



FPT POLYTECHNIC



Bài 4:
NGÔN NGỮ TRUY VẤN CỐ CẤU TRÚC (SQL)

www.poly.edu.vn

hoclaptrinhweb.com

- Gi i thi u ngôn ng SQL
- Gi i thi u Oracle SQL
- Gi i thi u các nhóm ngôn ng c a SQL
- Tìm hi u nhóm ngôn ng truy v n d li u v i m nh
SELECT

Mục tiêu bài học hôm nay

- Tìm hiểu câu lệnh SELECT trên nhiều bảng
- Tìm hiểu 3 nhóm lệnh SQL:
 - Nhóm định nghĩa dữ liệu (DDL): Gồm các lệnh tạo, thay đổi các bảng dữ liệu (Create, Drop, Alter, ...)
 - Nhóm thao tác dữ liệu (DML): Gồm các lệnh làm thay đổi dữ liệu (Insert, Delete, Update,...) lưu trong các bảng
 - Nhóm quyền dữ liệu (DCL): Gồm các lệnh quản lý quyền truy cập vào dữ liệu và các bảng (Grant, Revoke, ...)

Mệnh đề **SELECT** trên nhiều bảng

- Khi truy vấn trên nhiều bảng, phải kết nối các bảng
- Có hai kiểu kết nối:
 - **Kết nối trong:** mệnh đề WHERE chia ra các trường khóa của các bảng cần kết nối với nhau hoặc dùng từ khóa JOIN trong mệnh đề FROM
 - **Kết nối ngoài:** sử dụng từ khóa LEFT/RIGHT OUTER JOIN trong mệnh đề FROM

Kết nối trong

☒ Autocommit
 Display 10 ▼

```

SELECT  EMPLOYEE_ID, LAST_NAME, FIRST_NAME, DEPARTMENT_NAME
      FROM  EMPLOYEES, DEPARTMENTS
      WHERE EMPLOYEES.DEPARTMENT_ID  =  DEPARTMENTS.DEPARTMENT_ID;
  
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_NAME
100	King	Steven	Executive
101	Kochhar	Neena	Executive
102	De Haan	Lex	Executive
103	Hunold	Alexander	IT
104	Ernst	Bruce	IT
105	Austin	David	IT
106	Pataballa	Valli	IT
107	Lorentz	Diana	IT
108	Greenberg	Nancy	Finance
109	Faviet	Daniel	Finance
More than 10 rows available. Increase rows selector to view more rows.			

Kết nối ngoài

☒ Autocommit Display 10 ▼

```
SELECT  EMPLOYEE_ID, LAST_NAME, FIRST_NAME, DEPARTMENT_NAME
        FROM EMPLOYEES LEFT OUTER JOIN DEPARTMENTS
        ON EMPLOYEES.DEPARTMENT_ID = DEPARTMENTS.DEPARTMENT_ID;
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_NAME
100	King	Steven	Executive
101	Kochhar	Neena	Executive
102	De Haan	Lex	Executive
103	Hunold	Alexander	IT
104	Ernst	Bruce	IT
105	Austin	David	IT
106	Pataballa	Valli	IT
107	Lorentz	Diana	IT
108	Greenberg	Nancy	Finance
109	Faviet	Daniel	Finance
More than 10 rows available. Increase rows selector to view more rows.			

So sánh kết quả hai kiểu kết nối

- Kết nối trong trường v 106 b d li u
- Kết nối ngoài trường v 107 b d li u
- Giải thích:
 - Kết nối trong trường v kết quả khi tìm c DEPARTMENT_ID t ng ng c a nhân viên trong b ng DEPARTMENT
 - Kết nối ngoài trường v kết quả ngay c khi không tìm c DEPARTMENT_ID c a nhân viên trong b ng DEPARTMENT

Các hàm truy vấn nhóm (Aggregate Functions)

- Hàm truy vấn nhóm thực hiện trên một tập các bản ghi
- Một số hàm nhóm như :
 - **Max(column)** - Tìm giá trị lớn nhất trong cột **column**
 - **Min(column)** - Tìm giá trị nhỏ nhất trong cột **column**
 - **Avg(column)** - Tìm giá trị trung bình của cột **column**
 - **Count** – Hàm đếm số

