

CHƯƠNG 4

NGÔN NGỮ TRUY VẤN DỮ LIỆU SQL (SQL- Structured Query Language)

chuc.nv@due.edu.vn

4.1. Giới thiệu chung về SQL

SQL là ngôn ngữ truy vấn dữ liệu quan hệ được tích hợp trong hầu hết các hệ quản trị CSDL hiện nay, xuất hiện đầu tiên 1970, do IBM phát triển. Năm 1987 ISO thừa nhận SQL là ngôn ngữ chuẩn xử lý dữ liệu quan hệ.

SQL Sub Languages

DDL (Data Definition Language): Ngôn ngữ định nghĩa dữ liệu

DML (Data Manipulation Language): Ngôn ngữ thao tác dữ liệu

DQL (Data Query Language): Ngôn ngữ truy vấn dữ liệu

DCL (Data Control Language): Ngôn ngữ kiểm soát dữ liệu

TCL (Transaction Control Language): Ngôn ngữ kiểm soát giao tác

SQL				
DDL	DML	DQL	DCL	TCL
Create Alter Drop Truncate	Insert Update Delete	Select	Grant Revoke	Commit Rollback Savepoint

4.1. Giới thiệu chung về SQL

Common data types

int|bigint|smallint|tinyint

decimal(n,d)

bit

char(n)|nchar(n)

varchar(n)|nvarchar(n)

date|dateTime

Date Functions

Getdate()

Day

Month

Year

DateDiff

String Functions

Left

Right

Len

STR

Substring

Maths

Logic

Comparative

Set Operators

Statistical Functions

+

And

>, >=

Union

Max|Min

-

Or

<,<=

Intersect

Sum

*

Not

=

Except

AVG

/

<>

In|Not in

Count

DDL (Data Definition Language)

Create table: Tạo bảng dữ liệu

Alter Table: Sửa đổi cấu trúc dữ liệu của bảng

Truncate Table: Xóa tất cả dữ liệu trong bảng

Drop Table: Xóa bảng dữ liệu

4.2. DDL (Data Definition Language)

CREATE TABLE: Tạo bảng dữ liệu

Syntax:

*Create Table <table_name>(attribute1 <data_type> [Not null]
[primary key], attribute2 <data_type>] [Not null] [primary key],...)*

Một số ràng buộc (Constraints) khi tạo bảng dữ liệu:

Not null

Primary key

Foreign key

Check

Unique

4.2. DDL (Data Definition Language)

Database: JOB_INFOR

Tables:

EMPLOYEES(Eno, Ename, Skill, Gender, Salary, DateOfBirth, Dno)

DEPARTMENTS(Dno, Dname, Location)

Khóa chính gạch dưới, khóa ngoại in nghiêng, Dname: không được trùng, Salary \geq 0

--Create Database JOB_INFOR

CREATE DATABASE JOB_INFOR

USE JOB_INFOR -- Change the Current Database

--Create table Departments

Create table Departments(Dno char(4) not null primary key,
Dname nvarchar(50) not null unique, Location nvarchar(100))

--Create table Employees

Create table Employees(Eno char(5) not null primary key,
Ename nvarchar(50) not null, Gender bit,
Skill nvarchar(20) not null, DateOfBirth date,
Salary int not null check (Salary \geq 0),
Dno char(4) not null,
Constraint FK_Dept Foreign key (Dno) References Departments(Dno))

4.2. DDL (Data Definition Language)

ALTER TABLE: Thay đổi cấu trúc bảng dữ liệu

Thêm cột vào bảng:

Alter Table <TableName>

Add <AttributeName> <DataType...>

Thêm cột HireDate kiểu Date vào bảng Employees

Alter Table Employees

Add HireDate **Date** not null

Xóa cột đã có:

Alter Table <TableName>

Drop Column <ColumnName>

Xóa cột HireDate trong bảng Employees

Alter table Employees

Drop Column HireDate

4.2. DDL (Data Definition Language)

Thêm ràng buộc (Add Constraint)

Syntax: Alter Table <table_name>

Add Constraint <constraint_name> <Primary Key|Foreign Key...>

Giả sử bảng giữa bảng Employees và bảng Departments chưa có khóa ngoại

Thêm ràng buộc khóa ngoại cho bảng Employees

Alter table Employees

Add constraint FK_Dept Foreign key (Dno)
references Departments(Dno)

