Teach Computer Science

Structured query language

Lesson Objectives

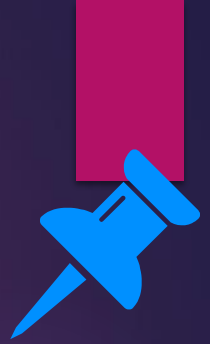
Students will learn about:

- How SQL works?
- What are the different queries available to work with the database?
- Operators used in queries to filter results

1.

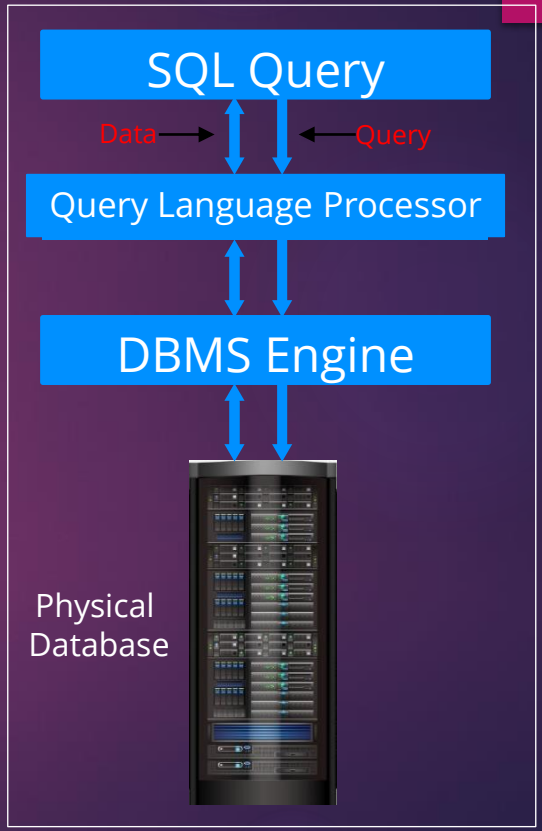
Content

Structured Query Language (SQL)



- SQL is a programming language used to access and manage databases.
- The SQL statements use simple language and are easy to understand and remember.

- SQL queries are processed using a Query Language Processor.
- The Query Processor is a Structured Query Language (SQL) parser, optimiser and query execution engine.
- The Query Processor accepts and executes SQL commands according to a chosen plan and interacts with the DBMS engine to return the expected results.
- The DBMS engine is the underlying software component that a database management system (DBMS) uses to create, read, update and delete data from a physical database.



Operators

- Query languages use arithmetic and Boolean operators to filter results and to manage the information.
- Arithmetic operators are given in the table

Operator	Function
=	Equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<>	Not equal to

Operators

- Boolean operators are used to filter data in a database according to a few conditions.

Operator	Example	Function
AND	(condition 1) AND (condition 2)	Filters data that satisfies both the conditions
OR	(condition 1) OR (condition 2)	Filters data that satisfies either one of the conditions
NOT	NOT (condition 1)	Filters data that doesn't satisfy the condition

Example:

Table with table name “Employeelist”

Employee ID	Name	Department	Date_of_joining	Age
1	David Watson	Engineering	2024-03-28	45
2	Helen Williams	Human Resource	2023-04-30	30
3	Paul Ingram	Research and development	2016-05-14	25
4	Rosemarie Gallagher	Engineering	2017-09-07	24
5	David Rayner	Research and development	2021-12-11	42
6	Alan Cross	Human Resource	2020-12-12	32

SELECT query

- The syntax for a query in SQL is:

  SELECT field_names FROM table_name WHERE condition

- A * symbol is used for field_names if all columns are to be displayed.

The query to find employees in ‘Engineering’ and ‘Research and development’ department is:

- SELECT * FROM Employeelist WHERE Department = ‘Engineering’ OR ‘Research and development’;

Result for the query:

SELECT * FROM Employeelist WHERE Department = ‘Engineering’ OR ‘Research and development’;

Employee ID	Fullname	Department	Date_of_joining	Age
1	David Watson	Engineering	2024-03-28	45
3	Paul Ingram	Research and development	2016-05-14	25
4	Rosemarie Gallagher	Engineering	2017-09-07	24
5	David Rayner	Research and development	2021-12-11	42

Query to find employees working in Engineering department of age less than 40 years is:

SELECT * FROM Employeelist WHERE Age<40 AND Department= ‘Engineering’;

Result for the query:

Employee ID	Fullname	Department	Date of joining	Age
4	Rosemarie Gallagher	Engineering	2017-09-07	24