Ví dụ: Hàm nhóm đơn giản

- Tìm mức lương thấp nhất, lương cao nhất, lương trung bình (làm tròn), và số nhân viên có trong bảng EMPLOYEES

```
SELECT MIN(SALARY), MAX(SALARY), ROUND(AVG(SALARY)), COUNT(*)  
FROM EMPLOYEES;
```

Results Explain Describe Saved SQL History

MIN(SALARY)	MAX(SALARY)	ROUND(AVG(SALARY))	COUNT(*)
2100	24000	6462	107

1 row returned in 0.00 seconds CSV Export

Hàm nhóm cùng với mệnh đề GROUP BY

- Tìm kiếm tổng, giá trị lớn nhất, giá trị nhỏ nhất, tỉ lệ trung bình cho các bộ phận và số lượng nhân viên trong từng bộ phận

```
SELECT DEPARTMENT_ID, MIN(SALARY), MAX(SALARY), ROUND(AVG(SALARY),2), COUNT(*)
FROM EMPLOYEES
GROUP BY DEPARTMENT_ID;
```

Results Explain Describe Saved SQL History

DEPARTMENT_ID	MIN(SALARY)	MAX(SALARY)	ROUND(AVG(SALARY),2)	COUNT(*)
100	6900	12000	8600	6
30	2500	11000	4150	6
-	7000	7000	7000	1
90	17000	24000	19333.33	3
20	6000	13000	9500	2
70	10000	10000	10000	1
110	8300	12000	10150	2
50	2100	8200	3475.56	45
80	6100	14000	8955.88	34
40	6500	6500	6500	1

Ngôn ngữ thao tác dữ liệu (DML)

- Ngôn ngữ thao tác dữ liệu (DML – Data Manipulation Language) gồm các truy vấn cho phép thêm, sửa, xóa dữ liệu trong các bảng
- Bao gồm các truy vấn bắt đầu bằng từ khóa:
 - INSERT INTO - thêm dữ liệu mới vào bảng
 - UPDATE - cập nhật/sửa đổi dữ liệu trong bảng
 - DELETE - xóa dữ liệu trong bảng

Thêm dữ liệu mới vào bảng

- Cú pháp:

```
INSERT [INTO] TableName VALUES(val1,val2,...)
```

```
INSERT [INTO] TableName(column1,...,columnN) VALUES  
(val1,...,valN)
```

- Ví dụ :

```
INSERT INTO EMPLOYEES (EMPLOYEE_ID, FIRST_NAME, LAST_NAME,  
EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY,  
COMMISSION_PCT, MANAGER_ID, DEPARTMENT_ID)  
VALUES (921, 'Werdna', 'Leppo', 'leppo@whatever.com', null, SYSDATE,  
'IT_PROG', 15000, 0.0, 103, 60);
```

Thêm dữ liệu mới vào bảng

- Chương trình thực hiện thêm dữ liệu và thông báo kết quả như sau:

```
Autocommit Display 10
INSERT INTO EMPLOYEES
(EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID,
SALARY, COMMISSION_PCT, MANAGER_ID, DEPARTMENT_ID)
VALUES (921, 'Werdna', 'Leppo', 'leppo@whatever.com', null, SYSDATE,
'IT_PROG', 15000, 0.0, 103, 60);
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.28 seconds

Thêm dữ liệu mới vào bảng

- Dòng dữ liệu mới sẽ được thêm vào trong bảng EMPLOYEES

EMPLOYEES Create											
Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL											
Query Count Rows Insert Row											
EDIT	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
	921	Werdna	Leppo	leppo@whatever.com		14-JUL-11	IT_PROG	15000	0	103	60
	100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90
	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	-	102	60
	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	-	103	60
	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-97	IT_PROG	4800	-	103	60
	106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-98	IT_PROG	4800	-	103	60
	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	-	103	60
	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-94	FL_MGR	12000	-	101	100
	109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-94	FL_ACCOUNT	9000	-	108	100
	110	John	Chen	JCHEN	515.124.4269	28-SEP-97	FL_ACCOUNT	8200	-	108	100
	111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-SEP-97	FL_ACCOUNT	7700	-	108	100
	112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-MAR-98	FL_ACCOUNT	7800	-	108	100

Cập nhật dữ liệu trong bảng

- Cú pháp:

