**TRUY VẤN SQL**

1. **Toán tử trong SQL**

Toán tử là từ dành riêng hoặc ký tự được sử dụng chủ yếu trong mệnh đề WHERE của [lệnh SQL](https://quantrimang.com/13-cau-lenh-sql-quan-trong-programmer-nao-cung-can-biet-136595) để thực hiện các thao tác, chẳng hạn như so sánh, các phép tính số học. Những toán tử này được sử dụng để chỉ định điều kiện trong một câu lệnh SQL và dùng như giao của nhiều điều kiện trong một câu lệnh.

Toán tử trong SQL bao gồm:

* Toán tử số học
* Toán tử so sánh
* Toán tử logic
* Toán tử được sử dụng để phủ nhận các điều kiện
  1. Toán tử số học trong SQL

Giả sử biến a lưu giá trị là 10, biến b lưu giá trị 20, sau đó

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Mô tả** | **Ví dụ** |
| + (Cộng) | Cộng các giá trị ở hai bên của toán tử (phép cộng). | a + b cho kết quả là 30 |
| - (Trừ) | Lấy toán hạng bên trái trừ toán hạng bên phải (phép trừ). | a - b cho kết quả là -10 |
| \* (Nhân) | Nhân giá trị của hai toán hạng ở hai bên toán tử (phép nhân). | a \* b cho kết quả là 200 |
| / (Chia) | Chia toán hạng bên trái cho toán hạng bên phải (phép chia). | b / a cho kết quả là 2 |
| % (Chia lấy số dư) | Chia toán hạng bên trái cho toán hạng bên phải rồi lấy số dư (phép chia lấy phần dư). | b % a cho kết quả là 0 |

* 1. Toán tử so sánh

Với a=10, b=20

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Mô tả** | **Ví dụ** |
| = | Kiểm tra xem giá trị của 2 toán hạng ở hai bên dấu bằng có bằng nhau không, nếu có bằng thì điều kiện là true. | (a = b) không true. |
| != | Kiểm tra xem giá trị của 2 toán hạng có bằng nhau không, nếu khác thì điều kiện là true. | (a != b) là true. |
| <> | Kiểm tra xem giá trị của 2 toán hạng có bằng nhau không, nếu không bằng thì điều kiện là true. | (a <> b) làs true. |
| > | Kiểm tra xem giá trị của toán hạng bên trái có lớn hơn giá trị của toán hạng bên phải không, nếu có thì điều kiện là true. | (a > b) không true. |
| < | Kiểm tra xem giá trị của toán hạng bên trái có nhỏ hơn giá trị của toán hạng bên phải không, nếu có thì điều kiện là true. | (a < b) là true. |
| >= | Kiểm tra xem giá trị của toán hạng bên trái có lớn hơn hoặc bằng giá trị của toán hạng bên phải không, nếu có thì điều kiện là true. | (a >= b) không true. |
| <= | Kiểm tra xem giá trị của toán hạng bên trái có nhỏ hơn hoặc bằng giá trị của toán hạng bên phải không, nếu có thì điều kiện là true. | (a <= b) là true. |
| !< | Kiểm tra xem giá trị của toán hạng bên trái là không nhỏ hơn giá trị của toán hạng bên phải không, nếu có thì điều kiện là true. | (a !< b) là false. |
| !> | Kiểm tra xem giá trị của toán hạng bên trái là không lớn hơn giá trị của toán hạng bên phải không, nếu có thì điều kiện là true. | (a !> b) là true. |

