SQL TUTORIAL

Bài 1: SQL là gì?

* SQL là 1 ngôn ngữ chuẩn để lưu trữ, thao tác và lấy dữ liệu trong CSDL
* SQL là viết tắt của ngôn ngữ truy vấn có cấu trúc
* SQL cho phép truy cập và thao tác cơ sở dữ liệu
* SQL trở thành một tiêu chuẩn của Viện Quốc gia Hoa Kỳ Tiêu chuẩn (ANSI) vào năm 1986, và của International Organization for Standardization (ISO) vào năm 1987

Bài 2: SQL có thể làm gì?

* SQL có thể thực hiện truy vấn đối với một cơ sở dữ liệu
* SQL có thể lấy dữ liệu từ cơ sở dữ liệu
* SQL có thể chèn các bản ghi trong cơ sở dữ liệu
* SQL có thể cập nhật các bản ghi trong cơ sở dữ liệu
* SQL có thể xóa các bản ghi từ một cơ sở dữ liệu
* SQL có thể tạo ra cơ sở dữ liệu mới
* SQL có thể tạo bảng mới trong một cơ sở dữ liệu
* SQL có thể tạo thủ tục lưu trữ trong cơ sở dữ liệu
* SQL có thể tạo ra quan điểm trong một cơ sở dữ liệu
* SQL có thể thiết lập quyền truy cập vào bảng, thủ tục, và quan điểm

Bài 3: Cú pháp và câu lệnh trong SQL

Cú pháp:

* SELECT \* FROM Customers;

Các câu lệnh:

* **SELECT** – Trích xuất dữ liệu từ CSDL
* **UPDATE** – Cập nhật dữ liệu trong CSDL
* **DELETE** – Xóa dữ liệu trong CSDL
* **INSERT INTO** – Chèn dữ liệu mới vào trong CSDL
* **CREATE DATABASE** – Tạo ra 1 CSDL mới
* **ALTER DATABASE** – Thay đổi CSDL
* **CREATE TABLE** – Tạo ra 1 bảng mới trong CSDL
* **ALTER TABLE** – Thay đổi 1 bảng trong CSDL
* **DROP TABLE** – Xóa 1 bảng CSDL
* **CREATE INDEX** – Tạo ra 1 chỉ số (Key tìm kiếm)
* **DROP INDEX** – Xóa 1 chỉ số

Bài 4: SELECT

* Các câu lệnh SELECT được sử dụng để chọn dữ liệu từ một cơ sở dữ liệu.
* Các dữ liệu trả về được lưu trữ trong một bảng kết quả, được gọi là kết quả-set.

EX:

* SELECT column1, column2, column3
* FROM table\_name;
* SELECT \* FROM table\_name;
* SELECT Customers, City FROM Customers;
* SELECT \* FROM Customers;

## Bài 5: SELECT DISTINCT

## Câu lệnh SELECT DISTINCT được sử dụng để trả về giá trị duy nhất khác biệt.

## Bên trong 1 bảng, 1 cột chứa nhiều giá trị trùng lặp thì ta dùng SELECT DISTINCT

## SELECT DISTINCT column1, column2,…

## FROM table\_name;

## SELECT Country FROM Customers;

## SELECT DISTINCT Country FROM Customers;

## SELECT COUNT(DISTINCT Country) FROM Customers;

## Bài 6: WHERE

* Các mệnh đề WHERE được dùng để lọc hồ sơ.
* Các mệnh đề WHERE được sử dụng để trích xuất chỉ những hồ sơ mà đáp ứng 1 điều kiện nhất định
* Lệnh WHERE được sử dụng để lọc các kết quả từ CSDL theo các điều kiện khác nhau

EX:

* SELECT column1, column2,…
* FROM table\_name;
* WHERE condition;
* SELECT \* FROM Customers;
* WHERE CustomersID = 1;

Bài 7: AND, OR & NOT

**Toán tử AND**

* Khi muốn WHERE theo nhiều điều kiện và đáp ứng thỏa mãn tất cả điều kiện ta dùng toán tử AND

Cú pháp:

