

2 CƠ BẢN VỀ SQL

Giảng viên: Cao Thị Nhâm

Nội dung chính

- Giới thiệu về SQL
- DML
- DDL
- Tối ưu hóa câu lệnh SQL
- View
- Index
- Synonym
- Sequence

SQL

- ❖ SQL: Structured Query Language
- ❖ Các loại:
 - DML (Data Manipulation Language)
 - DDL (Data Definition Language)
 - Transaction Control
 - Session Control
 - System Control

DML

(Data Manipulation Language)

Kiểu dữ liệu

Character	Numeric	Datetime	LOB	ROWID	Binary
CHAR (<i>n</i>)	NUMBER(<i>m,n</i>)	DATE	CLOB	ROWID	RAW(<i>size</i>)
NCHAR(<i>n</i>)	FLOAT	TIMESTAMP WITH TIMEZONE	NCLOB	UROWID	LONG RAW
VARCHAR2(<i>n</i>)	BINARY_FLOAT	TIMESTAMP WITH LOCAL TIMEZONE	BLOB		
NVARCHAR2(<i>n</i>)	BINARY_DOUBLE	INTERVAL YEAR[(<i>n</i>)] TO MONTH	BFILE		
		INTERVAL DAY[(<i>m</i>)] TO SECOND[(<i>n</i>)]			