```
UPDATE TableName  
SET columnName= val  
[WHERE condition];
```

- Ví dụ :

```
UPDATE EMPLOYEES  
SET PHONE_NUMBER = '301.555.1212'  
WHERE EMPLOYEE_ID = 921;
```

Cập nhật dữ liệu trong bảng

- Kết quả :

```
Autocommit Display 10
UPDATE EMPLOYEES
SET  PHONE_NUMBER = '301.555.1212'
WHERE EMPLOYEE_ID = 921;
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.05 seconds



EDIT	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
	921	Werdna	Leppo	leppo@whatever.com	 301.555.1212 	14-JUL-11	IT_PROG	15000

Xóa dữ liệu trong bảng

- Cú pháp:

```
DELETE FROM TableName
```

```
[WHERE condition];
```

- Có thể xóa toàn bộ nội dung bảng bằng cú pháp:

```
DELETE FROM TableName ;
```

- Ví dụ :

```
DELETE FROM EMPLOYEES
```

```
WHERE EMPLOYEE_ID = 921;
```

Ngôn ngữ định nghĩa dữ liệu

- Ngôn ngữ định nghĩa dữ liệu (DDL – Data Definition Language) gồm các lệnh cho phép tạo ra, thay đổi hoặc xóa các bảng
- Chúng ta cũng có thể định nghĩa các khóa (key), chỉ mục (index), chế độ các liên kết giữa các bảng và thiết lập các quan hệ ràng buộc giữa các bảng trong CSDL

Ngôn ngữ định nghĩa dữ liệu

■ Các lệnh:

- CREATE TABLE: tạo mới bảng mới trong CSDL
- ALTER TABLE: thay đổi cấu trúc của bảng trong CSDL
- CREATE VIEW: tạo mới View mới
- CREATE INDEX: tạo chỉ mục (khóa tìm kiếm)
- DROP INDEX: xóa chỉ mục đã có

Lệnh CREATE TABLE

- CREATE TABLE table_name (
 column_name1 data_type[option],,
 column_name2 data_type[option],,
 );
- Trong đó:
 - table_name là tên bảng cần tạo,
 - column_name là tên các trường cần tạo,
 - data_type là kiểu dữ liệu tương ứng.
 - option là mệnh đề ràng buộc như giá trị mặc định, not null,...

Ví dụ lệnh CREATE TABLE

```
CREATE TABLE EMPLOYEE_INPUT (  
    EMPLOYEE_ID          NUMBER(6)          NOT NULL,  
    FIRST_NAME           VARCHAR2(20)        NULL,  
    LAST_NAME            VARCHAR2(25)        NOT NULL,  
    EMAIL                VARCHAR2(25)        NOT NULL,  
    PHONE_NUMBER         VARCHAR2(20)        NULL,  
    HIRE_DATE            DATE                NOT NULL,  
    JOB_ID               VARCHAR2(10)        NOT NULL,  
    SALARY               NUMBER(8,2)         NULL,  
    COMMISSION_PCT       NUMBER(2,2)         NULL,  
    MANAGER_ID           NUMBER(6)           NULL,  
    DEPARTMENT_ID        NUMBER(4)           NULL  
);
```

Lệnh ALTER TABLE

- Lệnh ALTER TABLE cho phép thay đổi các thuộc tính của bảng như :
 - Thêm/xóa cột trong bảng
 - Thay đổi kiểu dữ liệu cho các cột trong bảng
 - Thay đổi thuộc tính bảng như chủ khóa cho bảng
 - Thêm/xóa/thay đổi các ràng buộc

Lệnh ALTER TABLE

- Thêm m t c t:

```
ALTER TABLE table_name
```

```
    ADD COLUMN column_name    data_type;
```

- Xóa m t c t:

```
ALTER TABLE table_name
```

```
    DROP COLUMN column_name;
```

Lệnh ALTER TABLE

- Thêm mệnh đề ràng buộc Check:

```
ALTER TABLE table_name  
    ADD CONSTRAINT constraint_name  
    CHECK (condition);
```