Xóa ràng buộc:

Syntax: Alter table

Drop Constraint <constraint_name>

Xóa ràng buộc khóa ngoại trong bảng Employees

Alter table Employees

Drop constraint FK_Dept

4.2. DDL (Data Definition Language)

TRUNCATE TABLE: Xóa tất cả các dữ liệu trong bảng

Syntax: Truncate Table <table_name>

Xóa tất cả dữ liệu trong bảng Departments

Truncate table Departments

DROP TABLE: Xóa bảng dữ liệu

Syntax: Drop table <TableName>

Xóa bảng dữ liệu Departments

Drop table Departments

Chú ý: Khi thực hiện lệnh Truncate table và Drop table sẽ ảnh hưởng đến ràng buộc Foreign key.

4.3. DML (Data Manipulation Language)

DML (Data Manipulation Language)

Insert into: Thêm dữ liệu vào bảng

Update: Cập nhật dữ liệu

Delete: Xóa dữ liệu

4.3. DML (Data Manipulation Language)

Insert into: Thêm dữ liệu vào bảng

Syntax:

Insert into <table_name> (attribute_list)

Values (value_list)

Thêm dữ liệu vào bảng Departments

```
insert into Departments(Dno,Dname,Location)
values('D001', 'IT', 'Room A301'), ('D002','HR', 'Room B202')
```

Thêm dữ liệu vào bảng Employees

```
Insert into Employees (Eno, Ename, Gender, Skill,
DateOfBirth,Salary, Dno)
```

Values

```
('E0001', 'Khoa', 1, 'SQL', '1980-01-20', 900, 'D001'),
('E0002', 'Nga', 0, 'Excel', '1985-06-15', 600, 'D002')
```

4.3. DML (Data Manipulation Language)

Delete: Xóa dữ liệu trong bảng

Syntax:

Delete From <table_name>

[Where <condition>]

Xóa công việc có mã “DB01” ra khỏi bảng Jobs

```
Delete from Jobs  
where Jno = 'DB01'
```

Chú ý cần phân biệt lệnh Delete với Truncate table và Drop table

4.3. DML (Data Manipulation Language)

Update: Cập nhật dữ liệu

Syntax:

Update <table>

Set <attribute=expression>

[Where <condition>]

Tăng lương cho tất cả nhân viên lên 20%

Update Employees

set Salary = Salary*1.2

Đặt lại Skill và Salary của nhân viên có mã nhân viên là “E0001” thành “C#” và 950

Update Employees

set Skill='C#', Salary =950

where Eno='E0001'

4.3. DML (Data Manipulation Language)

Cấu trúc điều khiển IIF và CASE ...WHEN

Cấu trúc IIF

Syntax:

IIF(<condition>, <true_value>, <>false_value>)

Cấu trúc CASE ... WHEN

Syntax:

CASE

WHEN <condition_1> THEN <value_1>

WHEN <condition_2> THEN <value_2>

...

WHEN <condition_n> THEN <value_n>

[ELSE <value_n+1>]

END

4.3. DML (Data Manipulation Language)

Ví dụ về cấu trúc IIF và CASE ...WHEN

Cập nhật Salary theo Skill như sau:

SQL: 1000, Java: 900, C#: 800, khác:650

Sử dụng cấu trúc IIF

Update Employees

```
Set Salary = IIF(Skill='SQL', 1000, IIF(Skill='Java', 900,  
IIF(Skill='C#',800, 650)))
```

Sử dụng cấu trúc Case... when

Update Employees

set Salary =

Case

when Skill='SQL' **then** 1000

when Skill='Java' **then** 900

when Skill='C#' **then** 800

Else 650

End

4.4. DQL (Data Query Language)

Select: Tìm kiếm, tổng hợp dữ liệu

Syntax:

SELECT <attributes|*|expression>

FROM <table_list>

[WHERE <condition>

[GROUP BY <attribute_group>]

[HAVING <group_condition>]

[ORDER BY <attribute> ASC|DESC]