ABC

42

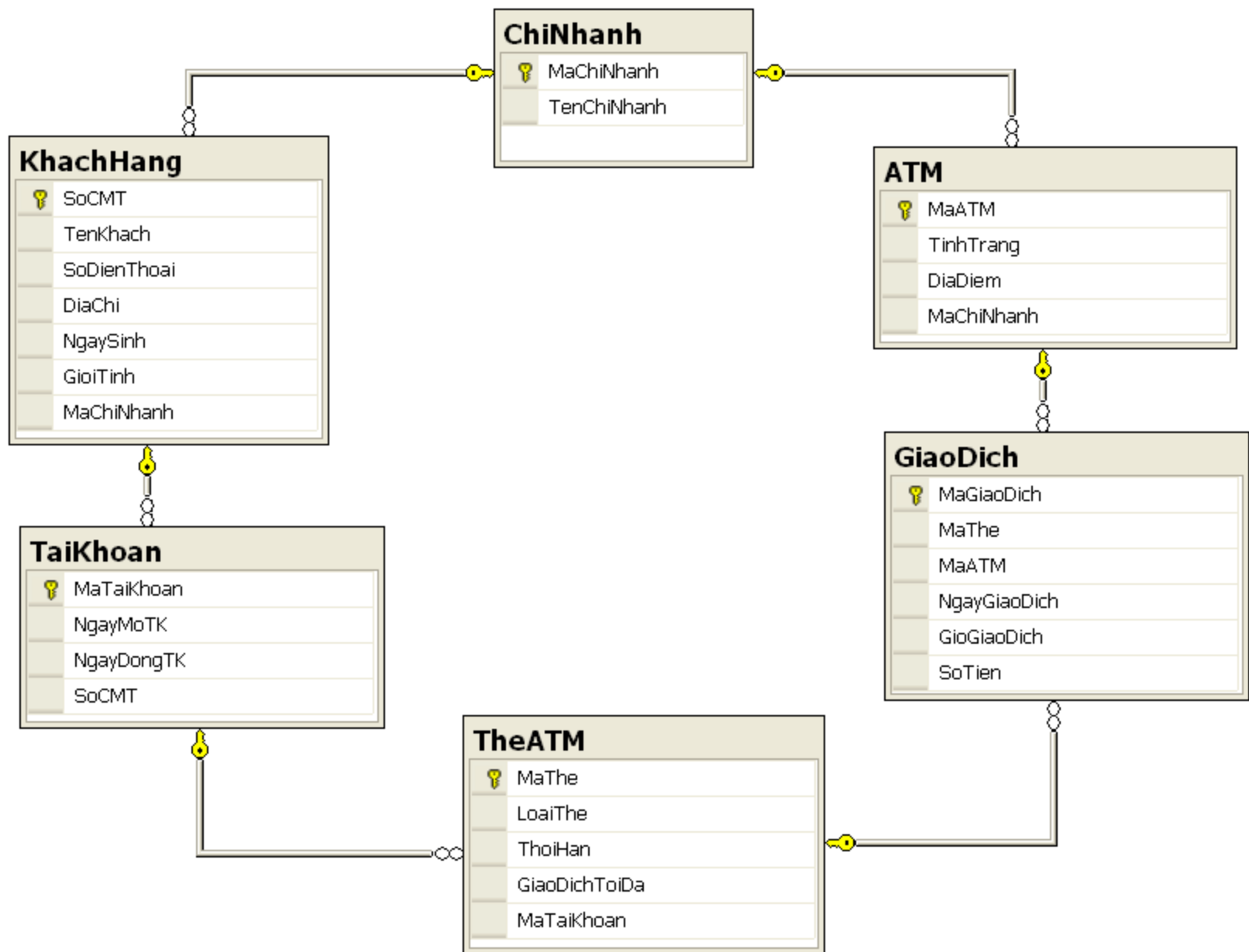


AAAM4yAABAAA09KAAA

ORACLE

Truy vấn dữ liệu

```
SELECT [DISTINCT] danh_sách_cột  
FROM {table_name | view_name}  
[WHERE điều_kiện]  
[GROUP BY danh_sách_cột_1]  
[HAVING điều_kiện_lọc]  
[ORDER BY danh_sách_cột_2 [ASC | DESC]]
```



Phép toán

Loại phép toán	Phép toán
Toán học	+, -, *, /
So sánh	=, {!=, <>, ^=}, <, >, <=, >=
→ SOME/ANY, ALL	
Logic	NOT, AND, OR
Các phép toán chuỗi	, LIKE, NOT LIKE
Các phép toán khác	IN, NOT IN, BETWEEN, EXISTS, IS NULL, IS NOT NULL

Một hàm phục vụ cho truy vấn

Chuỗi	Số	Thời gian	Chuyển đổi	Rẽ nhánh	Gộp
LENGTH	FLOOR, CEIL, ROUND	ADD_MONTHS	CAST	CASE	MIN, MAX
LOWER, UPPER	MOD	SYSDATE	TO_CHAR	DECODE	COUNT
LPAD, RPAD	SQRT	EXTRACT	TO_DATE		AVG
LTRIM, RTRIM, TRIM		MONTHS_BETWEEN	TO_NUMBER		SUM
SUBSTR					

(Tham khảo thêm trong tài liệu tham khảo trang 127-187)

Lấy dữ liệu từ nhiều bảng

