



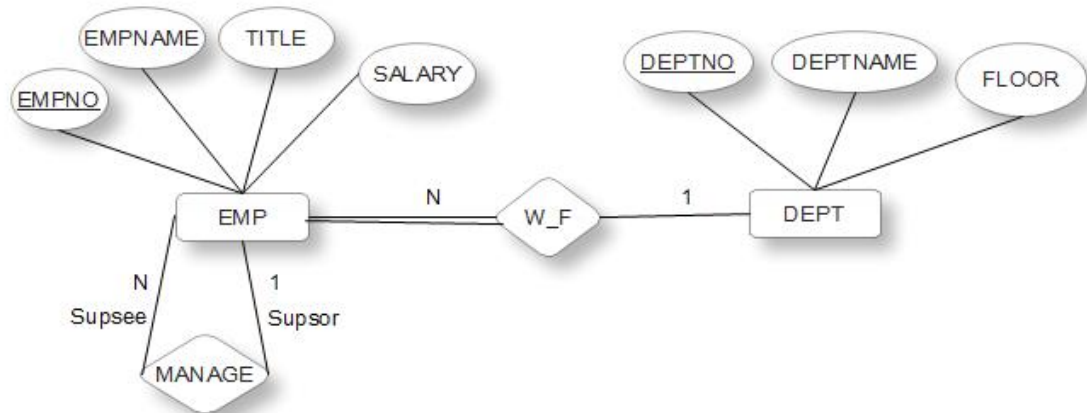
Database Systems

ER-WIN, MS SQL SVR, SSMS, and
SQL Introduction

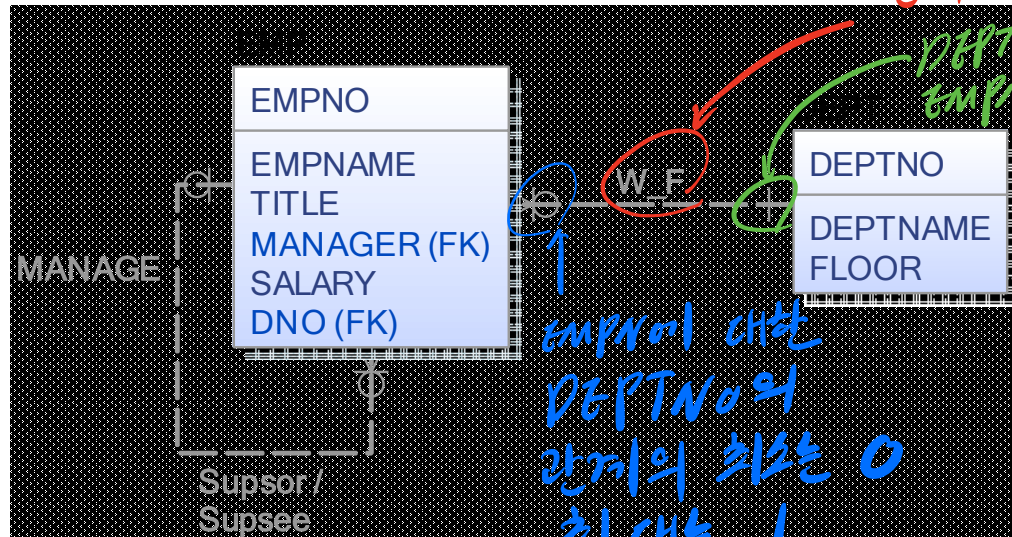


인하대학교

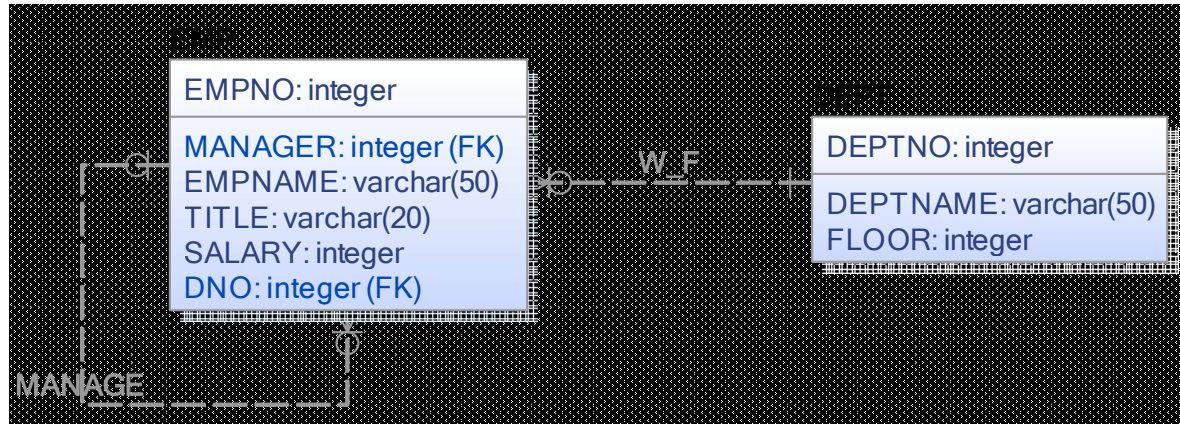
ER-Diagram for Sample DB



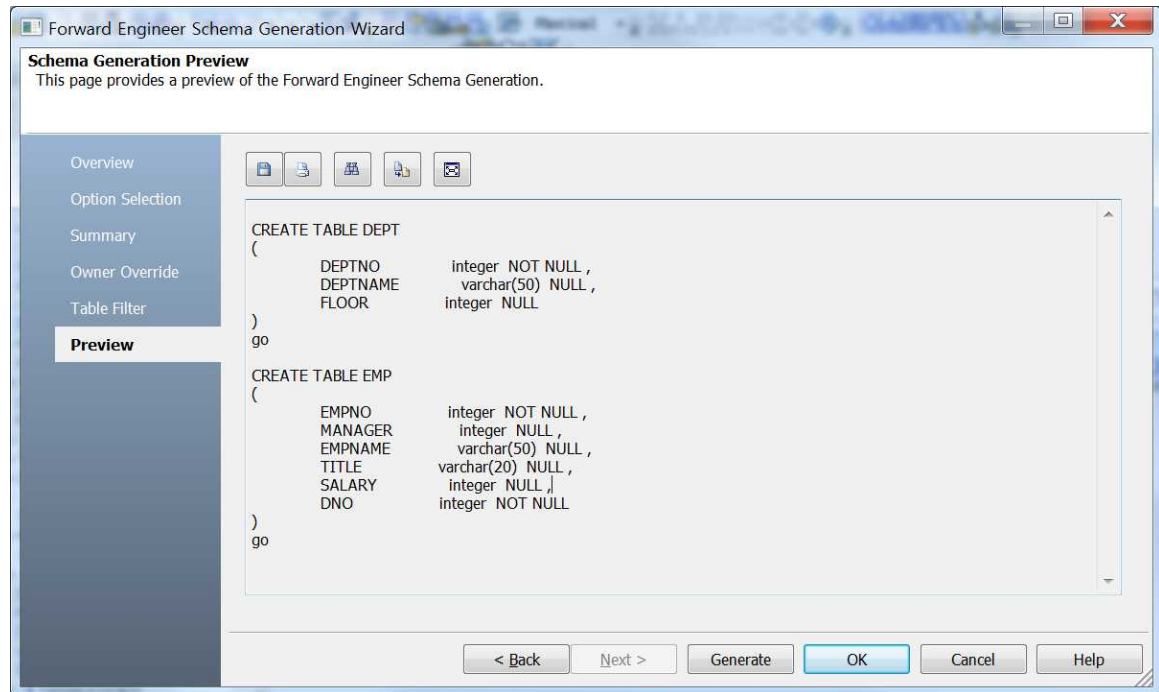
ER-WIN Diagram for Sample DB



ER-WIN Diagram for Sample DB



ER-WIN Diagram for Sample DB

