPANY;
```

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
Command Palette... Ctrl+Shift+P

Examples :

After creating our company database we need to create the tables

Employee Table:



```sql
USE COMPANY;


CREATE TABLE EMPLOYEE
(
    EMPLOYEE_ID INT AUTO_INCREMENT PRIMARY KEY,
    NAME VARCHAR(255) NOT NULL,
    ADDRESS VARCHAR(500),
    CITY VARCHAR(255),
    STATE VARCHAR(255)
);
```

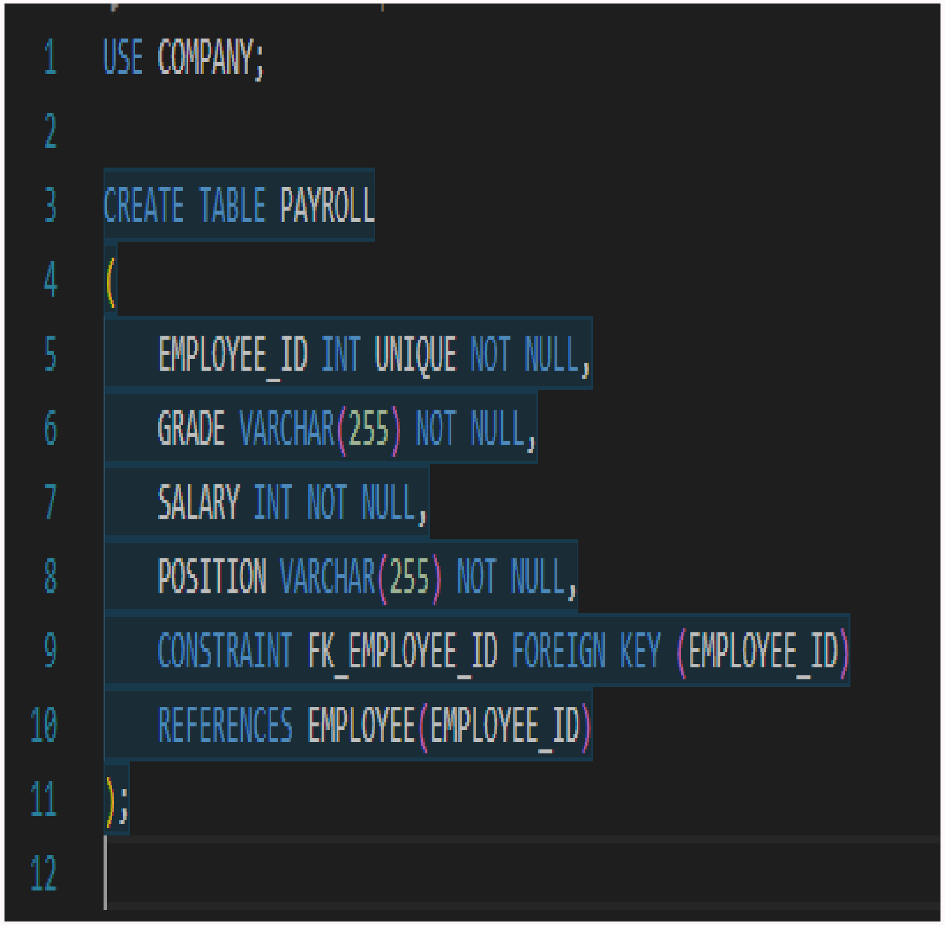
USE COMPANY ;
This SQL statement is used to select the "COMPANY" database

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).

Examples :

Payroll Table:



```sql
USE COMPANY;