- ❖ Sử dụng phép nối
 - INNER JOIN
 - NATURAL JOIN
 - OUTER JOIN
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL OUTER JOIN

TRUY VẤN LÒNG (SUBQUERY)

❖ Subquery có thể đặt ở:

- SELECT
- FROM
- WHERE
-

Tối ưu hóa truy vấn

- Chỉ SELECT những cột và những bảng ghi cần thiết
- Sử dụng JOIN thay vì subquery
- Tránh truy vấn trên view
- Gọi tên cột tường mình
- Dùng CASE thay vì sử dụng nhiều truy vấn
- Dùng INDEX
- Dùng WHERE tốt hơn HAVING
- Dùng EXISTS/NOT EXISTS tốt hơn IN/NOT IN
- Hạn chế sử dụng các phép tính toán trong mệnh đề WHERE

Sửa dữ liệu

```
UPDATE Tên_bảng  
SET cột1 = giá_trị1, ..., cộtn = giá_trị_n  
[WHERE điều_kiện];
```

Xóa dữ liệu

❖ Xóa từng bản ghi

```
DELETE Tên_bảng WHERE [điều_kiện];
```

❖ Xóa toàn bộ dữ liệu

```
TRUNCATE TABLE Tên_bảng;
```

Thêm mới dữ liệu

- ❖ Nhập giá trị cho mọi cột trong bảng

```
INSERT INTO Tên_bảng VALUES (gt1, gt2, ...)
```