* SELECT column1*,*column2,...  
  FROM table\_name  
  WHERE điều\_kiện\_1 AND điều\_kiện\_2 AND điều\_kiện\_3 ...*;*

**Toán tử OR**

* Khi muốn WHERE theo nhiều điều kiện và yêu cầu chỉ cần thỏa mãn 1 trong số các điều kiện được đưa ra ta dùng toán tử OR

Cú pháp:

* SELECT column1*,*column2,...  
  FROM table\_name  
  WHERE điều\_kiện\_1 OR điều\_kiện\_2 OR điều\_kiện\_3...*;*

**NOT**

Cú pháp:

* SELECT column1*,*column2,...  
  FROM table\_name  
  WHERE NOT điều\_kiện;

Bài 8: Lệnh ORDER BY

* Đây là câu lệnh sắp xếp kết quả lấy ra từ 1 câu lệnh SELECT

Cú pháp :

* SELECT \* FROM tên\_cột ORDER BY tên\_cột ASC

Hay

* SELECT \* FROM tên\_cột ORDER BY tên\_cột DESC

Hoặc

* SELECT column1*,* column2, ...  
  FROM table\_name  
  ORDER BY column1, column2, ... ASC|DESC;
* ASC là tăng dần
* DESC là giảm dần

Bài 9: INSERT INTO

* Dùng để thêm dữ liệu vào bảng

Cú pháp:

* INSERT INTO table\_name(column1*,* column2*,* column3*,* ...)  
  VALUES (value1*,* value2*,* value3*, ..*.);

EX:

* INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)  
  VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');

Bài 10: Giá trị NULL

* NULL là giá trị rỗng, không có giá trị

Cú pháp:

* SELECT column\_namesFROM table\_nameWHERE column\_name IS NULL;

Hay

* SELECT column\_namesFROM table\_name  
  WHERE column\_name IS NOT NULL;

Bài 11: Lệnh UPDATE

* Câu lệnh UPDATE dùng để cập nhật 1 hay nhiều bản ghi

Cú pháp:

* UPDATE table\_name  
  SET column1 *=* value1*,* column2 *=* value2*, ...*  
  WHERE điều kiện;

Bài 12: DELETE

* Câu lệnh DELETE được sử dụng để xóa 1 hay nhiều bản ghi đang tồn tại trong 1 bảng

Cú pháp:

* DELETE FROM table\_name WHERE điều\_kiện;

Bài 13: SELECT TOP

* Mệnh đề SELECT TOP được dùng để xác định số lượng hồ sơ trở lại.
* SELECT TOP khoản rất hữu ích trên bảng lớn với hàng ngàn hồ sơ

Cú pháp:

**SQL SERVER**

* SELECT TOP number|percentcolumn\_name(s)FROM table\_nameWHERE điều\_kiện;

**MySQL**

* SELECT column\_name(s)  
  FROM table\_nameWHERE điều\_kiện  
  LIMIT number*;*

**Oracle**

* SELECT column\_name(s)FROM table\_name  
  WHERE ROWNUM <= number;

Bài 14: MIN & MAX

* MIN () chức năng trả về giá trị nhỏ nhất của cột được chọn.
* MAX () chức năng trả về giá trị lớn nhất của cột được chọn.

Cú pháp MIN:

* SELECT MIN(column\_name)  
  FROM table\_name  
  WHERE condition*;*

Cú pháp MAX:

* SELECT MAX(column\_name)  
  FROM table\_name  
  WHERE condition*;*

Bài 15: COUNT, AVG, SUM

* COUNT () hàm trả về số hàng phù hợp với một tiêu chí nhất định.
* AVG () chức năng trả về giá trị trung bình của một cột số.
* SUM () trả về tổng của một cột số.

