

# INSERT – UPDATE - DELETE

---

NGUYEN Hong Phuong

Email: [phuongnh@soict.hust.edu.vn](mailto:phuongnh@soict.hust.edu.vn)

Site: <https://users.soict.hust.edu.vn/phuongnh>

# Contents

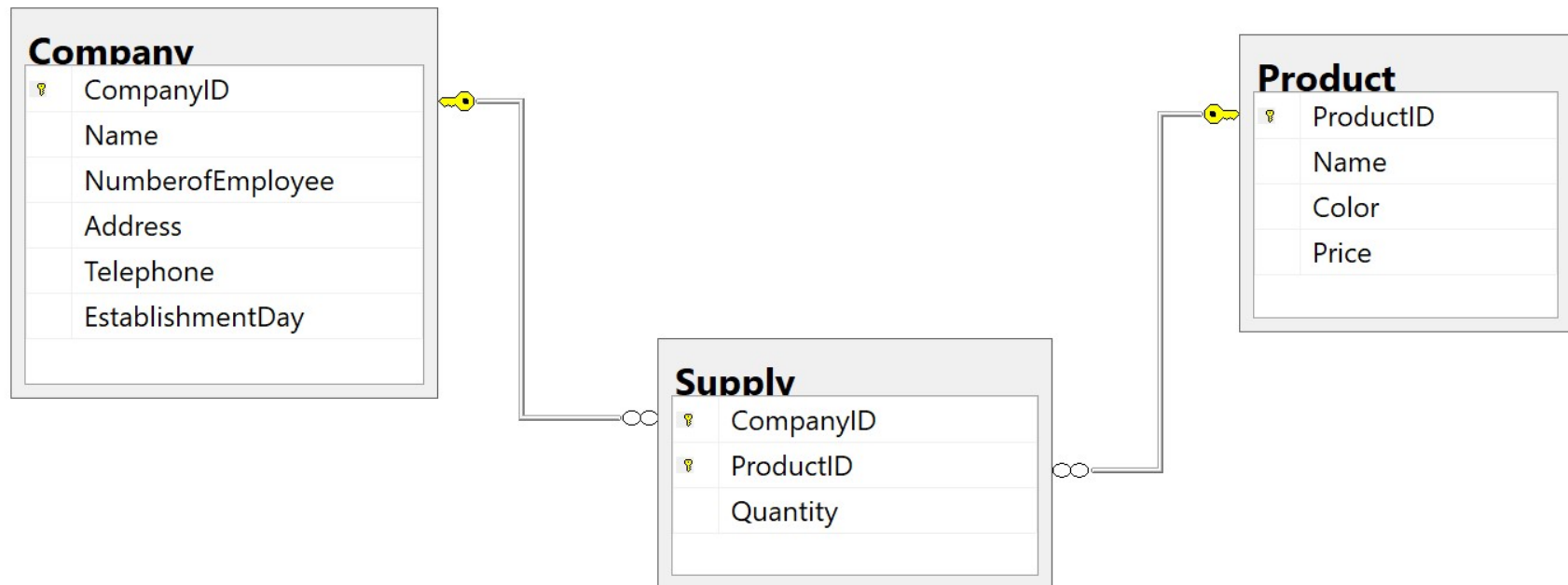
---

- Insert
- Update
- Delete
- ON UPDATE CASCADE ON DELETE CASCADE

# Database sample

---

- ❑ Company-Supply-Product
- ❑ Database diagram



# INSERT INTO Statement

---

□ The INSERT INTO statement is used to insert new records in a table.

□ Syntax

- Specify both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

- Add values for all the columns of the table, do not need to specify the column names => make sure the order of the values is in the same order as the columns in the table.

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

# INSERT INTO Statement (cont'd)

---

## □ Example

```
INSERT INTO Product(Name, Color, Price)
VALUES('Lexus ES 250', 'black', 15000)
```

```
INSERT INTO Product(Name, Color, Price)
VALUES('Lexus GS Turbo', 'red', 22000),
('Lexus ES 350', 'black', 20000),
('Lexus LS 500h', 'blue', 30000)
```

```
INSERT INTO Supply
VALUES(1, 29, 3000),
(2, 30, 2500),
(14, 31, 4000)
```

# INSERT INTO Statement (cont'd)

---

- Using the INSERT statement to copy all records from a table (table1) to a new table (table2)
  - table1 and table2 must have the same column with the same datatype

```
INSERT INTO table2  
SELECT * FROM table1
```

## □ Example

```
CREATE TABLE [Company2] (  
    [CompanyID] int,  
    [Name] varchar(40),  
    [NumberOfEmployee] int,  
    [Address] varchar(50),  
    [Telephone] char(15),  
    [EstablishmentDay] date,  
    PRIMARY KEY ([CompanyID])  
);
```

```
INSERT INTO Company2  
SELECT * FROM Company
```

# UPDATE Statement

---

- The UPDATE statement is used to modify the existing records in a table.
- Syntax

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

- Example

```
UPDATE Product  
SET Color = 'white'  
WHERE Name LIKE 'Lexus ES 250'
```

# DELETE Statement

---

□ The DELETE statement is used to delete existing records in a table.

