
Bài 19

Index-View-Stored Procedures

Module: **ADVANCED PROGRAMMING WITH PHP 2.0**

Mục tiêu



- Trình bày được khái niệm chỉ mục (index)
- Tạo mới, sửa và xóa chỉ mục
- Sử dụng chỉ mục trong truy vấn SQL
- Trình bày được khái niệm khung nhìn (view)
- Tạo mới, sửa và xóa khung nhìn
- Trình bày được khái niệm thủ tục lưu (stored procedure)
- Tạo mới, sửa và xóa thủ tục lưu

Thảo luận

Chỉ mục – Index

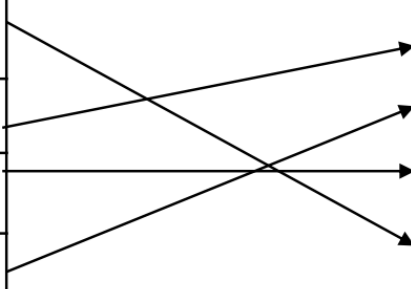
Khái niệm chỉ mục - index

- Các bản ghi được lưu trữ vào trong bảng theo đúng thứ tự như khi nhập vào. Do đó có thể dữ liệu này không được sắp xếp.
- Khó khăn: Khi muốn tìm kiếm dữ liệu trên bảng cần phải quét toàn bộ bảng. Điều này làm chậm tốc độ thực thi truy vấn.
- Việc tìm kiếm có thể được trợ giúp nhiều nếu dữ liệu được chứa trong các khối được đánh chỉ mục (index).
- Khi một chỉ mục được tạo ra trên bảng, chỉ mục sẽ tạo nên thứ tự cho các hàng dữ liệu hay các bản ghi trong bảng đó.

Khái niệm chỉ mục – Ví dụ



Emp_No	Emp_No	Emp_Name	Emp_DOB	Emp_DOJ
305	345	James	24-Sep-1968	30-May-1990
345	873	Pamela	27-Jul-1970	19-Nov-1993
693	693	Allan	10-Sep-1970	01-Jul-1992
873	305	Geoff	12-Feb-1973	29-Oct-1996



Chỉ mục – Ưu và nhược điểm



Ưu điểm:

- Cho phép chương trình tìm dữ liệu trong bảng mà không cần duyệt qua toàn bộ bảng.
- Tăng tốc độ thực thi các truy vấn cần nối nhiều bảng hay cần sắp xếp dữ liệu.
- Cải thiện sự thực thi cơ sở dữ liệu bởi giảm bớt thời gian truy cập.

Hạn chế:

- Các bảng có các chỉ mục đòi hỏi nhiều dung lượng bộ nhớ hơn trong CSDL.
- Các lệnh thao tác dữ liệu đòi hỏi nhiều thời gian xử lý hơn, vì chúng cần cập nhật sự thay đổi tới các chỉ mục.



Tạo chỉ mục

- Sử dụng câu lệnh "CREATE INDEX".
- Cú pháp:

```
CREATE[UNIQUE][CLUSTERED|NONCLUSTERED]INDEX index_name  
ON <table or view> ( column_name [ ASC | DESC ] [ ,...n ] )
```

- Ví dụ:
CREATE INDEX IX_Country ON Customer(Country);



Xóa chỉ mục

- Các chỉ mục không còn cần thiết có thể bị xóa khỏi CSDL để giải phóng bộ nhớ bằng cách sử dụng lệnh **DROP INDEX**

- Cú pháp:

`DROP INDEX table_name.index_name`

- Ví dụ:

`DROP INDEX sales.NCLINDX_ordnum`

Sử dụng chỉ mục

MySQL sử dụng indexes trong các hoạt động sau:

- Tìm các rows thỏa mãn điều kiện trong mệnh đề WHERE
- Nếu bảng có dữ liệu index trên nhiều cột, bất kỳ tiền tố tận cùng bên trái của index có thể được sử dụng bởi các trình tối ưu hóa để tìm kiếm hàng
- Để lấy các hàng từ việc joins các bảng khác nhau
- Để tìm giá trị MIN() và MAX() cho các cột index xác định

Thảo luận – View



View – Khung nhìn

- Một view là một bảng ảo được tạo ra từ việc tập hợp các cột của một hoặc nhiều bảng
- Một view có thể bao gồm các cột của một view khác
- Dữ liệu trong view được tập hợp từ các bảng cơ sở tham chiếu

Ưu điểm của view



- Bảo mật thông qua quyền truy cập cá nhân
- Tùy biến việc hiển thị dữ liệu
- Kết hợp dữ liệu từ các bảng hay các view
- Các thao tác trên bản ghi
- Toàn vẹn ràng buộc kiểm tra
- Với người sử dụng cuối:
 - Dễ dàng để hiểu được kết quả
 - Dễ dàng thu được dữ liệu mong muốn
- Với nhà phát triển:
 - Dễ dàng hạn chế việc nhận dữ liệu trả về
 - Dễ dàng bảo trì ứng dụng