Cú pháp COUNT:

* SELECT COUNT(column\_name)  
  FROM table\_name  
  WHERE condition;

EX:

* SELECT COUNT(ProductID)  
  FROM Products;

Cú pháp AVG:

* SELECT AVG(column\_name)  
  FROM table\_name  
  WHERE condition;

EX:

* SELECT AVG(Price)  
  FROM Products;

Cú pháp SUM:

* SELECT SUM(column\_name)  
  FROM table\_name  
  WHERE condition;

EX:

* SELECT SUM(Quantity)  
  FROM OrderDetails;

Bài 16: LIKE

* Để sử dụng tìm kiếm với lệnh WHERE
* Sử dụng kết hợp với 2 ký tự (%, \_)
* % đại diện cho không, một hoặc nhiều ký tự
* \_ đại diện cho 1 ký tự đơn

Cú pháp:

* SELECT column1, column2, ...  
  FROM table\_nameWHERE columnNLIKE pattern;

|  |  |
| --- | --- |
| WHERE CustomerName LIKE 'a%' | Tìm mọi giá trị bắt đầu bằng chữ “a” |
| WHERE CustomerName LIKE '%a' | Tìm mọi giá trị kết thúc bằng chữ “a” |
| WHERE CustomerName LIKE '%or%' | Tìm mọi giá trị có chứa chữ “or” ở vị trí bất kỳ |
| WHERE CustomerName LIKE '\_r%' | Tìm mọi giá trị có chứa chữ “r” ở vị trí thứ 2 |

|  |  |
| --- | --- |
| WHERE CustomerName LIKE 'a\_%\_%' | Tìm bất kỳ giá trị "a" và có độ dài ít nhất 3 ký tự |
| WHERE ContactName LIKE 'a%o' | Tìm mọi giá trị bắt đầu bằng chữ "a" và kết thúc với giá trị "o" |

Bài 17: Câu lệnh IN

* Câu lệnh IN được sử dụng trong câu lệnh WHERE
* IN là cách viết tắt cho nhiều điều kiện OR

Cú pháp:

* SELECT column\_name(s)  
  FROM table\_name  
  WHERE column\_nameIN(value1*,* value2*, ...*);
* Hoặc
* SELECT column\_name(s)  
  FROM table\_name  
  WHERE column\_name IN (SELECT STATEMENT);

EX:

* SELECT \* FROM Customers  
  WHERE Country IN ('Germany', 'France', 'UK');

Hay

* SELECT \* FROM Customers  
  WHERE Country NOT IN ('Germany', 'France', 'UK');

Hoặc

* SELECT \* FROM Customers  
  WHERE Country IN (SELECT Country FROM Suppliers);

Bài 18: Câu lệnh BETWEEN

* Câu lệnh BETWEEN lựa chon các giá trị trong 1 phạm vi nhất định. Các giá trị có thể là số, văn bản hoặc ngày.

Cú pháp:

* SELECT column\_name(s)FROM table\_nameWHERE column\_name BETWEEN value1 AND value2;

EX:

* SELECT \* FROM Products  
  WHERE Price BETWEEN 10 AND 20;

Hay

* SELECT \* FROM Products  
  WHERE Price NOT BETWEEN 10 AND 20;

Hoặc

* SELECT \* FROM Products  
  WHERE (Price BETWEEN 10 AND 20)  
  AND NOT CategoryID IN (1,2,3);

Bài 19: Câu lệnh Aliases

* Câu lệnh Aliases được sử dụng để cung cấp cho 1 bảng, 1 cột trong bảng 1 cái tên tạm thời
* Aliases thường được sử dụng để làm cho tên cột dễ đọc hơn
* Aliases chỉ tồn tại trong thời gian truy vấn

Cú pháp:

* SELECT column\_name AS alias\_nameFROM table\_name;

Alias bảng:

* SELECT column\_name(s)  
  FROM table\_name AS alias\_name;

Bài 20, 21, 22, 23, 24, 25:

Câu lệnh JOIN, INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, SELF JOIN

* Câu lệnh JOIN được sử dụng để kết hợp từ 2 hoặc nhiều bảng, dựa trên 1 cột liên quan giữa chúng
* (INNER) JOIN: Trả hồ sơ có giá trị tương ứng trong cả hai bảng
* LEFT (OUTER) JOIN: Return tất cả hồ sơ từ bảng bên trái, và các hồ sơ phù hợp từ bảng bên phải
* RIGHT (OUTER) JOIN: Return tất cả hồ sơ từ bảng bên phải, và các hồ sơ phù hợp từ bảng trái
* FULL (OUTER) JOIN : Return tất cả hồ sơ khi có một trận đấu trong cả hai bảng bên trái hoặc phải



