

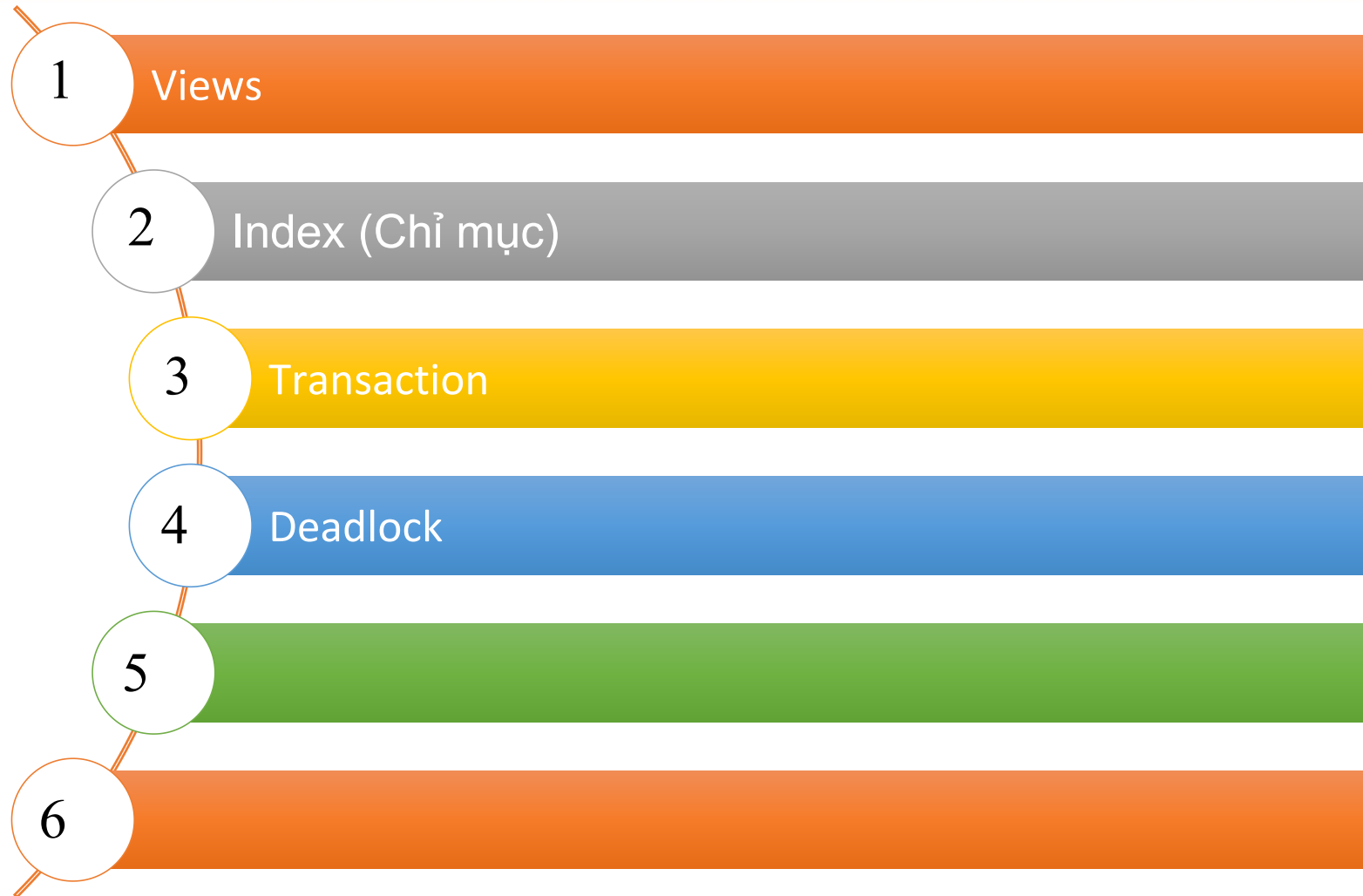


COLE.VN
Connecting knowledge

View, Index, Transaction

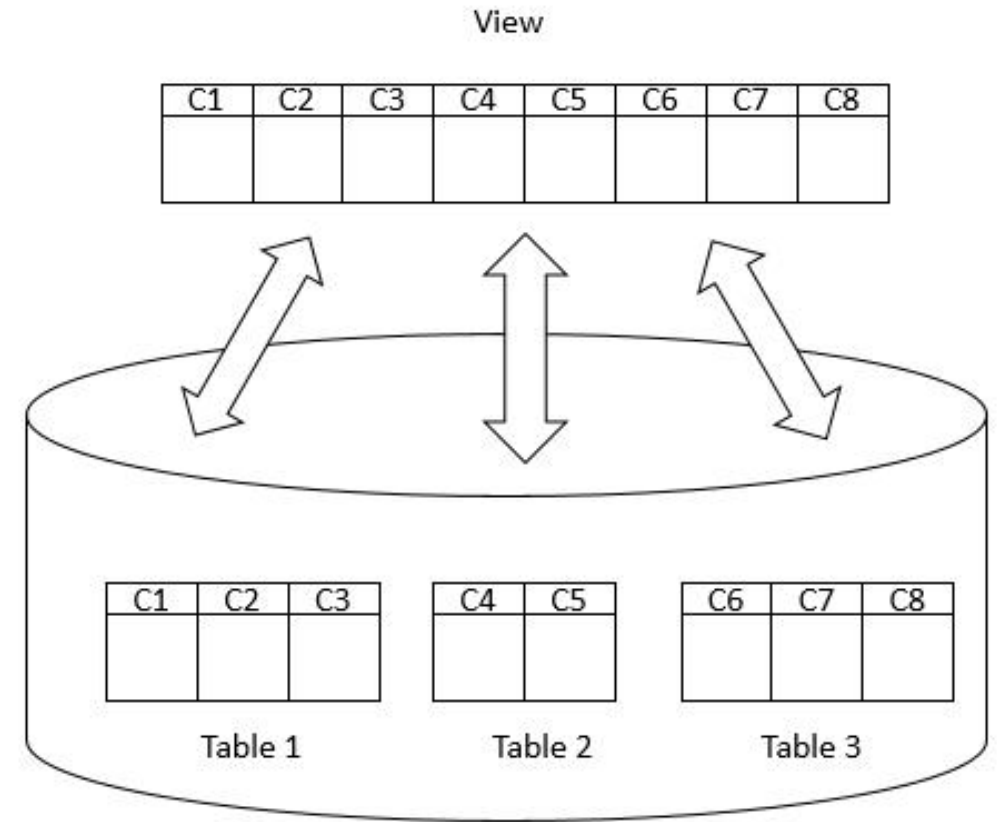
Trình bày: Nguyễn Văn Phúc

NỘI DUNG CHÍNH



1. Views

- Views là gì?
 - Views trong SQL là một bảng ảo dựa trên tập kết quả trả về của một mệnh đề select. Bản thân Views không chứa dữ liệu thực và không phải là bảng thực trong database
- Mục đích của Views?
 - Một cách để bảo mật, ẩn dữ liệu nhạy cảm. Hạn chế truy cập vào dữ liệu, chỉ cho phép truy cập vào các dữ liệu được phép
 - Đơn giản hóa những câu query phức tạp đối với những đối tượng không chuyên về SQL
- Một View có thể bao gồm các cột từ nhiều bảng hoặc một tập hợp con của các cột trong bảng.



1. Views

➤ Các câu lệnh:

▪ Tạo View

```
CREATE VIEW Name AS  
Select column1, Column2...Column N From tables  
Where conditions;
```

▪ Thay đổi View

```
ALTER VIEW Name AS  
Select column1, Column2...Column N From tables  
Where conditions;
```