- ❖ Nhập giá trị cho một số cột trong bảng

```
INSERT INTO Tên_bảng (cột1, cột2, ...)  
VALUES (gt1, gt2, ...)
```

- ❖ Lấy giá trị từ bảng khác

```
INSERT INTO Tên_bảng (cột1, cột2, ...)  
SELECT cotx, coty, ...  
FROM...WHERE...
```

DDL

(Data Definition Language)

Bảng

- ❖ Tên bảng & tên cột tuân theo quy tắc:
 - 1-30 kí tự
 - Bắt đầu bằng chữ cái
 - Bao gồm chữ cái, số, _, #, \$ (hạn chế dùng #, \$)
 - Không dùng những từ có sẵn trong Oracle (NUMBER, INDEX...)
 - Tên cột phải duy nhất trong bảng
 - Tên bảng phải duy nhất trong *namespace*

Tạo bảng

❖ Ví dụ:

```
CREATE TABLE change_log  
(log_id      NUMBER  
 ,who        VARCHAR2(64)  
 ,when       TIMESTAMP  
 ,what       VARCHAR2(200)  
 );
```

```
CREATE TABLE change_log  
(log_id      NUMBER  
 ,who        VARCHAR2(64)  
 ,when       TIMESTAMP  
 ,what       VARCHAR2(200)  
 ) TABLESPACE users;
```

SQL> describe change_log

Name	Null?	Type
LOG_ID		NUMBER
WHO		VARCHAR2(64)
WHEN		TIMESTAMP(6)
WHAT		VARCHAR2(200)

Constraint

- ❖ Đảm bảo tính toàn vẹn của dữ liệu
- ❖ Có thể tạo constraint lúc tạo bảng hoặc sau khi tạo bảng



Các loại constraint

❖ NOT NULL

❖ UNIQUE

- Không cho phép nhập giá trị giống nhau
- Oracle tự động tạo unique index cho cột có ràng buộc UNIQUE

❖ PRIMARY KEY

- Có thể tạo khóa chính cho 1 hoặc nhiều cột
- Oracle tự động tạo unique index cho cột làm khóa chính

❖ FOREIGN KEY

- Thiết lập mối quan hệ của 1 bảng với bảng khác

❖ CHECK

- Kiểm tra giá trị của một cột thỏa mãn điều kiện cho trước

Khai báo constraint - 1

❖ NOT NULL constraint

```
CREATE TABLE employees
(employee_id    NUMBER    CONSTRAINT nn_emp_id NOT NULL
,hire_date      DATE                                NOT NULL
,first_name     VARCHAR2(42)
,last_name      VARCHAR2(42)
);
```

❖ UNIQUE constraint

```
CREATE TABLE employees
(employee_id    NUMBER    NOT NULL
      CONSTRAINT uniq_payroll_id UNIQUE
,hire_date      DATE    NOT NULL
,first_name     VARCHAR2(42)
,last_name      VARCHAR2(42)
,payroll_id     VARCHAR2(10)
);
```

Khai báo constraint - 2

❖ PRIMARY KEY constraint

```
CREATE TABLE employees
(employee_id    NUMBER          NOT NULL
,hire_date     DATE            NOT NULL
,first_name    VARCHAR2(42)
,last_name     VARCHAR2(42)
,payroll_id    VARCHAR2(10)
,CONSTRAINT employees_pk PRIMARY KEY (employee_id))
```

Khai báo constraint - 3

❖ FOREIGN KEY constraint

- **Chú ý**: Không được phép tạo khóa ngoại cho những cột có kiểu dữ liệu: CLOB, NCLOB, BLOB, LONG, LONG RAW, TIMESTAMP WITH TIMEZONE