Cú pháp:

* SELECT column\_name(s)  
  FROM table1  
  INNER JOIN table2 ON table1.column\_name *=* table2.column\_name;

EX:

SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName  
FROM ((Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)  
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);



Cú pháp:

* SELECT column\_name(s)FROM table1LEFT JOIN table2 ON table1.column\_name *=* table2.column\_name;

EX:

* SELECT Customers.CustomerName, Orders.OrderID  
  FROM Customers  
  LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
  ORDER BY Customers.CustomerName;



Cú pháp:

* SELECT column\_name(s)  
  FROM table1  
  RIGHT JOIN table2 ON table1.column\_name *=* table2.column\_name*;*

EX:

SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
FROM Orders  
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID  
ORDER BY Orders.OrderID;



Cú pháp:

* SELECT column\_name(s)FROM table1  
  FULL OUTER JOIN table2 ON table1.column\_name *=* table2.column\_name*;*

EX:

* SELECT Customers.CustomerName, Orders.OrderID  
  FROM Customers  
  FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID  
  ORDER BY Customers.CustomerName;

SELT JOIN

Cú pháp:

* SELECT column\_name(s)  
  FROM table1 T1, table1 T2WHERE condition*;*

EX:

* SELECT A.CustomerName AS CustomerName1, B.CustomerName AS CustomerName2, A.City  
  FROM Customers A, Customers B  
  WHERE A.CustomerID <> B.CustomerID  
  AND A.City = B.City   
  ORDER BY A.City;

Bài 26: Câu lệnh Union

* UNION được sử dụng để kết hợp kết quả tập hợp của 2 hoặc nhiều SELECT
* Mỗi câu SELECT trong UNION phải cùng số cột
* Các cột cũng phải cũng dữ liệu tương tự
* Các cột trong SELECT phải theo thứ tự

Cú pháp:

* SELECT column\_name(s) FROM table1  
  UNION  
  SELECT column\_name(s) FROM table2*;*

Hay

* SELECT column\_name(s) FROM table1  
  UNION ALL  
  SELECT column\_name(s) FROM table2;

EX:

* SELECT City FROM Customers  
  UNION  
  SELECT City FROM Suppliers  
  ORDER BY City;

Hay

* SELECT City FROM Customers  
  UNION ALL  
  SELECT City FROM Suppliers  
  ORDER BY City;

Hoặc

* SELECT City, Country FROM Customers  
  WHERE Country='Germany'  
  UNION  
  SELECT City, Country FROM Suppliers  
  WHERE Country='Germany'  
  ORDER BY City;
* SELECT City, Country FROM Customers  
  WHERE Country='Germany'  
  UNION ALL  
  SELECT City, Country FROM Suppliers  
  WHERE Country='Germany'  
  ORDER BY City;

Bài 27: Câu lệnh GROUP BY

* Câu lệnh GROUP BY được sử dụng với các hàm COUNT, MAX, MIN, SUM, AVG để nhóm kết quả theo 1 cột hoặc nhiều cột

Cú pháp:

* SELECT column\_name(s)FROM table\_nameWHERE conditionGROUP BY column\_name(s)ORDER BY column\_name(s);

EX:

* SELECT COUNT(CustomerID), Country  
  FROM Customers  
  GROUP BY Country  
  ORDER BY COUNT(CustomerID) DESC;

Hoặc

* SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders
* FROM Orders  
  LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID  
  GROUP BY ShipperName;

Bài 28: Mệnh đề HAVING

* Mệnh đề HAVING đã được thêm vào SQL vì từ khóa WHERE không thể được sử dụng với các hàm tổng hợp.

### Cú pháp:

* SELECT column\_name(s)  
  FROM table\_nameWHERE conditionGROUP BY column\_name(s)HAVING conditionORDER BY column\_name(s);

EX:

* SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders  
  FROM (Orders  
  INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID)  
  GROUP BY LastName  
  HAVING COUNT(Orders.OrderID) > 10;

## Bài 29: Toán tử EXISTS

Toán tử EXISTS được sử dụng để kiểm tra sự tồn tại của bất kỳ bản ghi nào trong truy vấn con.