- Ví dụ :

```
ALTER TABLE EMPLOYEES  
    ADD CONSTRAINT EMPLOYEES_CHK_SALARY_MIN  
    CHECK (SALARY > 0);
```


Lệnh ALTER TABLE

```
ALTER TABLE EMPLOYEE_INPUT
```

```
ADD CONSTRAINT EMP_INPUT_DEPT_FK
```

```
FOREIGN KEY (DEPARTMENT_ID)
```

```
REFERENCES DEPARTMENTS (DEPARTMENT_ID);
```

- Ví dụ này nhằm minh họa rằng bạn có thể tham chiếu có tên

EMP_DEPT_FK sẽ được thêm vào bảng **EMPLOYEES**

nhằm minh họa cột **DEPARTMENT_ID** là khóa ngoại liên kết

với cột khóa chính (**DEPARTMENT_ID**) của bảng

DEPARTMENTS

Lệnh ALTER TABLE

- Ví dụ ràng buộc UNIQUE:

```
ALTER TABLE EMPLOYEE_INPUT  
ADD CONSTRAINT EMPLOYEES_UNQ_EMAIL  
UNIQUE (EMAIL);
```

- Ràng buộc UNIQUE có thể xóa đi bằng lệnh ALTER

```
ALTER TABLE EMPLOYEE_INPUT  
DROP CONSTRAINT EMPLOYEES_UNQ_EMAIL;
```

- Lệnh DROP dùng để xóa bỏ đối tượng không còn cần thiết khỏi CSDL

- Ví dụ :

```
DROP TABLE EMPLOYEE_INPUT CASCADE CONSTRAINTS;
```

(mệnh CASCADE CONSTRAINTS được thêm vào để xóa bỏ tất cả các ràng buộc tham chiếu trong bảng)

- Gồm các lệnh cho phép thực hiện các quy tắc truy xuất dữ liệu cho ngôn ngữ sử dụng CSDL
- Các lệnh:
 - **GRANT**: Cung cấp các quy tắc (như tạo View – Create View, cập nhật - Update, xóa - Delete...) cho ngôn ngữ sử dụng trên CSDL hoặc trên các bảng.
 - **REVOKE**: Thu hồi các quy tắc đã cấp cho ngôn ngữ sử dụng trên CSDL hoặc trên các bảng.

■ Cú pháp:

- Cung cấp quyền trên CSDL:

```
GRANT privilege1, privilege2, ... TO database;
```

- Cung cấp quyền trên bảng:

```
GRANT privilege1, privilege2,...
```

```
ON database.TableName TO UserName;
```

- Ví dụ cấp quyền tạo View cho người dùng CSDL HR:

```
GRANT CREATE VIEW TO HR;
```

- Ví dụ cấp các quyền Insert, thêm, cập nhật dữ liệu trên bảng EMPLOYEES cho admin của CSDL HR:

```
GRANT SELECT, INSERT, UPDATE  
ON HR.EMPLOYEES TO HR_ADMIN;
```

- REVOKE cho phép thu hồi các quyền đã cấp ban đầu GRANT.

- Ví dụ :

- Thu hồi quyền tạo View cho người dùng CSDL HR:

```
REVOKE CREATE VIEW TO HR;
```

- Thu hồi các quyền Insert, thêm, cập nhật dữ liệu trên bảng EMPLOYEES cho admin của CSDL HR:

```
REVOKE SELECT, INSERT, UPDATE  
ON HR.EMPLOYEES TO HR_ADMIN;
```

- Ngôn ngữ SQL cho phép Tạo CSDL và cung cấp các Thao tác cho phép làm việc trên dữ liệu như Thêm, Sửa, hoặc Xóa dữ liệu.
- 4 nhóm mệnh đề chính của SQL:
 - Nhóm truy vấn dữ liệu (DQL) : Select
 - Nhóm định nghĩa dữ liệu (DDL): Create, Drop, Alter ...
 - Nhóm thao tác dữ liệu (DML): Insert, Delete, Update ...
 - Nhóm cấp quyền dữ liệu (DCL): Grant, Revoke, ...