* 1. Toán tử logic

|  |  |
| --- | --- |
| **Toán tử** | **Mô tả** |
| **ALL** | Toán tử ALL được sử dụng để so sánh một giá trị với tất cả các giá trị trong tập hợp giá trị khác. |
| **AND** | Toán tử AND cho phép nhiều điều kiện cùng tồn tại trong mệnh đề WHERE của lệnh SQL. |
| **ANY** | Toán tử ANY được sử dụng để so sánh một giá trị với bất kỳ giá trị thích hợp nào trong danh sách tùy theo điều kiện. |
| **BETWEEN** | Toán tử BETWEEN được sử dụng để tìm kiếm giá trị trong một tập hợp giá trị, với giá trị nhỏ nhất và giá trị lớn nhất được cho trước. |
| **EXISTS** | Toán tử EXISTS được sử dụng để tìm kiếm sự có mặt của một hàng trong bảng thỏa mãn điều kiện cụ thể. |
| **IN** | Toán tử IN được sử dụng để so sánh giá trị với danh sách chuỗi giá trị đã được xác định trước. |
| **LIKE** | Toán tử LIKE được sử dụng để so sánh giá trị với các giá trị tương tự, sử dụng các toán tử Wildcard. |
| **NOT** | Toán tử NOT đảo ngược ý nghĩa của toán tử logic mà nó được sử dụng. Ví dụ: NOT EXISTS, NOT BETWEEN, NOT IN,... Đây là một toán tử phủ định. |
| **OR** | Toán tử OR thường được sử dụng để nối nhiều điều kiện trong mệnh đề WHERE của lệnh SQL. |
| **IS NULL** | Toán tử NULL được sử dụng để so sánh một giá trị với giá trị NULL. |
| **UNIQUE** | Toán tử UNIQUE tìm kiếm tính đơn nhất cho mọi hàng trong bảng (không có bản sao). |

* Toán tử LIKE

|  |  |
| --- | --- |
| **Ký tự** | **Mô tả** |
| \_ | Thay cho một ký tự đơn |
| % | Thay cho một chuỗi |
| [] | Thay cho một ký tự đơn trong khoảng được bao bởi cặp dấu ngoặc vuông |
| [^] | Thay cho một ký tự đơn bất kỳ không nằm trong khoảng được bao bởi cặp dấu ngoặc vuông |

1. **Truy vấn cơ bản**
2. Cú pháp cơ bản

|  |
| --- |
| SELECT [tính chất] <danh sách column>  FROM <danh sách Table/Query>  [WHERE Conditions]  [GROUP BY Row(s)]  [HAVING]  [ORDER BY Column(s) [asc|desc]] |

Trong đó:

**[tính chất]** có thể là một trong các từ khóa \* (Lấy tất cả dữ liệu), DISTINCT(lấy dữ liệu không trùng lặp) , TOP<n> (lấy dữ liệu thứ n đầu tiên)

**<danh sách column>** tên các bảng cột cần hiển thị ở kết quả truy vấn

* Các column được ngăn cách với nhau bằng dấu phẩy (,)
* Có thể gán tên cho các column với cú pháp: <tên Column> AS 'Tên thay thế'.

**<danh sách table/query>** tên các bảng, nguồn để lấy dữ liệu khi truy vấn

* Các table được ngăn cách bằng dấu phẩy (,)
* **Alias** là bí danh (hay còn gọi là tên gọi tắt) của table dùng cho các table có tên quá dài, hoặc một table được dùng nhiều lần trong câu truy vấn. Có thể đặt Alias theo cú pháp: <tên table> AS <tên Alias>
* Mệnh đề **WHERE** cho phép truy vấn lựa chọn theo hàng
* Mệnh đề **GROUP BY** cho phép nhóm dữ liệu theo hàng
* Mệnh đề **HAVING** cho phép truy vấn lựa chọn theo nhóm
* Mệnh đề **ORDER BY** cho phép sắp xếp dữ liệu theo cột

1. Truy vấn lấy tất cả

|  |
| --- |
| SELECT \* FROM <tên table1>, < tên table2>,…  --- hoặc  SELECT <Tên table>.\* FROM <tên table1>, < tên table2>,… |

1. Truy vấn không trùng

Trong table, ngoài khóa chính, các column khác đều có thể chứa giá trị trùng lặp. Và đôi khi bạn cần lấy ra những dữ liệu hoàn toàn riêng biệt. Để đảm bảo điều này ta dùng từ khóa DISTINCT

|  |
| --- |
| SELECT DISTINCT <tên column>  FROM <tên table> |

1. Truy vấn lấy TOP bản ghi đầu

Với những table có số lượng record lên đến hàng nghìn, trăm nghìn, thì việc truy xuất một số lượng lớn dữ liệu như vậy có thể gây ảnh hưởng đến hiệu suất. Việc sử dụng mệnh đề SELECT TOP giúp trả về một lượng record theo yêu cầu, hữu ích hơn cho hệ thống.

|  |
| --- |
| --- Truy xuất n tập tin theo column  SELECT TOP <n> <Tên column> FROM <tên table>  --- Hoặc để truy xuất theo phần tram  SELECT TOP <m> PERCENT <Tên column>  FROM <tên table> |