❖ Ví dụ:

```
CREATE TABLE employees
(employee_id      NUMBER          NOT NULL
,hire_date       DATE            NOT NULL
,first_name      VARCHAR2(42)
,last_name       VARCHAR2(42)
,payroll_id      VARCHAR2(10)
,dept_nbr        NUMBER
,CONSTRAINT uniq_payroll_id UNIQUE (payroll_id)
,CONSTRAINT emp_dept_fk FOREIGN KEY (dept_nbr)
REFERENCES departments(dept_nbr)
);
```

Khai báo constraint - 4

❖ CHECK constraint

- Kiểm tra giá trị của một cột có thỏa mãn điều kiện cho trước hay không

❖ Ví dụ:

```
create table SinhVien (  
MaSV          char(10) primary key,  
NgàySinh     date      ,  
TenSV        nvarchar2(50),  
constraints TenCK check (length(TenSV) > 3)  
);
```


Các thao tác liên quan tới bảng

❖ Sửa giá trị mặc định của cột

```
ALTER TABLE change_log MODIFY  
who          VARCHAR2(64) DEFAULT USER;
```

❖ Đổi tên bảng

```
RENAME change_log TO demo_change_log;
```

```
ALTER TABLE change_log RENAME TO demo_change_log;
```

❖ Xóa cột

```
ALTER TABLE change_log DROP (how,why);
```

```
ALTER TABLE change_log DROP COLUMN how;
```

Các thao tác liên quan tới bảng

❖ Sửa cột

```
ALTER TABLE change_log MODIFY what VARCHAR2(250);
```

```
ALTER TABLE change_log MODIFY  
(what    VARCHAR2(250)  
,who    VARCHAR2(50)  DEFAULT user  
);
```

❖ Thêm cột

```
ALTER TABLE change_log ADD how VARCHAR2(45);
```

```
ALTER TABLE change_log ADD  
(how    VARCHAR2(45)  
,why    VARCHAR2(60)  
);
```

Các thao tác liên quan tới bảng

❖ Thêm constraints

- Ví dụ

```
ALTER TABLE employees ADD CONSTRAINT  
    validate_hire_date CHECK  
    (hire_date > TO_DATE('15-Apr-1999', 'DD-Mon-YYYY'));
```

Các thao tác liên quan tới bảng

❖ Sửa constraints

- Xóa

```
ALTER TABLE employees DROP CONSTRAINT validate_hire_date;
```

```
ALTER TABLE employees DROP PRIMARY KEY CASCADE;
```

- Đổi tên

```
ALTER TABLE employees  
    RENAME CONSTRAINT validate_hire_date TO hire_date_check;
```

- Vô hiệu hóa (disable)

```
ALTER TABLE employees DISABLE CONSTRAINT mgr_emp_fk;  
-- bulk load the table  
ALTER TABLE employees ENABLE CONSTRAINT mgr_emp_fk;
```

Các thao tác liên quan tới bảng

❖ Khi xóa bảng, Oracle tiến hành:

- Xóa dữ liệu
- Xóa cấu trúc dữ liệu lưu trữ bảng
- Xóa các trigger liên quan tới bảng
- Xóa các quyền liên quan tới bảng

```
DROP TABLE hr.employees PURGE;
```

❖ Một số tùy chọn cho câu lệnh xóa bảng

- PURGE: không cho phép flashback
- CASCADE CONSTRAINTS: xóa mọi ràng buộc dữ liệu có liên quan

Sequence

- ❖ Là một đối tượng trong schema dùng để tự động sinh ra các số nguyên theo thứ tự nào đó (thường dùng cho khóa chính)
- ❖ Đặc điểm
 - Mỗi sequence có 1 tên xác định
 - Không gắn với 1 cột hay 1 bảng nào
 - Có thể tạo ra số nguyên theo thứ tự tăng hoặc giảm dần đều
 - Khoảng cách giữa 2 số nguyên do người dùng tùy đặt
 - Có thể quay vòng nếu giá trị sinh ra đã đạt ngưỡng

Sequence ...