Creating a table

- The statement for creating a table in SQL is CREATE TABLE.
- The syntax for a CREATE TABLE statement is:

CREATE TABLE table_name(column1_name datatype, column2_name datatype,....);

- The query for creating Empolyeedetails is:

CREATE TABLE Employeedetails (Employee ID int(3), Name varchar(20), Department varchar(30), Date of joining date, Age int(2));

```
CREATE TABLE Employee_details_V1(
Employee_ID INTEGER Primary Key,
Fullname CHAR(25),
Department CHAR(30),
Date_joined DATE,
Age INTEGER
);
```

Creating a table

- ‘Int’ is used to represent integers.
- ‘Varchar’ allows only character strings.
- The numbers in the brackets represent the maximum number of digits a data type can hold.
- Date is entered in the format YYYY-MM-DD.

```
CREATE TABLE Employeedetails
(Employee ID int(3), Name varchar(20),
Department varchar(30), Date of
joining date, Age int(2) );
```

Employee ID	Name	Department	Date of joining	Age
1	David Watson	Engineering	2024-03-28	45

CREATING A TABLE

Creating a table

Syntax:
CREATE TABLE <table name>
(
<field name> <datatype> <attribute>,
<field name> <datatype> <attribute>,
<field name> <datatype> < attribute >,
<field name> <datatype> < attribute >,
<field name> <datatype> < attribute >
)

Creating a table

Syntax:
CREATE TABLE <table name>
(
<field name> <datatype> <attribute>,
<field name> <datatype> <attribute>,
<field name> <datatype> < attribute >,
<field name> <datatype> < attribute >,
<field name> <datatype> < attribute >
)

Create a TABLE with the following parameters/attributes

Field Name	Datatype	Attribute
Employee_ID	INTEGER	
Lastname	TEXT	50
Firstname	TEXT	50
Phone	TEXT	10
Email	TEXT	50

Creating a table

Example:

```
CREATE TABLE Employee_List (
Employee_ID INTEGER,
LASTNAME TEXT (50),
FIRSTNAME TEXT (50),
Phone TEXT (10),
Email TEXT (50))
```

Dropping/Removing a table

Dropping/Removing a table

Syntax: DROP TABLE <table name>

DROP TABLE Employee_list

INSERT INTO statement

INSERT INTO statement

- To add new records, an INSERT INTO statement is used.
- The values are separated by commas and the character strings are entered in the quotes (‘ ’).
- Numbers do not require quotes.
- To enter a new employee into the table Employeedetails, the following query is used.

INSERT INTO Employeedetails (EmployeeID, Name, Department, Dateofjoining, Age) VALUES (7, ‘Chris Sunley’, ‘Human Resource’, ‘2010-05-14’, 38);

```
INSERT INTO Employee_details(
Emp_number, Lastname, Firstname, Gender, Date_joined, Department, Salary)
VALUES (
'C00007', 'THANH', 'QUYNH', 'F', '4/3/2024', 'Management','2800');
```

INSERT INTO statement

INSERT INTO <tablename>(<Field 1>, <Field 2>, <Field 3>, <Field 4> <…>)
VALUES (‘data 1', ‘data2', ‘data3', ‘data4’, ‘…');

(‘0001', 'Bansag', ‘Fritz', '0919100001', 'eugene@febstar.com'),
(‘0002', 'Bansag', 'Anthony', '0919100002', 'anthony@febstar.com');

INSERT INTO statement

Insert the following values to the table: Employee List

Emp_ID	**00001**
Lastname	**Your lastname**
Firstname	**Your Firstname**
Phone	**Your Phone**
Email	**Your email**

(‘0008', 'Bansag', ‘Fritz', '0919100001', 'eugene@febstar.com'),

(‘0009', 'Bansag', 'Anthony', '0919100002', 'anthony@febstar.com');

INSERT INTO statement

- To add new records, an INSERT INTO statement is used.
- The values are separated by commas and the character strings are entered in the quotes (‘ ’).
- Numbers do not require quotes.
- To enter a new employee into the table Employeedetails, the following query is used.