CREATE TABLE PAYROLL
(
    EMPLOYEE_ID INT UNIQUE NOT NULL,
    GRADE VARCHAR(255) NOT NULL,
    SALARY INT NOT NULL,
    POSITION VARCHAR(255) NOT NULL,
    CONSTRAINT FK_EMPLOYEE_ID FOREIGN KEY (EMPLOYEE_ID)
    REFERENCES EMPLOYEE(EMPLOYEE_ID)
);
```

SQL

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

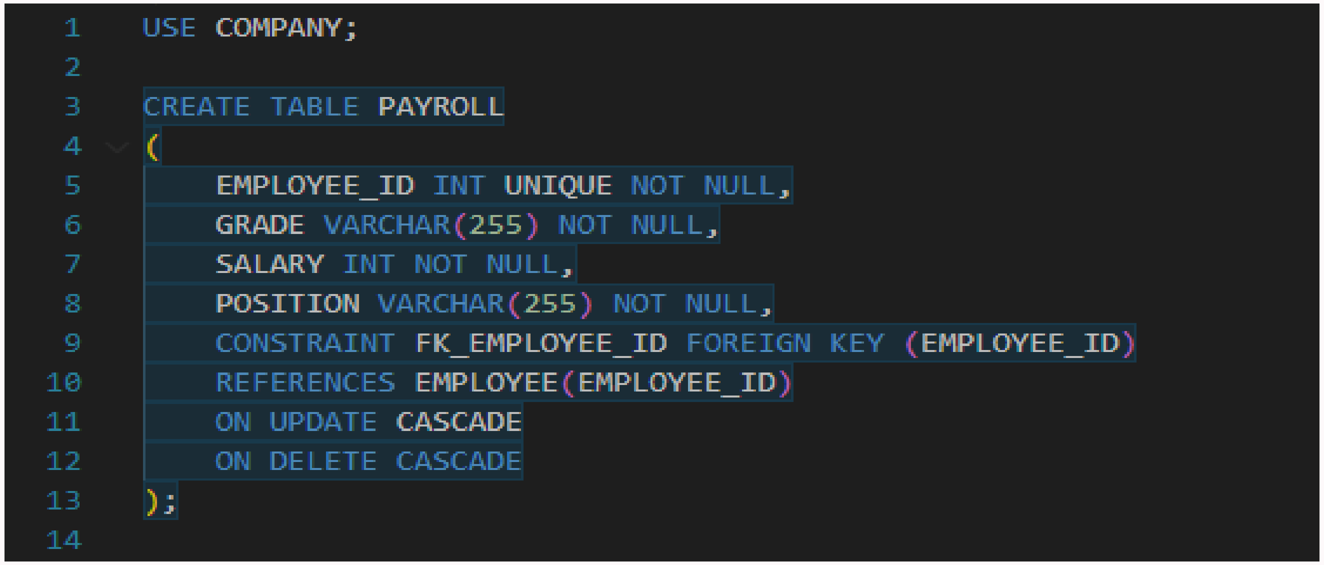
EMPLOYEE_ID INT UNIQUE NOT NULL: This column is named "EMPLOYEE_ID" and is of the integer data type (INT). It is marked as both "UNIQUE" and "NOT NULL."

The "UNIQUE" constraint ensures that each value in this column must be unique across all records in the table.

The "NOT NULL" constraint specifies that a value for this column is required.

Examples :

Payroll Table:



```sql
USE COMPANY;

CREATE TABLE PAYROLL
(
    EMPLOYEE_ID INT UNIQUE NOT NULL,
    GRADE VARCHAR(255) NOT NULL,
    SALARY INT NOT NULL,
    POSITION VARCHAR(255) NOT NULL,
    CONSTRAINT FK_EMPLOYEE_ID FOREIGN KEY (EMPLOYEE_ID)
    REFERENCES EMPLOYEE(EMPLOYEE_ID)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