❖ Các thuộc tính:

- START WITH
- INCREMENT BY
- MAXVALUE/NOMAXVALUE
- MINVALUE/NOMINVALUE
- CACHE/NOCACHE
- CYCLE/NOCYCLE

```
CREATE SEQUENCE PKValue  
START WITH 1  
INCREMENT BY 2  
MAXVALUE 10000  
MINVALUE 1;
```

```
CREATE SEQUENCE PKValue  
START WITH 1  
INCREMENT BY 2  
NOMAXVALUE  
NOMINVALUE;
```

```
SELECT PKValue.nextval from DUAL;
```

1 3 5 7 9 11 13 15 ...

Quản lý sequence

❖ Sửa sequence

- Có thể sửa các thuộc tính: INCREMENT BY, MAXVALUE, MINVALUE, CYCLE, CACHE
- Không thể sửa: START WITH
- Ví dụ:

```
ALTER SEQUENCE employees_seq INCREMENT BY -1;
```

❖ Xóa sequence

```
DROP SEQUENCE employee_seq;
```

❖ Sử dụng sequence

```
INSERT INTO Employee(EmployeeID, Name)  
VALUES(employee_seq.nextval, 'Nguyen Van A')
```


VIEW

- ❖ Biểu diễn dữ liệu từ một hoặc nhiều bảng → bảng ảo
- ❖ Câu lệnh truy vấn lưu ở Data Dictionary, dữ liệu vẫn để ở bảng gốc
- ❖ Truy vấn dữ liệu trên view như trên các bảng thông thường
- ❖ Cú pháp:

```
CREATE VIEW tên_view  
AS  
câu_truy_vấn;
```

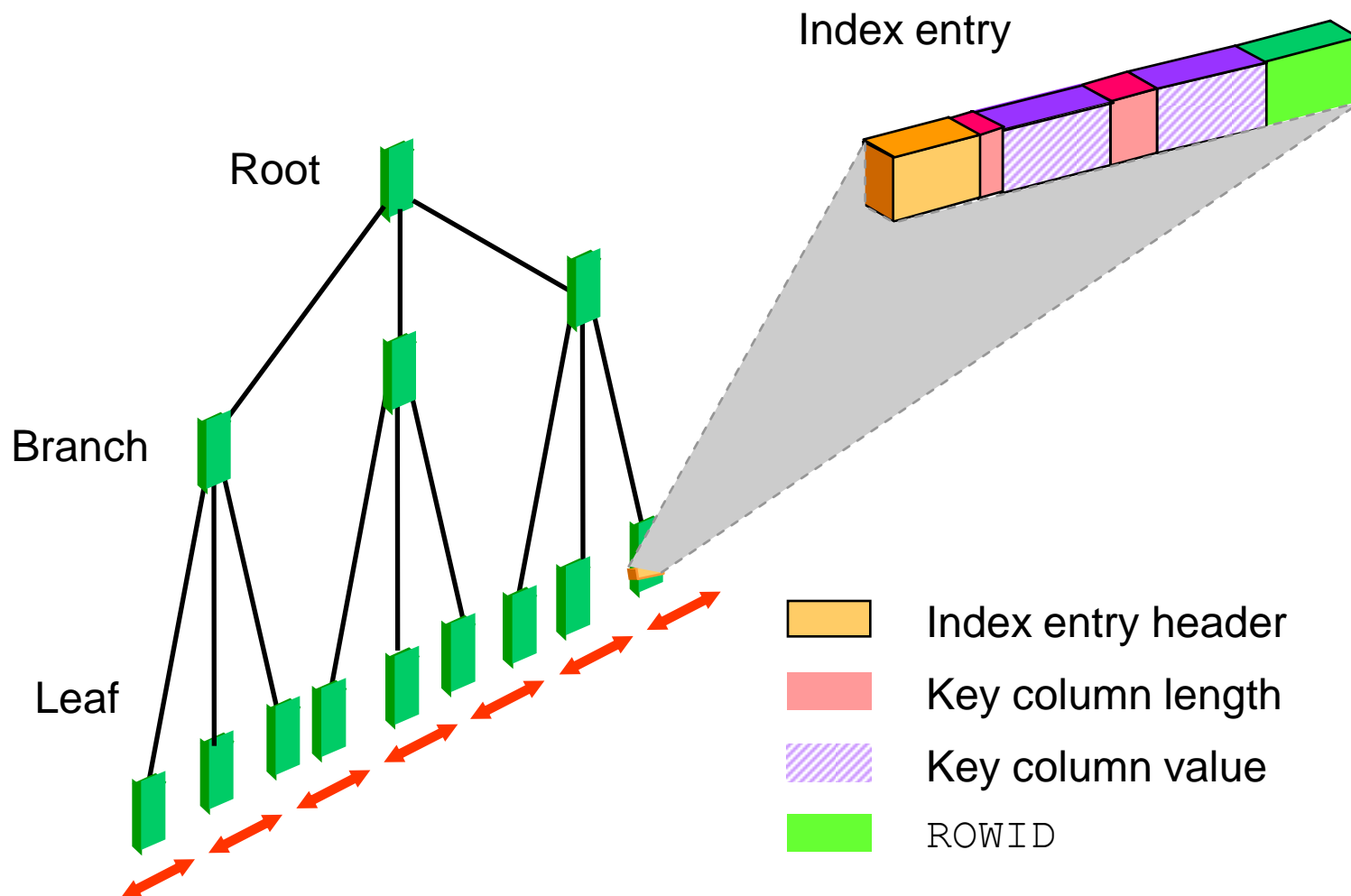
VIEW...