INSERT INTO Employeedetails (EmployeeID, Name, Department, Dateofjoining, Age) VALUES (7, ‘Chris Sunley’, ‘Human Resource’, ‘2010-05-14’, 38);

Result for the query:

INSERT INTO Employeedetails (EmployeeID, Name, Department, Dateofjoining, Age) VALUES (7, ‘Chris Sunley’, ‘Human Resource’, ‘2010-05-14’, 38);

Employee ID	Name	Department	Date of joining	Age
1	David Watson	Engineering	2024-03-28	45
2	Helen Williams	Human Resource	2023-04-30	30
3	Paul Ingram	Research and development	2016-05-14	25
4	Rosemarie Gallagher	Engineering	2017-09-07	24
5	David Rayner	Research and development	2021-12-11	42
6	Alan Cross	Human Resource	2020-12-12	32
7	Chris Sunley	Human Resource	2020-05-14	38

DISPLAYING RESULTS

Displaying Results

- To display selected fields from a table the SELECT statement is used.
- To specify the field, the syntax used is: table_name.field_name
- The query to display names of employees and Department fields of employee table is:

SELECT Employeedetails.Name AND Employeedetails.Department FROM Employeedetails;

Result for the query:

SELECT Employeedetails.Name AND Employeedetails.Department
FROM Employeedetails;

Name	Department
David Watson	Engineering
Helen Williams	Human Resource
Paul Ingram	R&D
Rosemarie	Engineering
David Rayner	R&D
Alan Cross	Human Resource

Displaying Results

SYNTAX:

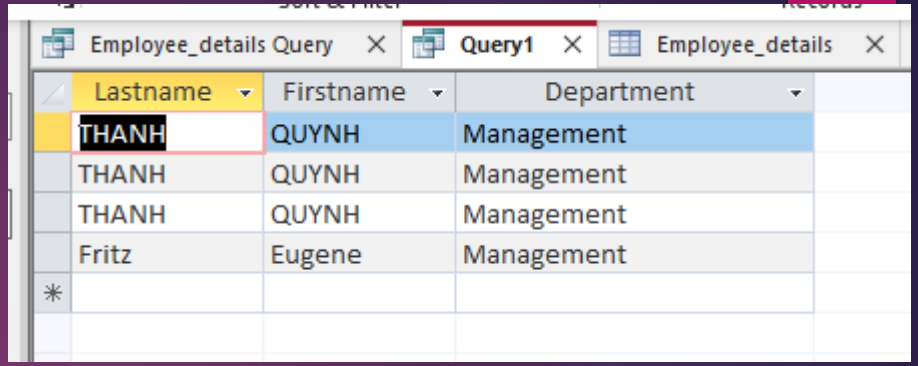
SELECT <Table name>.<Field> FROM <Table name> WHERE <Condition>

SELECT Employee_details.Lastname, Employee_details.Firstname FROM Employee_details WHERE Department = 'Management'

SELECT * FROM Employee_details WHERE Department = 'Management'

Displaying Results

Create an SQL command that will display the following output.

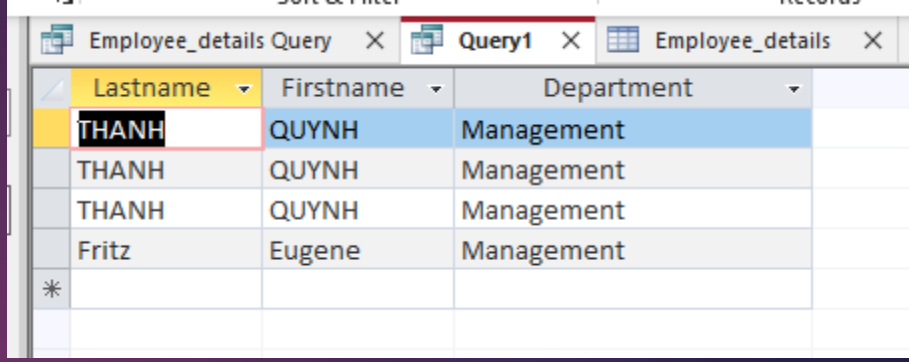


Lastname	Firstname	Department
THANH	QUYNH	Management
THANH	QUYNH	Management
THANH	QUYNH	Management
Fritz	Eugene	Management

Table name: Employee_details

ID	Emp_number	Lastname	Firstname	Gender	Date_joined	Department	Salary	Click to Add
1	C00001	Khanh	Tha	F	3/3/2024	Computer	$3,000.00	
2	C00002	Wu	Shang	M	2/13/2024	Finance	$3,500.00	
3	C00003	Fritz	Eugene	M	2/12/2024	Management	$6,000.00	
4	C00004	Do	Thu Ho	F	2/29/2024	HR	$3,000.00	
5	C00005	Amrisan	Cherry	F	4/5/2024	PR	$2,700.00	
6	C00006	Ardison	Bernardo	M	4/3/2024	Logistics	$2,900.00	
8	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
9	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
10	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
(New)							$0.00	