▪ Xóa view

```
DROP VIEW view_name;
```

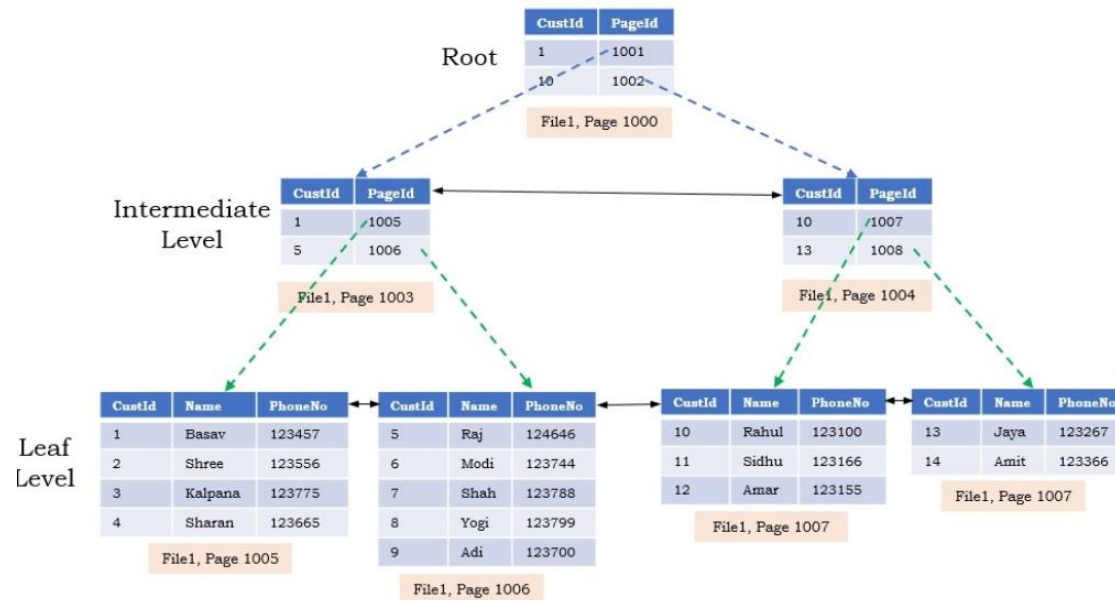
2. Index (Chỉ mục)

- Index là gì?
 - Index là cấu trúc dữ liệu đặc biệt được liên kết với các bảng hoặc Views giúp tăng tốc độ truy vấn.
- Phân loại:
 - Clustered index: Tổ chức dữ liệu bằng cách sử dụng một cây có cấu trúc đặc biệt được gọi là B-tree (Balance Tree) cho phép tìm kiếm, chèn, cập nhật, xóa theo thời gian phân bố logarit
 - Non-Clustered index : Tổ chức và lưu trữ dữ liệu riêng biệt với các hàng có trong bảng. Nó là một bản sao của các cột dữ liệu đã chọn từ một bảng với các liên kết đến bảng được liên kết.
- Khi nào nên sử dụng Index?
 - Chỉ nên sử dụng index khi cần search thông tin theo cột đó.
- Khi nào không nên sử dụng index?
 - Không nên sử dụng trong các bảng nhỏ, ít bản ghi
 - Không nên sử dụng index cho các cột thường xuyên bị sửa đổi, update dữ liệu

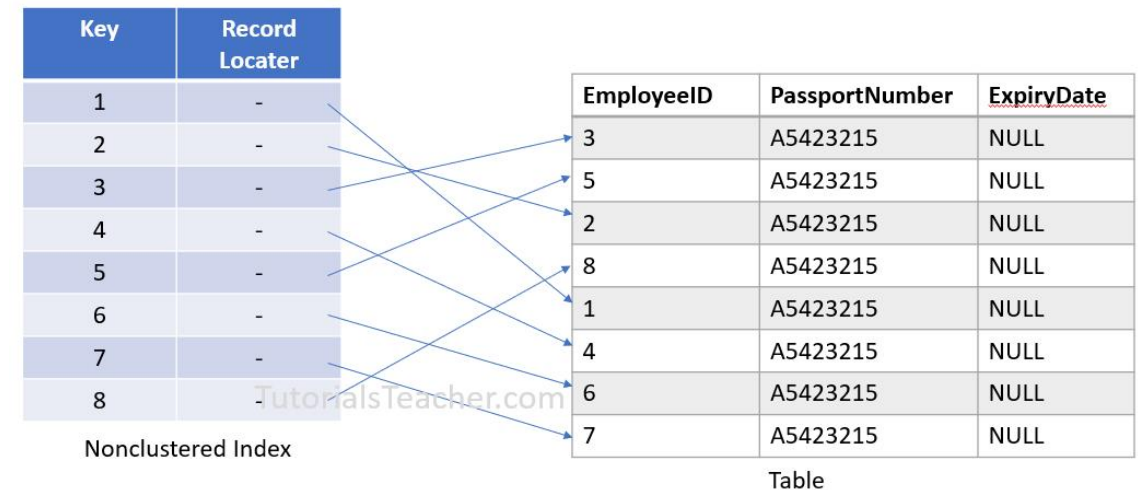
2. Index (Chỉ mục)