- ❖ Có thể thực hiện các thao tác DML, nếu view:
 - View không chứa các mệnh đề: DISTINCT, GROUP BY, START WITH, CONNECT BY, ROWNUM
 - View không chứa các phép toán tập quan hệ đại số: UNION, UNION ALL, INTERSECT, MINUS
 - View không chứa subquery
 - Câu lệnh DML tác động lên 1 bảng gốc

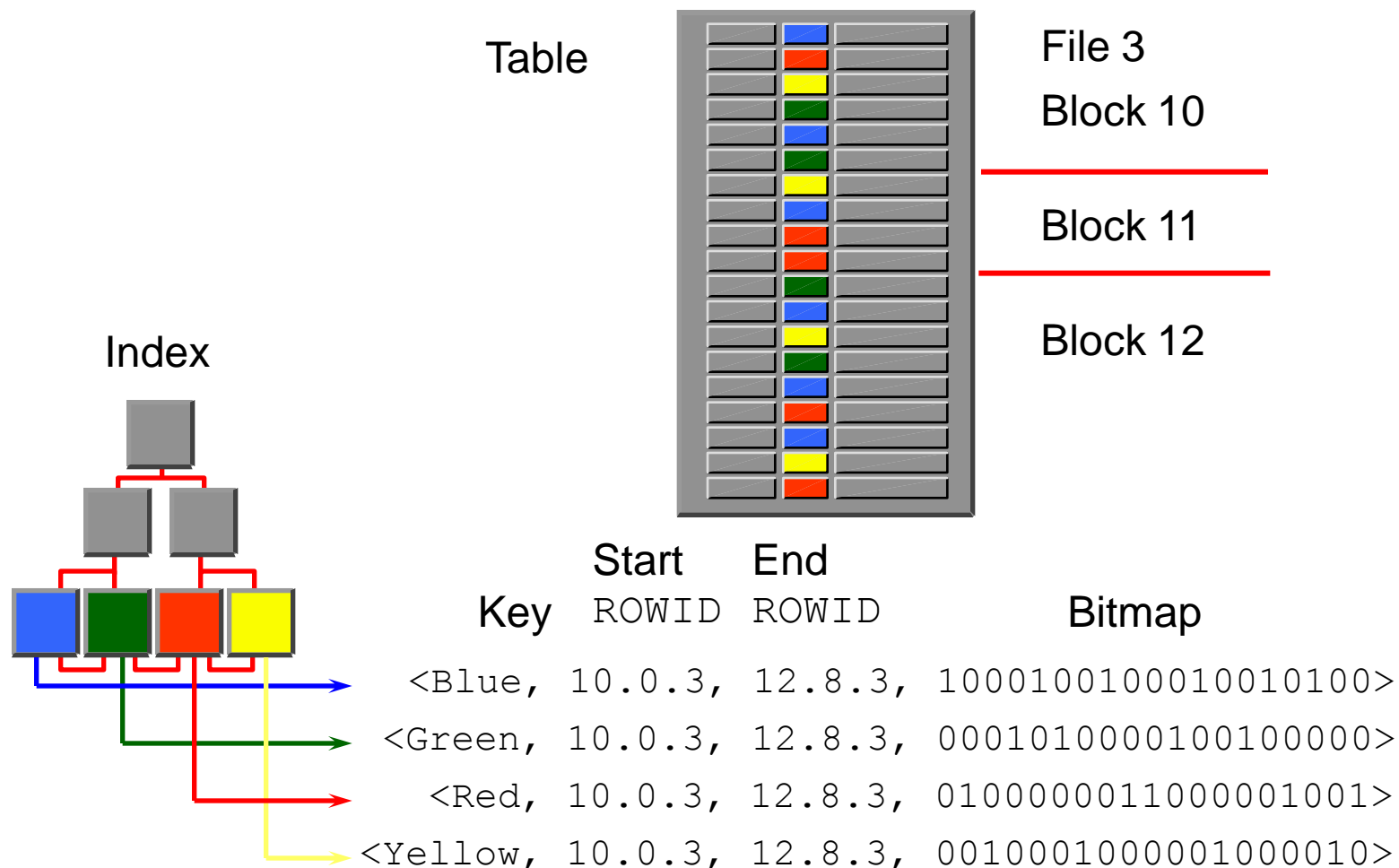
INDEX

- ❖ Là cấu trúc dữ liệu tùy chọn, xây dựng cho một bảng xác định → Tăng tốc độ truy vấn dữ liệu
- ❖ Có thể tạo index cho nhiều cột hoặc một cột có thể tạo nhiều index
- ❖ Có 2 loại index:
 - B-tree index (*mặc định*)
 - Sắp xếp giá trị khóa & ROWID dưới dạng B-tree
 - Phù hợp với những cột có ít giá trị giống nhau
 - Hỗ trợ “*row locking*”
 - Có 2 loại: unique và non-unique
 - Bitmap
 - Khóa và ROWID lưu dưới dạng bitmap
 - Phù hợp với những cột có nhiều giá trị giống nhau
 - Không hỗ trợ “*row-locking*”

B-Tree Index



Bitmap Index



INDEX...

❖ Tạo index

```
CREATE INDEX emp_dept_nbr ON employees (dept_nbr)  
TABLESPACE indx;
```

```
CREATE UNIQUE INDEX dname_uix ON departments (dept_name);
```

```
CREATE BITMAP INDEX state_bix ON geography (state);  
CREATE BITMAP INDEX region_bix ON geography (region);  
CREATE BITMAP INDEX metro_bix ON geography (metro_area);
```

❖ Xóa index

```
DROP INDEX emp_seniority;
```

❖ Quản lý index

```
ALTER INDEX emp_seniority REBUILD;
```

```
ALTER INDEX uniq_payroll_id COALESCE;
```

```
ALTER INDEX sys_c001428 RENAME TO employee_pk;
```

```
ALTER INDEX uniq_payroll_id REBUILD TABLESPACE hr_indx;
```

Synonym

- ❖ Bí danh cho mọi đối tượng trong Oracle
- ❖ Ưu điểm:
 - Đơn giản cho câu lệnh SQL
 - Không tốn không gian lưu trữ
 - Tăng tính bảo mật
- ❖ Ví dụ:

```
select * from nhamct.nhanvien;  
select * from sys.nhanvien;
```



```
create public synonym NV2 for sys.nhanvien;
```



```
select * from NV2;
```