Toán tử EXISTS trả về true nếu truy vấn con trả về một hoặc nhiều bản ghi.

### Cú pháp:

* SELECT column\_name(s)FROM table\_nameWHERE EXISTS  
  (SELECT column\_name FROM table\_nameWHERE condition);

EX:

* SELECT SupplierName  
  FROM Suppliers  
  WHERE EXISTS (SELECT ProductName FROM Products WHERE SupplierId = Suppliers.supplierId AND Price < 20);

Bài 30: Toán tử ANY & ALL

* Các toán tử ANY and và ALL được sử dụng với mệnh đề WHERE hoặc HAVING.
* Toán tử ANY return trả về true nếu bất kỳ giá trị truy vấn con nào đáp ứng điều kiện.
* Toán tử ALL trả về true nếu tất cả các giá trị truy vấn con đáp ứng điều kiện.

Cú pháp ANY:

* SELECT column\_name(s)FROM table\_nameWHERE column\_name operator ANY  
  (SELECT column\_name FROM table\_name WHERE condition);

EX:

* SELECT ProductName  
  FROM Products  
  WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);

Cú pháp ALL:

* SELECT column\_name(s)  
  FROM table\_name  
  WHERE column\_name operator ALL  
  (SELECT column\_name FROM table\_name WHERE condition);

EX:

* SELECT ProductName  
  FROM Products  
  WHERE ProductID = ALL (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);

Bài 31: Câu lệnh SELECT INTO

* Câu lệnh SELECT INTO sao chép dữ liệu từ một bảng vào một bảng mới.

Cú pháp:

Sao chép tất cả các cột vào 1 bảng mới

* SELECT \*  
  INTO newtable [IN externaldb]  
  FROM oldtableWHERE condition*;*

Chỉ sao chép một số cột vào một bảng mới

* SELECT column1*,*column2*,*column3*, ...*INTO newtable[IN externaldb]  
  FROM oldtableWHERE condition;

EX:

* SELECT \* INTO CustomersVietnam  
  FROM Customers  
  WHERE Country = 'Vietnam';

Bài 32: Câu lệnh INSERT INTO SELECT

* Câu lệnh INSERT INTO SELECT sao chép dữ liệu từ một bảng và chèn nó vào một bảng khác.
* INSERT INTO SELECT yêu cầu các loại dữ liệu trong bảng nguồn và bảng đích khớp
* Các hồ sơ hiện có trong bảng mục tiêu không bị ảnh hưởng

Cú pháp:

Sao chép tất cả các cột từ bảng này sang bảng khác

* INSERT INTO table2  
  SELECT \* FROM table1WHERE condition;

Chỉ sao chép một số cột từ một bảng vào một bảng khác

* INSERT INTO table2 (column1, column2, column3, ...)  
  SELECT column1, column2, column3, ...  
  FROM table1  
  WHERE condition;

Bài 33: CASE

* Câu lệnh CASE đi qua các điều kiện và trả về một giá trị khi điều kiện đầu tiên được đáp ứng (như câu lệnh IF-THEN-ELSE). Vì vậy, một khi một điều kiện là đúng, nó sẽ ngừng đọc và trả về kết quả. Nếu không có điều kiện nào đúng, nó sẽ trả về giá trị trong mệnh đề ELSE.
* Nếu không có phần ELSE và không có điều kiện nào đúng, nó sẽ trả về NULL.

Cú pháp:

* CASE  
      WHEN condition1 THEN result1  
      WHEN condition2 THEN result2    WHEN conditionN THEN resultN  
      ELSE result  
  END;

EX:

* SELECT OrderID, Quantity,  
  CASE  
      WHEN Quantity > 30 THEN "The quantity is greater than 30"  
      WHEN Quantity = 30 THEN "The quantity is 30"  
      ELSE "The quantity is under 30"  
  END  
  FROM OrderDetails;

Hoặc

* SELECT CustomerName, City, Country  
  FROM Customers  
  ORDER BY  
  (CASE  
      WHEN City IS NULL THEN Country  
      ELSE City  
  END);