4.4. DQL (Data Query Language)

Cho biết Ename, Skill và Salary của các nhân viên nữ

```
Select Ename, Skill, Salary  
From Employees  
where Gender = 'False'
```

Cho biết thông tin các nhân viên có kỹ năng SQL

```
Select *  
From Employees  
where Skill = 'SQL'
```

Cho biết thông tin các nhân viên, sắp xếp giảm dần theo lương

```
Select *  
From Employees  
Order by Salary DESC  
(Sắp tăng dần: ASC, giảm dần: DESC)
```

4.4. DQL (Data Query Language)

Một số options trong mệnh đề Select:

*****: Tất cả các thuộc tính

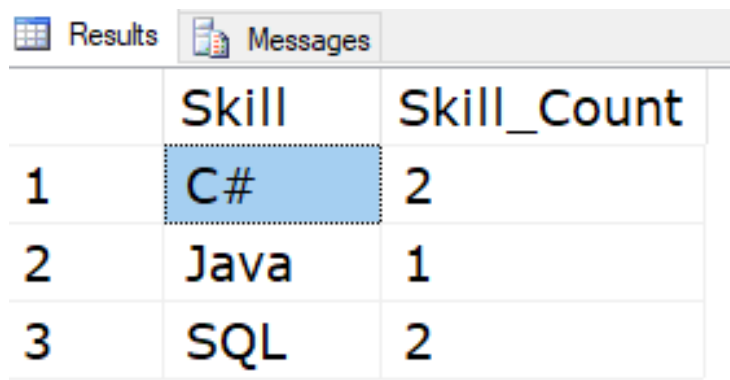
Distinct: Loại bỏ các dòng dữ liệu trùng nhau

Top n: Lấy n dòng đầu tiên trong kết quả truy vấn

Group by (xử lý dữ liệu theo nhóm)

Cho biết số nhân viên theo từng kỹ năng

```
Select Skill, Count(*) as Skill_Count  
From Employees  
Group by Skill
```



	Skill	Skill_Count
1	C#	2
2	Java	1
3	SQL	2

4.4. DQL (Data Query Language)

Having (Điều kiện theo nhóm)

Cho biết kỹ năng nào có từ 2 người trở lên

```
Select Skill, Count(*) as Skill_Count  
From Employees  
Group by Skill  
Having Count(*)>1
```

Results Messages		
	Skill	Skill_Count
1	C#	2
2	SQL	2

Cho biết số nhân viên, tổng lương, lương cao nhất, lương thấp nhất, lương TB theo từng kỹ năng

```
Select Skill, Count(*) as Num_emp, sum(Salary) as Sum_sal,  
max(Salary) as Max_Sal, min(Salary) as Min_Sal,  
AVG(Salary) as AVG_Sal  
From Employees  
Group by Skill
```

Results Messages						
	Skill	Num_emp	Sum_sal	Max_Sal	Min_Sal	AVG_Sal
1	C#	2	2000	1200	800	1000
2	Java	1	900	900	900	900
3	SQL	2	1650	1000	650	825

4.4. DQL (Data Query Language)

Các loại phép nối (Join types) in SQL

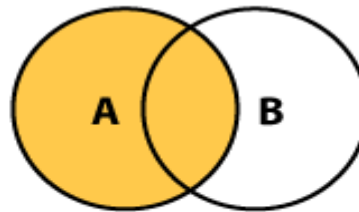
Inner Join

Left Outer Join

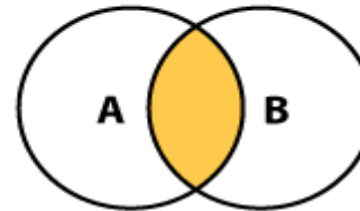
Right Outer Join

Full Outer Join

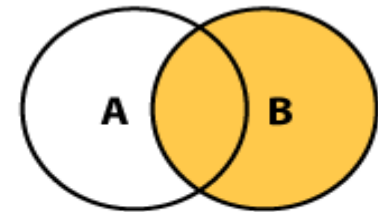
SQL JOINS



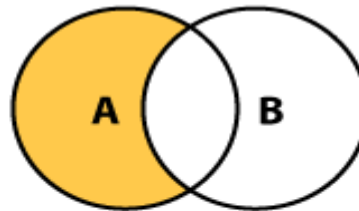
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key=B.Key
```



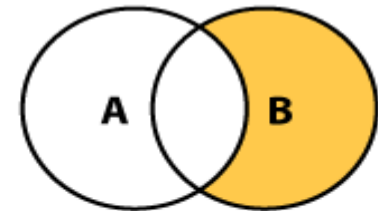
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key=B.Key
```