Với n là số lượng record mong muốn, m là số % record mong muốn trên toàn Table

1. Truy vấn có điều kiện

Để lấy ra các record thỏa mãn điều kiện yêu cầu

|  |
| --- |
| SELECT [tính chất] <danh sách column>  FROM <danh sách Table/Query>  WHERE <điều kiện> |

1. Mệnh đề ORDER BY

Cho phép sắp xếp kết quả truy vấn theo cột và có thể sắp xếp kết quả theo chiều: Tăng dần (asc) hoặc giảm dần (desc).

|  |
| --- |
| SELECT [DISTINCT] Column(s)  FROM TableName  [WHERE  Conditions ]  ORDER BY Column(s) [asc|desc] |

1. **Bài tập**

## Bài 1:

Ứng dụng quản lý tài khoản ngân hàng liên quan đến việc quản lý các đối tượng khách hàng và quản lý các tài khoản của khách hàng. Xây dựng CSDL quản lý tài khoản ngân hàng chính là việc xác định các thông tin về các đối tượng khách hàng và các thông tin về mỗi tài khoản. Sau đây là yêu cầu chi tiết

Thông tin về khách hàng gồm có mã, họ tên, địa chỉ và số điện thoại. Trong mã khách hàng là duy nhất.

Mỗi khách hàng được cung cấp một tài khoản gồm số tài khoản, kiểu tài khoản, ngày mở tài khoản và số tiền.

Một tài khoản có 1 hoặc nhiều giao dịch và mỗi giao dịch gồm mã giao dịch, thời gian giao dịch, số tiền giao dịch, số dư tài khoản và mô tả giao dịch.

- Tạo CSDL

- Thêm dữ liệu mỗi bảng ít nhất 5 bản ghi

- Tạo các truy vấn dữ liệu trong sql sử dụng trên một bảng

- Tạo truy vấn hiển thị thông tin tất cả các tài khoản có kiểu là “Checking”

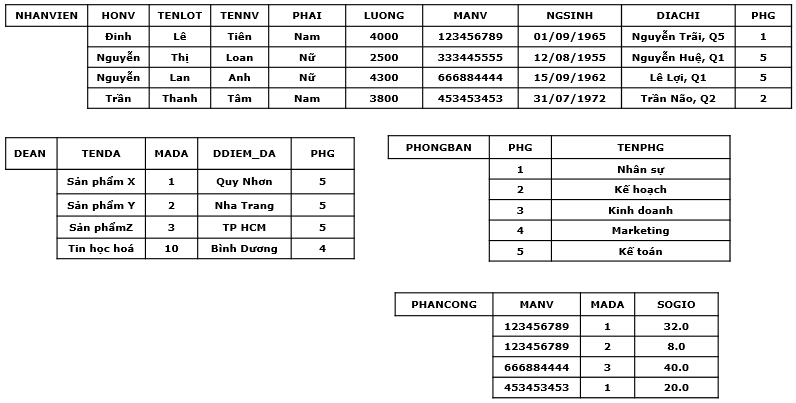
- Tạo truy vấn hiển thị các thông tin gồm (MaKH, SoTK, KieuTK, NgayMoTK) của các tài khoản có kiểu “Tài khoản cá nhân trong nước” và sắp xếp kết quả hiển thị sao cho ngày mở gần nhất sẽ được hiển thị trước (tức là ngày mở được sắp xếp giảm dần)

- Tạo truy vấn hiển thị Tên, địa chỉ, thành phố các khách hàng sống tại thành phố “Hà Nội”. Sắp xếp kết quả theo thứ tự Alphabet của tên khách hàng

* + - Tạo truy vấn hiển thị thông tin các khách hàng đã mở tài khoản trước ngày 01/07/2011.

## Bài 2:

Thực hiện:



Câu 1: Tạo cơ sở dữ liệu trên

Câu 2: Truy vấn dữ liệu trong sql:

Hiển thị tất cả thông tin của bảng NHANVIEN

Hiển thị thông tin của những nhân viên ở phòng số 5

Hiển thị mã nhân viên, họ nhân viên, tên lót và tên nhân viên của những nhân viên ở phòng số 5 và có lương >= 3000

Hiển thị mã nhân viên, tên nhân viên của những nhân viên có lương từ 2000 đến 8000

Hiển thị thông tin của những nhân viên ở địa chỉ có tên đường là Nguyễn

Cho biết số lượng nhân viên

Cho biết số lượng nhân viên trong từng phòng ban

Hiển thị thông tin về mã nhân viên, tên nhân viên và tên phòng ban ở phòng kế toán