Displaying Results



SYNTAX:

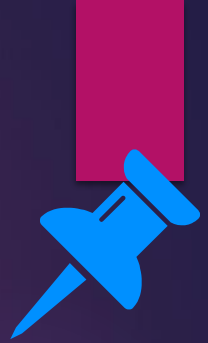
SELECT <Table name>.<Field> FROM <Table name> WHERE <Condition>

SELECT Employee_details.Lastname, Employee_details.Firstname FROM Employee_details.Department WHERE Department = 'Management'

Displaying all fields (*) with a criteria of Department is Management

SELECT * FROM Employee_details WHERE Department = 'Management'

Displaying Results



- The WHERE clause is used to select a few records according to the conditions specified.
- To display names of employees and Department fields of the employee table who joined in the year 2014, the following query is used.
- The year () function returns the value of year in the date.
- Year (Date of joining) returns the year in the date of joining fields.

SELECT Employeedetails.Name AND Employeedetails.Department FROM Employeedetails WHERE YEAR(Date of joining)=2024;

Result for the query:

```
SELECT Employeedetails.Name AND Employeedetails.Department
FROM Employeedetails WHERE YEAR(Date of joining)=2014;
```

Name	Department
David Watson	Engineering

UPDATING Function

UPDATE function

- For editing data in the table, such as field names and records, the UPDATE function is used.

UPDATE table_name SET column_name = value WHERE condition

UPDATE Employeedetails SET Employeedetails.Department= ‘Management’
WHERE Employeedetails.EmployeeID=7;

- The record with the EmployeeID as number 7 is updated. The department is updated as ‘Management’.

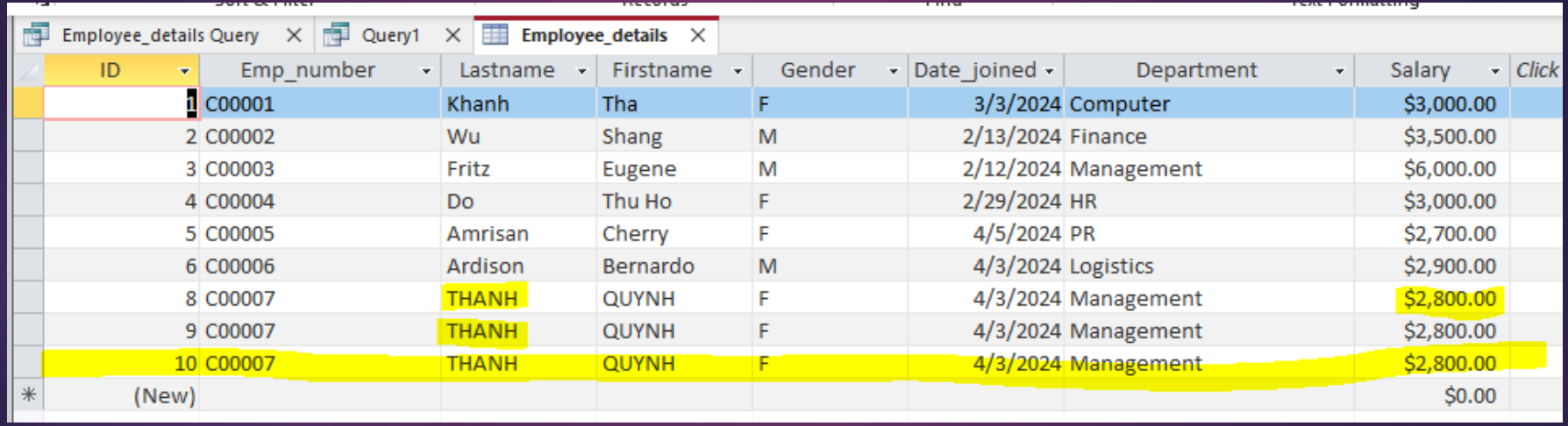
Result for the query:

UPDATE Employeedetails SET Employeedetails.Department= ‘Management’
WHERE Employeedetails.EmployeeID=7;

Employee ID	Name	Department	Date of joining	Age
1	David Watson	Engineering	2024-03-28	45
2	Helen Williams	Human Resource	2023-04-30	30
3	Paul Ingram	Research and development	2016-05-14	25
4	Rosemarie Gallagher	Engineering	2017-09-07	24
5	David Rayner	Research and development	2021-12-11	42
6	Alan Cross	Human Resource	2020-12-12	32
7	Chris Sunley	Management	2020-05-14	38