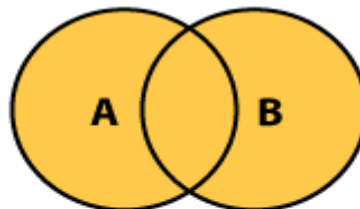
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key=B.Key
```



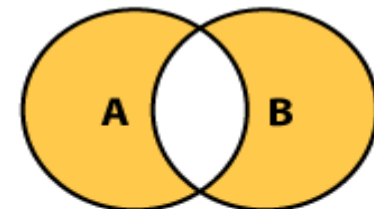
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key=B.Key  
WHERE B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key=B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key=B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key=B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

4.4. DQL (Data Query Language)

Join types in SQL

Giả sử có 2 bảng Employees và Jobs với cấu trúc và dữ liệu như sau:

```
Select * from Employees  
Select * from Jobs
```

100 % <							
Results Messages							
	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
1	E0001	Khoa	1	C#	1980-01-20	800	D001
2	E0002	Nga	0	Excel	1985-06-15	650	D002
3	E0003	Chuc	1	SQL	1975-03-18	1000	D001
4	E0004	Ngoc	0	Java	1985-03-15	900	D002
5	E0005	Chau	1	R	1975-02-18	1200	D001

	Jno	Jname	RequirementSkill	NumEmp
1	DB01	Database	SQL	7
2	DB02	Testing	Java	5
3	DB03	Programming	C#	10
4	DB04	Data Analysis	Python	4

4.4. DQL (Data Query Language)

```
Select * from Employees A inner join Jobs B  
On A.Skill = B.RequirementSkill
```

	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno	Jno	Jname	RequirementSkill	NumEmp
1	E0001	Khoa	1	C#	1980-01-20	800	D001	DB03	Programming	C#	10
2	E0003	Chuc	1	SQL	1975-03-18	1000	D001	DB01	Database	SQL	7
3	E0004	Ngoc	0	Java	1985-03-15	900	D002	DB02	Testing	Java	5

```
Select * from Employees A left join Jobs B  
On A.Skill = B.RequirementSkill
```

	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno	Jno	Jname	RequirementSkill	NumEmp
1	E0001	Khoa	1	C#	1980-01-20	800	D001	DB03	Programming	C#	10
2	E0002	Nga	0	Excel	1985-06-15	650	D002	NULL	NULL	NULL	NULL
3	E0003	Chuc	1	SQL	1975-03-18	1000	D001	DB01	Database	SQL	7
4	E0004	Ngoc	0	Java	1985-03-15	900	D002	DB02	Testing	Java	5
5	E0005	Chau	1	R	1975-02-18	1200	D001	NULL	NULL	NULL	NULL

4.4. DQL (Data Query Language)

```
Select * from Employees A Right join Jobs B  
On A.Skill = B.RequirementSkill
```

	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno	Jno	Jname	RequirementSkill	NumEmp
1	E0003	Chuc	1	SQL	1975-03-18	1000	D001	DB01	Database	SQL	7
2	E0004	Ngoc	0	Java	1985-03-15	900	D002	DB02	Testing	Java	5
3	E0001	Khoa	1	C#	1980-01-20	800	D001	DB03	Programming	C#	10
4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	DB04	Data Analysis	Python	4

```
Select * from Employees A full outer join Jobs B  
On A.Skill = B.RequirementSkill
```

	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno	Jno	Jname	RequirementSkill	NumEmp
1	E0001	Khoa	1	C#	1980-01-20	800	D001	DB03	Programming	C#	10
2	E0002	Nga	0	Excel	1985-06-15	650	D002	NULL	NULL	NULL	NULL
3	E0003	Chuc	1	SQL	1975-03-18	1000	D001	DB01	Database	SQL	7
4	E0004	Ngoc	0	Java	1985-03-15	900	D002	DB02	Testing	Java	5
5	E0005	Chau	1	R	1975-02-18	1200	D001	NULL	NULL	NULL	NULL
6	NULL	NULL	NULL	NULL	NULL	NULL	NULL	DB04	Data Analysis	Python	4

4.4. DQL (Data Query Language)