- Phân loại: Về cơ bản index phân thành 2 loại là Clustered Index và Non-Clustered Index
 - Clustered Index

B+ Tree Structure of a Clustered Index



- Non-Clustered Index



2. Index (Chỉ mục)

➤ Các câu lệnh:

- Tạo Index:

```
CREATE INDEX index_name  
ON table_name (column1, column2, ...);
```

- Đổi tên Index

```
EXEC sp_rename  
    index_name,  
    new_index_name,  
    N'INDEX';
```

```
EXEC sp_rename  
    @objname = N'index_name',  
    @newname = N'new_index_name',  
    @objtype = N'INDEX';
```

- Câu lệnh xóa Index:

```
DROP INDEX table_name.index_name;
```


2. Index (Chỉ mục)

- Vô hiệu hóa 1 Index: Trình tối ưu hóa truy vấn sẽ không xem xét chỉ mục vô hiệu hóa đó để tạo kế hoạch thực thi truy vấn.

```
ALTER INDEX index_name ON table_name DISABLE;
```

- Vô hiệu hóa tất cả Index của 1 bảng:

```
ALTER INDEX index_name ON table_name DISABLE;
```

- Bỏ vô hiệu hóa Index

```
ALTER INDEX index_name ON table_name REBUILD;
```

```
CREATE INDEX index_name ON table_name(column_list)  
WITH(DROP_EXISTING=ON)
```

```
ALTER INDEX ALL ON table_name REBUILD;
```

2. Index (Chỉ mục)

- Table Scan:
 - Table Scan xảy ra khi select dữ liệu ở một table không có clustered index
 - Performance thấp nhất
- Index Scan:
 - Index Scan xảy ra phụ thuộc vào mệnh đề where
 - Performance tốt hơn table scan
- Index Seek:
 - Một table với index được truy cập và cấu trúc B-tree có thể làm lọc được các kết quả trả về, giúp lấy được 1 số lượng nhất định rows từ table
 - Performance tốt nhất

3. Transaction

➤ Transaction là gì?

- Transaction là một tiến trình xử lý có xác định điểm đầu và điểm cuối, được chia nhỏ thành các operation (phép thực thi) , tiến trình được thực thi một cách tuần tự và độc lập các operation đó theo nguyên tắc hoặc tất cả đều thành công hoặc một operation thất bại thì toàn bộ tiến trình thất bại. Nếu việc thực thi một operation nào đó bị fail (hỏng) đồng nghĩa với việc dữ liệu phải rollback (trở lại) trạng thái ban đầu.

➤ Tính chất của Transaction:

- Atomicity – tính đơn vị: Nguyên tắc “all or nothing”
- Isolation – độc lập: Nếu hai transaction thực thi cùng lúc thì nguyên tắc thực thi là thực thi độc lập. Nghĩa là một transaction không thể “nhìn thấy” một transaction khác. “Không nhìn thấy” ở đây là không tác động lẫn nhau, chủ yếu trên dữ liệu
- Durability – bền vững: Dữ liệu của transaction sau khi thực thi xong được cố định, chính thức và bền vững. Nghĩa là những thay đổi đã được cố định, không có chuyện có thể chuyển lại trạng thái dữ liệu lúc trước khi thực hiện transaction