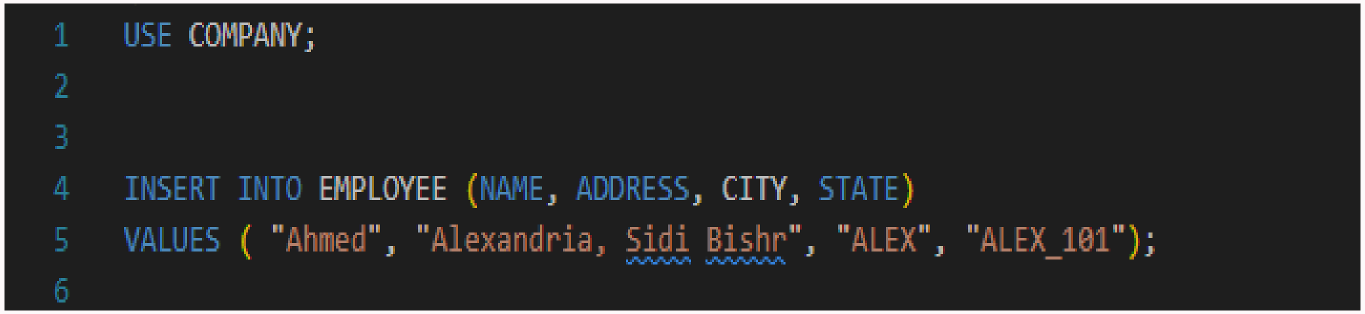
CONSTRAINT FK_EMPLOYEE_ID FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEE(EMPLOYEE_ID):
This line defines a foreign key constraint named "FK_EMPLOYEE_ID" on the "EMPLOYEE_ID" column.
It establishes a relationship between the "PAYROLL" table and the "EMPLOYEE" table.
This means that the "EMPLOYEE_ID" values in the "PAYROLL" table must correspond to the "EMPLOYEE_ID" values in the "EMPLOYEE" table. The "REFERENCES" clause specifies the referenced table and column.

ON UPDATE CASCADE : if I update anything related to Employee it will be updated in payroll table
ON DELETE CASCADE : if I delete employee record from employee table , every record related to this employee will be deleted in payroll

Examples :

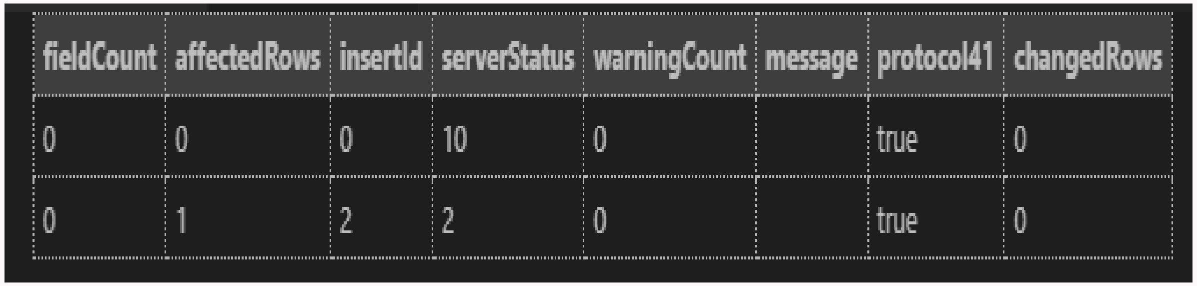
After creating our tables in the company database we need to insert the data

Inserting Data into Employee Table:



```sql
USE COMPANY;


INSERT INTO EMPLOYEE (NAME, ADDRESS, CITY, STATE)
VALUES ( "Ahmed", "Alexandria, Sidi Bishr", "ALEX", "ALEX_101");

```

Output:



fieldCount	affectedRows	insertId	serverStatus	warningCount	message	protocol41	changedRows
0	0	0	10	0		true	0
0	1	2	2	0		true	0

Affected Rows: $\longrightarrow$

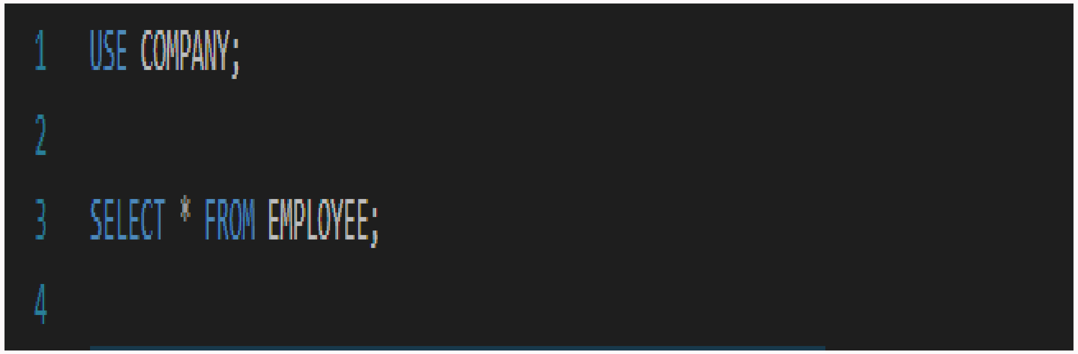
tells us how many rows affected by our statement