```
Select * from Employees A full outer join Jobs B
On A.Skill = B.RequirementSkill
where A.Skill is null
```

Results Messages		Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno	Jno	Jname	RequirementSkill	NumEmp
1		NULL	NULL	NULL	NULL	NULL	NULL	NULL	DB04	Data Analysis	Python	4

```
Select * from Employees A full outer join Jobs B
On A.Skill = B.RequirementSkill
where A.Skill is null or B.RequirementSkill is null
```

Results Messages		Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno	Jno	Jname	RequirementSkill	NumEmp
1		E0002	Nga	0	Excel	1985-06-15	650	D002	NULL	NULL	NULL	NULL
2		E0005	Chau	1	R	1975-02-18	1200	D001	NULL	NULL	NULL	NULL
3		NULL	NULL	NULL	NULL	NULL	NULL	NULL	DB04	Data Analysis	Python	4

4.4. DQL (Data Query Language)

Set Operators

```
Select RequirementSkill from Jobs  
Union  
Select Skill from Employees  
Intersect  
--  
Select Skill from Employees  
Intersect  
Select RequirementSkill from Jobs
```

```
Select Skill, Count(*) as Skill_Count  
From Employees  
Group by Skill  
Union all  
Select 'Number of employees',  
Count(*) as Skill_Count  
from Employees
```

Results			Messages		
	Skill	Skill_Count			
1	Java	1			
2	C#	2			
3	SQL	2			
4	Number of employees	5			

DCL (Data Control Language)

Grant: Cấp phát quyền

Revoke: Thu hồi quyền

4.5. DCL (Data Control Language)

GRANT: Cấp phát quyền truy cập

Syntax:

GRANT <Privileges>

ON <Database Objects>

TO <Users>

[WITH GRANT OPTION]

Tham số WITH GRANT OPTION cho phép người dùng được cấp quyền có thể dùng quyền đó cấp cho người dùng khác.

```
Grant Select, Update, Delete, insert  
on jobs  
to khoa  
with grant option
```

4.5. DCL (Data Control Language)

REVOKE: Thu hồi quyền

Syntax:

REVOKE <Privileges>

ON <Database Objects>

FROM <Users>

[CASCADE]