□ Syntax

```
DELETE FROM table_name
WHERE condition;
```

□ Example

```
DELETE FROM Product
WHERE Name LIKE 'Lexus LS 500h'
```



# DELETE Statement (cont'd)

---

- What happens with the following command?

```
DELETE FROM Product  
WHERE ProductID = 1
```



## **The order of DELETE command:**

Delete the records of the table created last  
then delete the records of the table created first



Or, when creating tables, use option

**ON UPDATE CASCADE**  
**ON DELETE CASCADE**

# ON UPDATE CASCADE ON DELETE CASCADE

---

## □ ON UPDATE CASCADE ON DELETE CASCADE

- DELETE CASCADE: When we create a foreign key using this option, it deletes the referencing rows in the child table when the referenced row is deleted in the parent table which has a primary key.
- UPDATE CASCADE: When we create a foreign key using UPDATE CASCADE the referencing rows are updated in the child table when the referenced row is updated in the parent table which has a primary key.

# ON UPDATE CASCADE ON DELETE CASCADE

## □ Using SQL Server Management Studio

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the 'Keys' folder under the 'dbo.Supply' table, with three keys listed: 'PK\_Supply\_F6D7D022DE97D66', 'FK\_Supply\_CompanyL\_145C0A3F', and 'FK\_Supply\_ProductL\_15502E78'. A red box highlights the 'FK\_Supply\_ProductL\_15502E78' key, and a red arrow points to it with the text 'Right click and select "Modify"'. The main window shows the 'Foreign Key Relationships' dialog box for the selected relationship. The 'Editing properties for existing relationship' section shows the 'Foreign Key Column CompanyID' and 'Primary/Unique Key Company' details. Under the 'Table Designer' section, the 'Enforce Foreign Key C Yes' checkbox is checked. The 'INSERT And UPDATE S' section is expanded, and the 'Delete Rule' and 'Update Rule' are both set to 'No Action'. A red box highlights this section, and a red arrow points to it. The 'Column Properties' window at the bottom shows the properties for the 'CompanyID' column, including '(General)', '(Table Designer)', and '(General)' sections.

Column Name	Data Type	Allow Nulls
CompanyID	int	<input type="checkbox"/>
ProductID	int	<input type="checkbox"/>
Quantity	int	<input checked="" type="checkbox"/>

Foreign Key Relationships

Selected Relationship:  
FK\_Supply\_CompanyL\_145C0A3F  
FK\_Supply\_ProductL\_15502E78

Editing properties for existing relationship.

Foreign Key Column CompanyID  
Primary/Unique Key Company  
Primary/Unique Key CompanyID

**Identity**  
(Name) FK\_Supply\_CompanyL\_145C0A3F  
Description

**Table Designer**  
Enforce For Replicator Yes  
Enforce Foreign Key C Yes

**INSERT And UPDATE S**  
Delete Rule No Action  
Update Rule No Action

Column Properties

(General)  
(Name) CompanyID  
Allow Nulls No  
Data Type int  
Default Value or Binding

**Table Designer**  
Collation <database default>

(General)

# ON UPDATE CASCADE ON DELETE CASCADE

---

## □ Using T-SQL

```
ALTER TABLE [Supply] DROP CONSTRAINT FK__Supply__CompanyI__145C0A3F
```

```
ALTER TABLE [Supply] DROP CONSTRAINT FK__Supply__ProductI__15502E78
```

```
ALTER TABLE [Supply] ADD CONSTRAINT FK_Supply_Company  
FOREIGN KEY (CompanyID) REFERENCES Company(CompanyID)  
ON UPDATE CASCADE ON DELETE CASCADE
```

```
ALTER TABLE [Supply] ADD CONSTRAINT FK_Supply_Product  
FOREIGN KEY (ProductID) REFERENCES Product(ProductID)  
ON UPDATE CASCADE ON DELETE CASCADE
```

# How to know constraints in a table?

---

```
SELECT *  
FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS  
WHERE TABLE_NAME = 'Supply'
```