Update

Can you UPDATE the Record with ID number = 8 to its LASTNAME : **Thantra** and **Salary : 3600**



Employee_details Query | Query1 | Employee_details

ID	Emp_number	Lastname	Firstname	Gender	Date_joined	Department	Salary	Click
1	C00001	Khanh	Tha	F	3/3/2024	Computer	$3,000.00	
2	C00002	Wu	Shang	M	2/13/2024	Finance	$3,500.00	
3	C00003	Fritz	Eugene	M	2/12/2024	Management	$6,000.00	
4	C00004	Do	Thu Ho	F	2/29/2024	HR	$3,000.00	
5	C00005	Amrisan	Cherry	F	4/5/2024	PR	$2,700.00	
6	C00006	Ardison	Bernardo	M	4/3/2024	Logistics	$2,900.00	
8	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
9	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
10	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
(New)							$0.00	

Update

```
UPDATE  Employee_details
SET Employee_details.Lastname = 'THANTRA', Employee_details.Salary = '3600'
WHERE  Employee_details.ID =8;
```

Wildcard Characters

Wildcard Characters

- Wildcard characters substitute other characters in a string. These characters are used with a LIKE operator. Two examples of wildcards are:
- Percentage (%) – represents zero or more characters
- Underscore(_) –represents a single character

The query to select employee names starting with Dav is:

```
SELECT * FROM Employeedetails WHERE Name LIKE 'Dav%';
```

Result for the query:

SELECT * FROM Employeedetails WHERE Name LIKE ‘Dav%’;

Employee ID	Name	Department	Date of joining	Age
1	David Watson	Engineering	2014-03-28	45
5	David Rayner	Research and development	2011-12-11	42

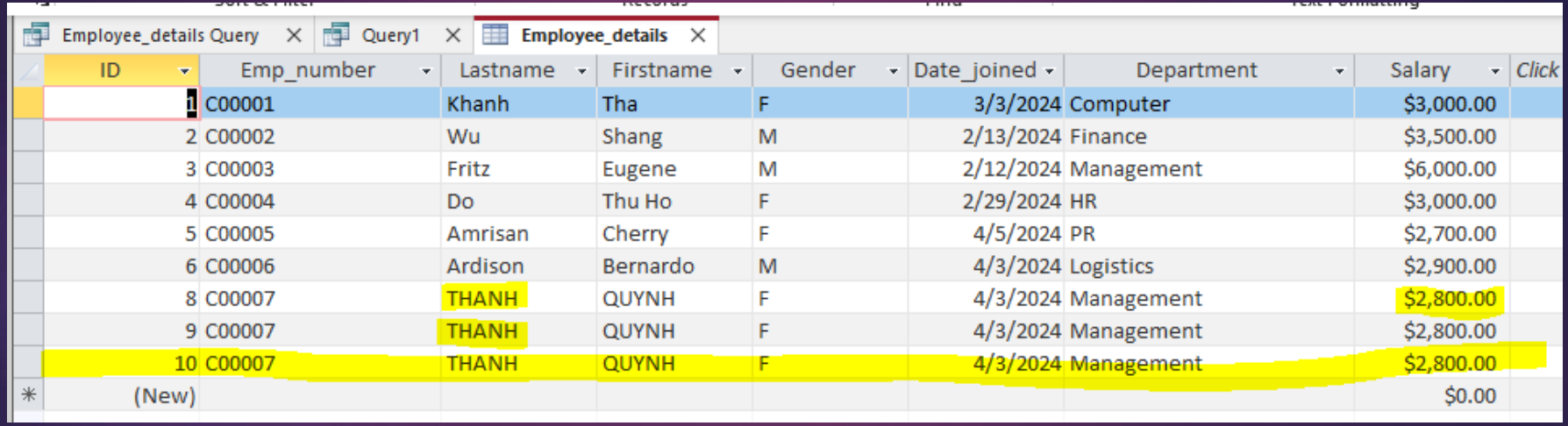
DELETE Command

DELETE command

- DELETE command is used to remove data from a table.