```
Revoke Delete, insert
on jobs
from khoa
cascade
```

Tham số CASCADE dùng để thu hồi các quyền mà người bị thu hồi quyền đã cấp cho các người dùng khác .

4.6. TCL (Transaction Control Language)

Giao tác (transaction) là tập các lệnh SQL được kết hợp lại với nhau để thực thi một công việc xác định. Các lệnh trong một giao tác phải đảm bảo rằng **hoặc phải được thực thi trọn vẹn hoặc không một câu lệnh nào được thực thi.**

Các tính chất của Transaction (ACID)

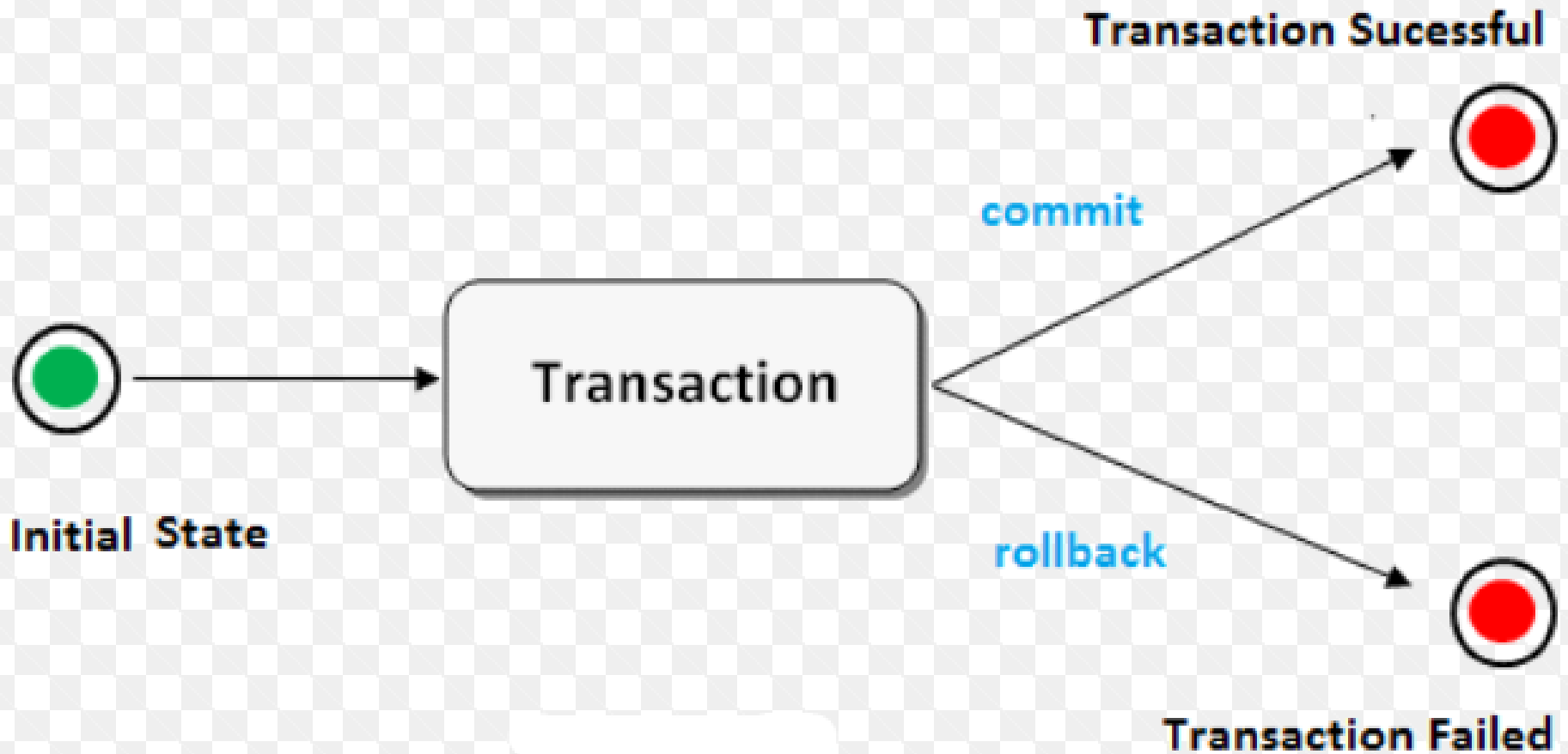
Atomicity (nguyên tố)

Consistency (nhất quán)

Isolation (cô lập)

Durability (bền vững)

4.6. TCL (Transaction Control Language)



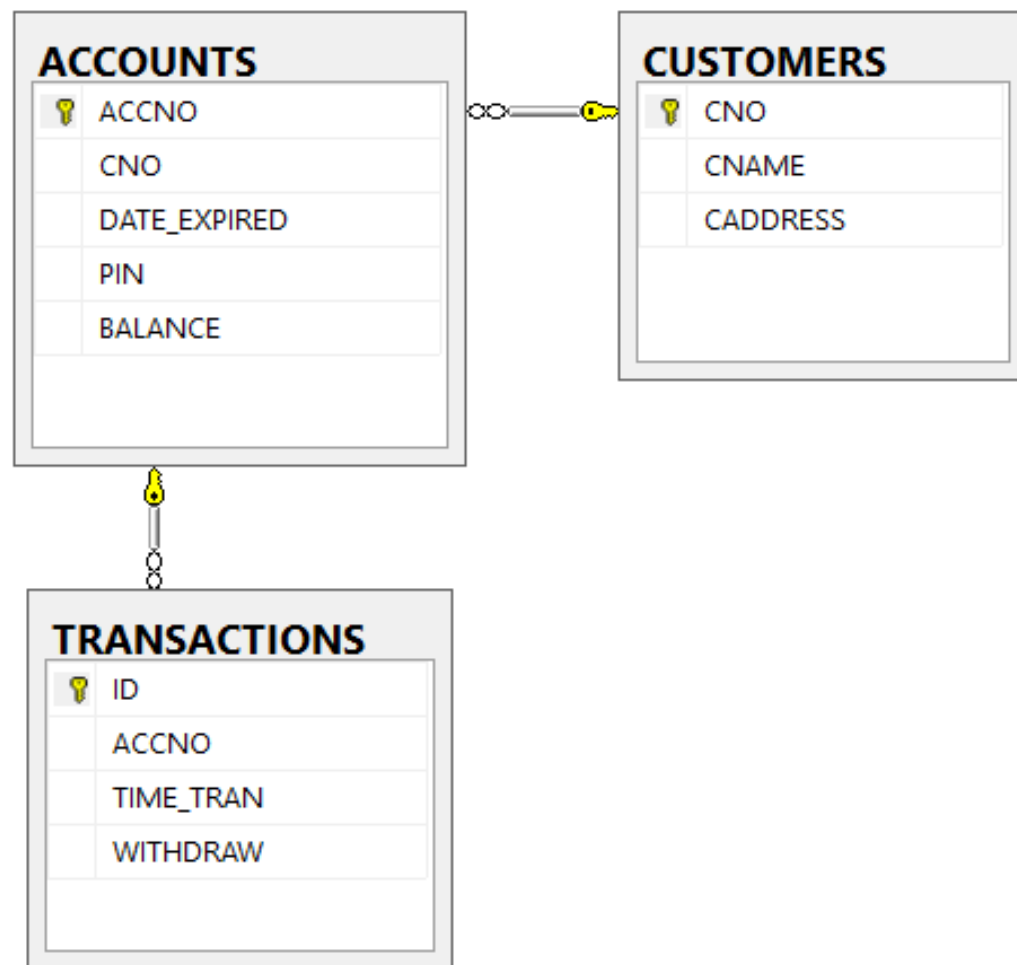
4.6. TCL (Transaction Control Language)

Minh họa các trường hợp của Transaction trong SQL

COMMIT TRANSACTION

ROLLBACK TRANSACTION

SAVE TRANSACTION



4.6. TCL (Transaction Control Language)

-- ROLLBACK CASE

BEGIN TRANSACTION T1

UPDATE ACCOUNTS

SET BALANCE = BALANCE-100

WHERE ACCNO='ACC01'

UPDATE ACCOUNTS

SET BALANCE = BALANCE+100