Examples :

Inserting Data into Payroll Table:

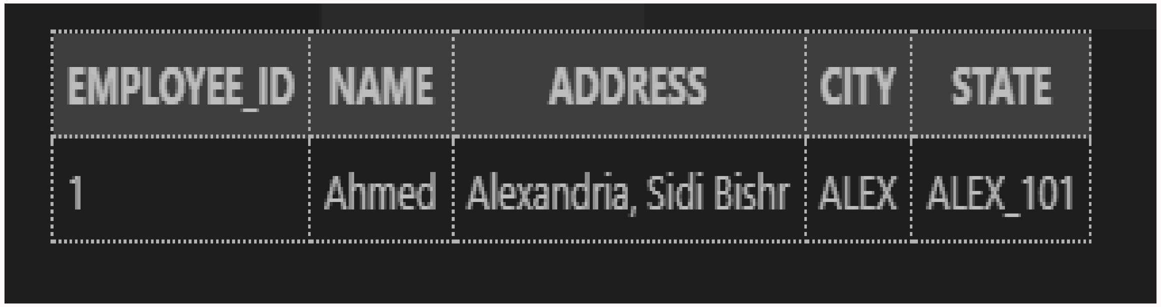
But first we need to know employee_id and this will be done by using SELECT STATEMENT

Retrieve data from employee table



```
USE COMPANY;

SELECT * FROM EMPLOYEE;

```

Output:

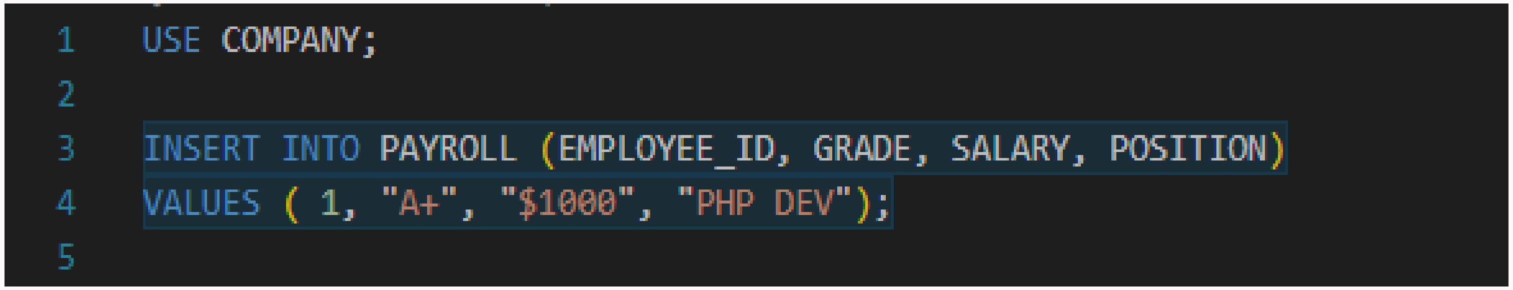


EMPLOYEE_ID	NAME	ADDRESS	CITY	STATE
1	Ahmed	Alexandria, Sidi Bishr	ALEX	ALEX_101

Examples :