Phân loại view



- **Standard View:** Sử dụng các cột từ một hoặc nhiều bảng
- **Indexed View:** View được tạo chỉ mục unique clustered index
- **Partitioned View:** View được tạo bằng cách kết hợp dữ liệu đã được chia ra của các bảng từ một hoặc nhiều máy chủ

Tạo view



- Cú pháp:

```
CREATE VIEW name_view AS  
SELECT column1, column1.....  
FROM table_name  
WHERE [condition];
```

- Ví dụ:

```
CREATE VIEW Salary_Details  
AS  
SELECT EmpID, FirstName, LastName, Salary FROM Employee_Details
```

Sửa view



- Cú pháp:

```
ALTER VIEW <view_name>  
AS <select statement>
```

- Ví dụ:

```
ALTER VIEW Salary_Details AS  
SELECT * FROM Employee_Details WHERE EmpID=1
```



Xoá view

- Việc xóa một View không tác động đến các bảng cơ sở mà View tham chiếu đến
- Cú pháp:

DROP VIEW <view_name>

- Ví dụ:
DROP VIEW Employee_Details



Tùy chọn “CHECK OPTION”

- Là một tùy chọn của lệnh CREATE VIEW. Mục đích của WITH CHECK OPTION là bảo đảm rằng tất cả [UPDATE](#) và [INSERT](#) thỏa mãn các điều kiện trong định nghĩa VIEW.
- Nếu chúng không thỏa mãn các điều kiện, UPDATE và INSERT sẽ trả về một lỗi.
- Cú pháp:

```
CREATE VIEW <view_name>  
AS <select statement> [WITH CHECK OPTION]
```



Stored Procedure

Thủ tục lưu trữ



Thủ tục lưu trữ - SPs

- Là tập hợp các câu lệnh transact-SQL được xem như một khối lệnh đơn nhằm thực hiện một tác vụ cụ thể.
- Hữu ích cho những tác vụ thực hiện lặp đi lặp lại
- Hỗ trợ các biến do người dùng khai báo, các điều kiện thực thi

Ưu điểm:



- Tăng tính bảo mật
- Giảm thiểu lưu thông trong mô hình Client/Server
- Khả năng sử dụng lại
- Tập trung logic nghiệp vụ trong cơ sở dữ liệu

Nhược điểm:



- Tốn tài nguyên hệ thống
- Khó xử lý khi gặp lỗi
- Khó mở rộng do đòi hỏi kỹ năng nâng cao



Tạo Store Procedure

- Cú pháp:

```
CREATE PROCEDURE procedure_name(parameter_list)
BEGIN
    //statements;
END
```

- Ví dụ:

```
CREATE PROCEDURE GetAllProducts()
BEGIN
    SELECT * FROM products;
END
```



Thực thi một thủ tục được lưu trữ

- Sử dụng câu lệnh **CALL**:
- Cú pháp:

CALL stored_procedure_name(argument_list);

- Ví dụ:

CALL GetAllProducts();

Truyền tham số Store Procedure



- Các tham số:
 - Tham số IN
 - Tham số OUT
 - Tham số INOUT

Truyền tham số Store Procedure



- Cú pháp:

```
CREATE PROCEDURE procedure_name(  
    [IN | OUT | INOUT] parameter_name datatype[(length)]  
)  
BEGIN  
    //statements;  
END
```

Truyền tham số Store Procedure



- Ví dụ:

DELIMITER //

```
CREATE PROCEDURE GetOfficeByCountry(  
    IN countryName VARCHAR(255)  
)  
BEGIN  
    SELECT *  
    FROM offices  
    WHERE country = countryName;  
END //
```

DELIMITER ;

Thực thi Store Procedure có tham số



- Ví dụ:

CALL GetOfficeByCountry('USA');

- Trong đó:
 - 'USA' giá trị của tham số, *countryName* sẽ nhận giá trị *USA*

Xóa thủ tục lưu



- Cú pháp:

DROP PROCEDURE procedure_name

- Ví dụ:

Drop Procedure Display_Customers



Tóm tắt bài học

- Sử dụng index để tăng tốc độ truy vấn.
- Một view là một bảng ảo được tạo ra từ việc tập hợp các cột của một hoặc nhiều bảng.
- Dữ liệu trong view được tập hợp từ các bảng cơ sở tham chiếu.
- Store Procedure là tập hợp các câu lệnh transact-SQL được xem như một khối lệnh đơn nhằm thực hiện một tác vụ cụ thể.
- Sử dụng Store Procedure hữu ích cho những tác vụ thực hiện lặp đi lặp lại.

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: **Database & CRUD**