WHERE ACCNO='ACC02'

ROLLBACK TRANSACTION T1

-- nothing change on Accounts 'ACC01' and 'ACC02'

-- the transaction T1 Failed

4.6. TCL (Transaction Control Language)

--COMMIT CASE

BEGIN TRANSACTION T2

UPDATE ACCOUNTS

SET BALANCE = BALANCE-100

WHERE ACCNO='ACC01'

UPDATE ACCOUNTS

SET BALANCE = BALANCE+100

WHERE ACCNO='ACC02'

COMMIT TRANSACTION T2

--update completed, 100 transfered from Account
ACC01 to ACC02

--the transaction T2 Successful

4.6. TCL (Transaction Control Language)

-- SAVE POINT TRANSACTION

BEGIN TRANSACTION T3

UPDATE ACCOUNTS SET BALANCE=100 WHERE ACCNO = 'ACC01'

SAVE TRANSACTION SP1

UPDATE ACCOUNTS SET BALANCE=200 WHERE ACCNO = 'ACC02'

SAVE TRANSACTION SP2

UPDATE ACCOUNTS SET BALANCE=300 WHERE ACCNO = 'ACC03'

ROLLBACK TRANSACTION SP1

UPDATE ACCOUNTS SET BALANCE=400 where ACCNO = 'ACC04'

COMMIT TRANSACTION T3

--nothing change on Accounts 'ACC02' and 'ACC03'

4.7. Một số kỹ thuật truy vấn trong SQL

Select ... into : Tạo bảng dữ liệu từ kết quả truy vấn Select