After we got employee_id, now we can make a payroll record for him

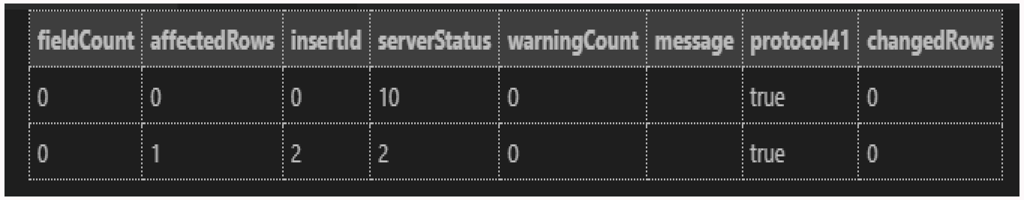
Inserting Data into Payroll Table:



```
USE COMPANY;

INSERT INTO PAYROLL (EMPLOYEE_ID, GRADE, SALARY, POSITION)
VALUES ( 1, "A+", "$1000", "PHP DEV");

```

Output:

Affected Rows:

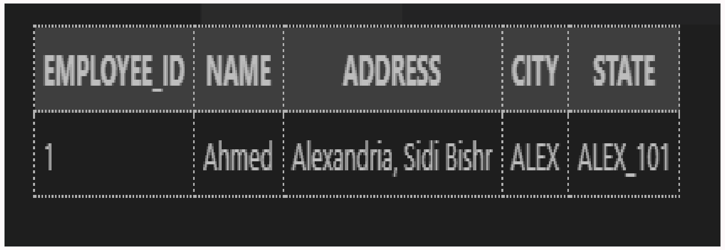


fieldCount	affectedRows	insertId	serverStatus	warningCount	message	protocol41	changedRows
0	0	0	10	0		true	0
0	1	2	2	0		true	0

tells us how many rows affected by our statement

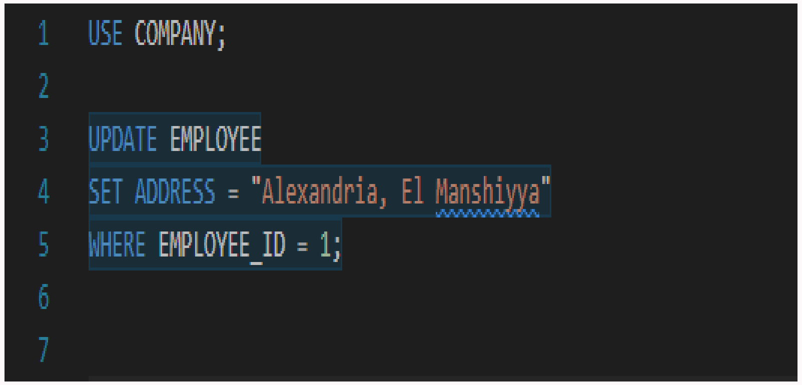
Examples :

Our Employee has moved to another apartment in El Manshiyya



EMPLOYEE_ID	NAME	ADDRESS	CITY	STATE
1	Ahmed	Alexandria, Sidi Bishr	ALEX	ALEX_101

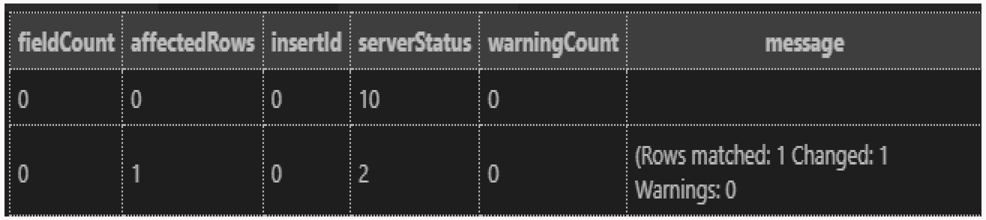
We need to change his ADDRESS and this will be done with UPDATE statement



```
USE COMPANY;

UPDATE EMPLOYEE
SET ADDRESS = "Alexandria, El Manshiyya"
WHERE EMPLOYEE_ID = 1;


```

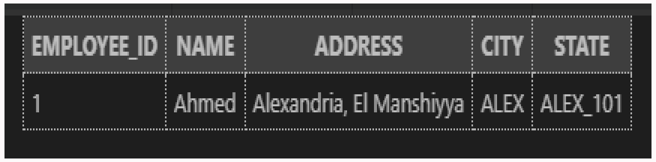
We used WHERE here to specify which record we will update it's address