DELETING

Can you DELETE the Record with ID number = 10

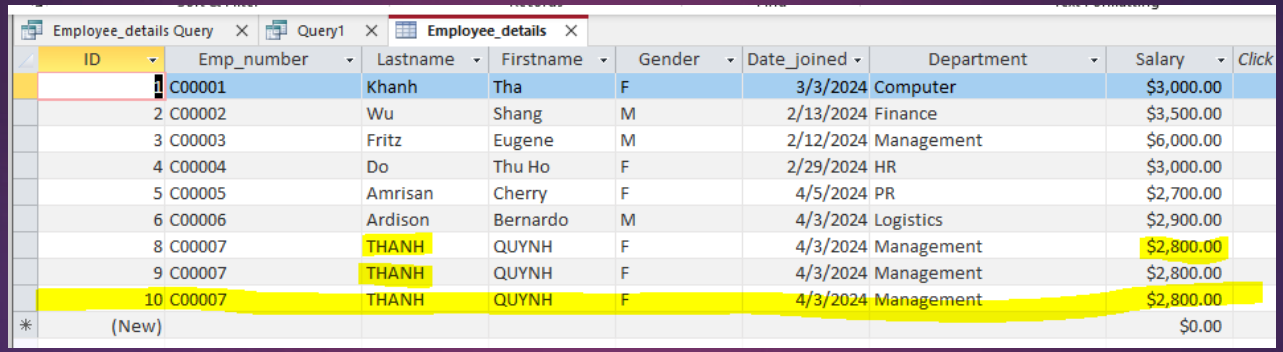


Employee_details Query | Query1 | Employee_details

ID	Emp_number	Lastname	Firstname	Gender	Date_joined	Department	Salary	Click
1	C00001	Khanh	Tha	F	3/3/2024	Computer	$3,000.00	
2	C00002	Wu	Shang	M	2/13/2024	Finance	$3,500.00	
3	C00003	Fritz	Eugene	M	2/12/2024	Management	$6,000.00	
4	C00004	Do	Thu Ho	F	2/29/2024	HR	$3,000.00	
5	C00005	Amrisan	Cherry	F	4/5/2024	PR	$2,700.00	
6	C00006	Ardison	Bernardo	M	4/3/2024	Logistics	$2,900.00	
8	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
9	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
10	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
(New)							$0.00	

DELETING

Can you DELETE the Record with ID number = 10



Employee_details Query | Query1 | Employee_details

ID	Emp_number	Lastname	Firstname	Gender	Date_joined	Department	Salary	Click
1	C00001	Khanh	Tha	F	3/3/2024	Computer	$3,000.00	
2	C00002	Wu	Shang	M	2/13/2024	Finance	$3,500.00	
3	C00003	Fritz	Eugene	M	2/12/2024	Management	$6,000.00	
4	C00004	Do	Thu Ho	F	2/29/2024	HR	$3,000.00	
5	C00005	Amrisan	Cherry	F	4/5/2024	PR	$2,700.00	
6	C00006	Ardison	Bernardo	M	4/3/2024	Logistics	$2,900.00	
8	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
9	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
10	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00	
(New)							$0.00	

DELETE FROM Employee_details

WHERE Employee_details**.**ID = 10;

DELETING

Can you DELETE the Record with ID number = 10

DELETE FROM Employee_details **WHERE** Employee_details.ID = 10;

Employee_details Query | Query1 | updating_fields | Employee_details | Query3

ID	Emp_number	Lastname	Firstname	Gender	Date_joined	Department	Salary
1	C00001	Khanh	Tha	F	3/3/2024	Computer	$3,000.00
2	C00002	Wu	Shang	M	2/13/2024	Finance	$3,500.00
3	C00003	Fritz	Eugene	M	2/12/2024	Management	$6,000.00
4	C00004	Do	Thu Ho	F	2/29/2024	HR	$3,000.00
5	C00005	Amrisan	Cherry	F	4/5/2024	PR	$2,700.00
6	C00006	Ardison	Bernardo	M	4/3/2024	Logistics	$2,900.00
8	C00007	THANTRA	QUYNH	F	4/3/2024	Management	$3,600.00
9	C00007	THANH	QUYNH	F	4/3/2024	Management	$2,800.00
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted
(New)							$0.00

Let's review some concepts

SQL

SQL is a programming language used to access and manage databases.

Syntax for SELECT statement

SELECT field_names FROM table_name WHERE condition

Syntax for creating a table

CREATE TABLE table_name(column1_name datatype, column2_name datatype,....);

Syntax for inserting data

INSERT INTO table_name (field_names) VALUES (values for all fields);

Syntax for updating data

UPDATE table_name SET column_name = value WHERE condition

Wildcard Characters

Wildcard characters substitute other characters in a string and are used with LIKE operator.

Thank you

Credit to text, images, illustrations, videos, etcs. owners on this presentation. For educational purposes only.