```
Select * into SQL_EMPS  
from employees  
where Skill='SQL'
```

Create View (tạo khung nhìn)

Syntax: Create View view_name AS Select Clause

```
Create view SQL_View AS  
Select * from Employees  
where Skill='SQL'  
Select * from SQL_View
```

Results		Messages					
	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
1	E0002	Nga	0	SQL	1985-06-15	650	D002
2	E0003	Chuc	1	SQL	1975-03-18	1000	D001

4.7. Một số kỹ thuật truy vấn trong SQL

Sub Queries:

Cho biết thông tin của người có lương cao nhất

```
Select * from Employees  
where Salary =(select Max(Salary) from Employees)
```

Hoặc có thể viết

```
Select top 1 * from Employees  
order by Salary DESC
```

Cho biết thông tin các nhân viên có kỹ năng mà dự án DB01, DB02 cần

```
Select * from Employees  
where Skill in (Select RequirementSkill from Jobs  
where Jno='DB01' or Jno ='DB02')
```

4.7. Một số kỹ thuật truy vấn trong SQL

Sub Queries:

Tăng lương cho 2 người có lương thấp nhất lên 20%

Update Employees

set Salary =Salary*1.2

where Eno in (**Select top** 2 Eno **from** Employees
order by Salary **ASC**)

Hoặc có thể viết

Update Employees

set Employees.Salary = T1.Salary*1.2

from (**select top** 2 * **from** Employees **Order by** Salary
ASC) **as** T1

where Employees.Eno=T1.Eno

4.7. Một số kỹ thuật truy vấn trong SQL

Self Join

Cho biết thông tin các nhân viên có Skill trùng với ít nhất 1 người trong công ty

```
Select T1.*  
From Employees as T1, Employees as T2  
where T1.Skill = T2.Skill and T1.Eno<>T2.Eno
```

Cho biết thông tin các nhân viên có Skill không trùng với bất kỳ ai trong công ty

```
Select * from Employees  
Except  
Select T1.*  
From Employees as T1, Employees as T2  
where T1.Skill = T2.Skill and T1.Eno<>T2.Eno
```

4.7. Một số kỹ thuật truy vấn trong SQL

Like Operator

- The percent sign (%): represents zero, one or multiple characters
- The underscore (_): represents a single character

```
Select * from Employees  
where Skill like '%SQL%'
```

	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
1	E0002	Nga	0	SQL	1985-06-15	720	D002
2	E0003	Chuc	1	SQL	1975-03-18	1000	D001
3	E0006	Thanh	1	mySQL	1979-02-18	1100	D001

```
Select * from Employees  
where Skill like '__L'
```

```
Select * from Employees  
where Ename like 'C%'
```

	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
1	E0002	Nga	0	SQL	1985-06-15	720	D002
2	E0003	Chuc	1	SQL	1975-03-18	1000	D001

	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
1	E0003	Chuc	1	SQL	1975-03-18	1000	D001
2	E0005	Chau	1	C#	1975-02-18	1200	D001

4.7. Một số kỹ thuật truy vấn trong SQL

--Using Between ... and operator

```
Select * from Employees  
where Salary Between 1000 and 1500
```

Results		Messages					
	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
1	E0003	Chuc	1	SQL	1988-02-14	1000	D001
2	E0005	Chau	1	C#	1985-02-20	1200	D001
3	E0006	Thanh	1	mySQL	1982-02-18	1100	D001

```
Select * from Employees  
where DateOfBirth Between '1985-02-10' and '1985-02-20'
```

Results		Messages					
	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
1	E0002	Nga	0	SQL	1985-02-15	720	D002
2	E0005	Chau	1	C#	1985-02-20	1200	D001

4.7. Một số kỹ thuật truy vấn trong SQL

Using Column Alias in SELECT Statement

```
Select Ename,  
Case  
when gender = 'True' then 'Male'  
else 'Female'  
end as Gender, DateOfBirth,  
DateDiff(Year, DateOfBirth, Getdate()) as Age, Skill, Salary  
from Employees
```

	Ename	Gender	DateOfBirth	Age	Skill	Salary
1	Khoa	Male	1980-01-20	38	C#	960
2	Nga	Female	1985-06-15	33	SQL	720
3	Chuc	Male	1988-02-14	30	SQL	1000
4	Ngoc	Female	1989-03-15	29	Java	900
5	Chau	Male	1985-02-18	33	C#	1200
6	Thanh	Male	1982-02-18	36	mySQL	1100