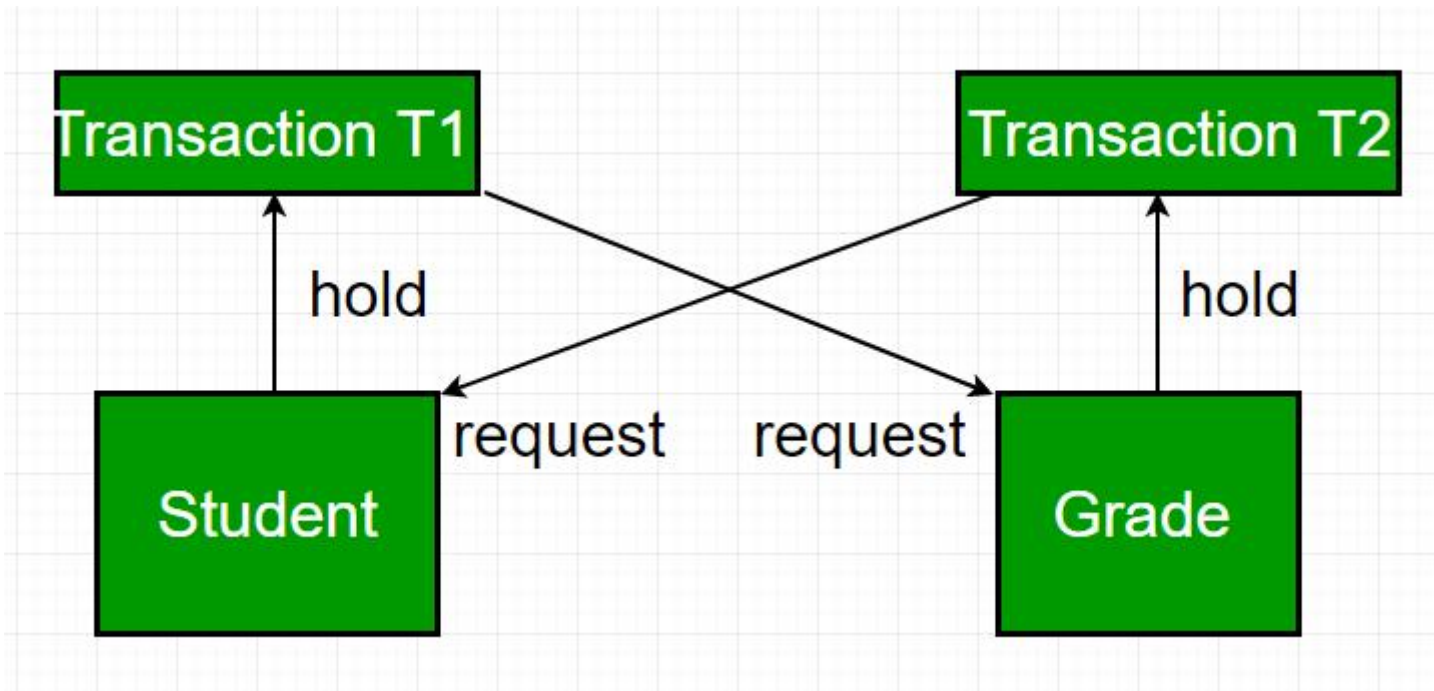
4. Deadlock

- Lock trong cơ sở dữ liệu?
 - Các hệ quản trị cơ sở dữ liệu đảm bảo tài nguyên trong database có tính nhất quán (consistency), có nghĩa là cùng một dữ liệu sẽ không thể đọc ghi tại cùng 1 thời điểm. Từ đây sinh ra khái niệm lock để đảm bảo tính nhất quán dữ liệu
- Cơ chế lock:
 - Lệnh Select sẽ xung đột với các lệnh Update, Insert, Delete. Vì các lệnh Update, Insert, Delete làm thay đổi dữ liệu của một table điều này dẫn đến việc câu lệnh Select sẽ không còn chính xác nữa trong trường hợp đang Select thì cơ sở dữ liệu lại bị thay đổi mất.
 - Các câu lệnh Select sẽ không lock nhau
 - Các câu lệnh Update, Insert, Delete sẽ chỉ lock trên cùng 1 row dữ liệu chứ không trên cùng 1 bảng. Tức là dù các câu lệnh cùng xảy ra trên 1 bảng nhưng nếu ko tranh chấp 1 row thì sẽ không lock lẫn nhau => vẫn thực hiện bình thường.

4. Deadlock

➤ Deadlock là gì?

- Deadlock là hiện tượng tranh chấp tài nguyên giữa hai hay nhiều lệnh trong đó lệnh này giữ tài nguyên mà lệnh kia cần dẫn tới việc không lệnh nào có thể kết thúc để giải phóng tài nguyên. Nếu không được xử lý sẽ dẫn đến hiện tượng các câu lệnh sẽ chờ nhau và không 1 tiến trình nào sẽ được thực hiện tiếp.



THANK YOU !

COLE.VN
Connecting knowledge