Output:



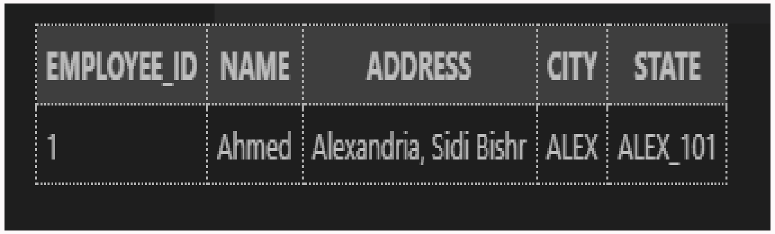
fieldCount	affectedRows	insertId	serverStatus	warningCount	message
0	0	0	10	0	
0	1	0	2	0	(Rows matched: 1 Changed: 1 Warnings: 0

Here I used SELECT to check if row changed successfully



EMPLOYEE_ID	NAME	ADDRESS	CITY	STATE
1	Ahmed	Alexandria, El Manshiyya	ALEX	ALEX_101

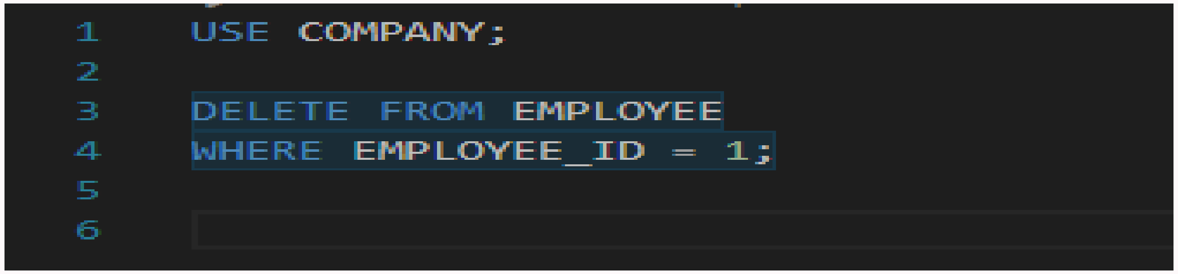
Examples :



EMPLOYEE_ID	NAME	ADDRESS	CITY	STATE
1	Ahmed	Alexandria, Sidi Bishr	ALEX	ALEX_101

Our Employee quit and went to work with another Company

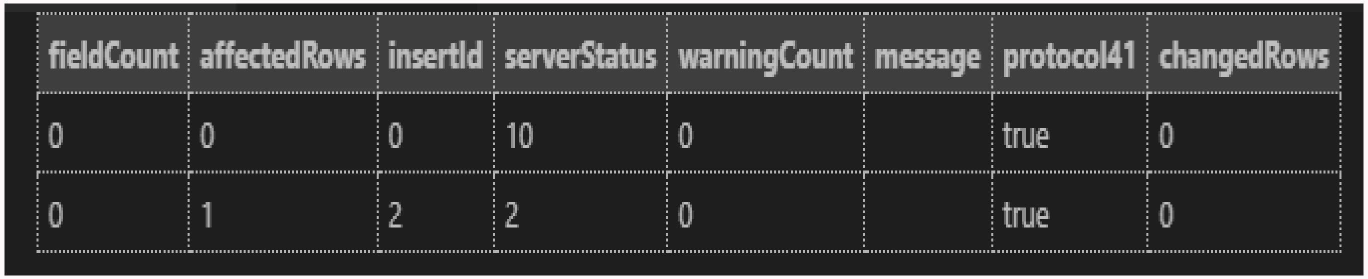
We don't need his data anymore
so we delete him from our records and this will be done with DELETE FROM



```
USE COMPANY;

DELETE FROM EMPLOYEE
WHERE EMPLOYEE_ID = 1;

```

Output:

Affected Rows:



fieldCount	affectedRows	insertId	serverStatus	warningCount	message	protocol41	changedRows
0	0	0	10	0		true	0
0	1	2	2	0		true	0

tells us how many rows affected by our statement

Thank you