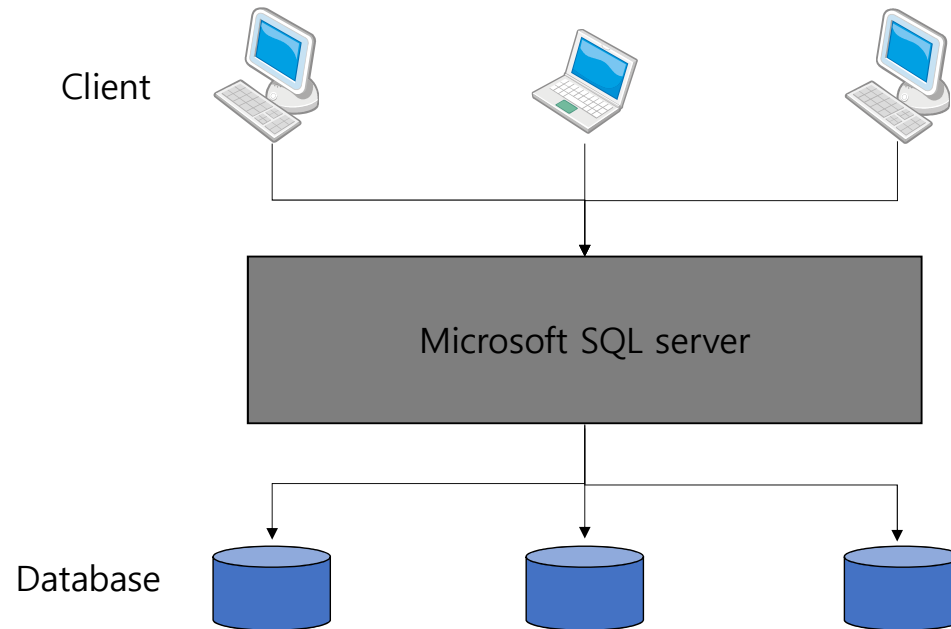




Microsoft SQL server

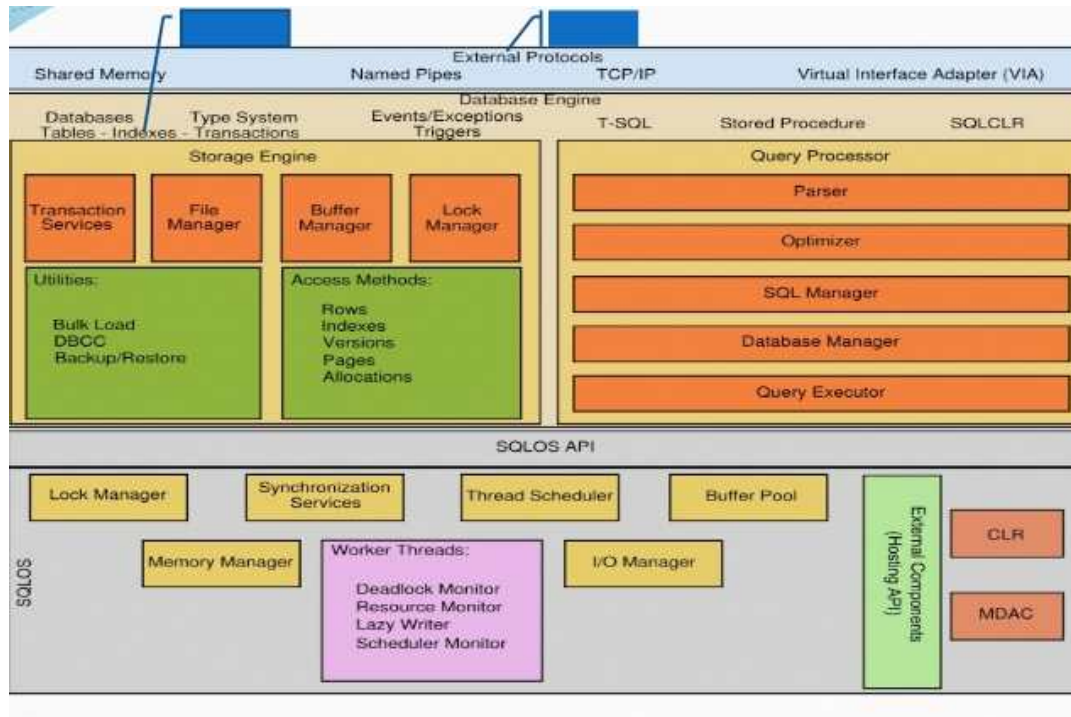
- A relational database management system developed by Microsoft
- As a database server, it is a software product with the primary functions of **defining**, **modifying**, and **retrieving data** as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).

MS SQL server architecture



MS SQL server architecture (cont'd.)

- Internal structure of MS SQL sever (reference)



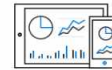
Install MS SQL server



개발자

SQL Server 2017 Developer는 비 프로덕션 환경에서 개발 및 테스트 데이터베이스로 사용하도록 라이선스가 제공되며 모든 기능을 갖춘 무료 버전입니다.

[지금 다운로드하기 >](#)



Express

SQL Server 2017 Express는 데스크톱 및 소형 서버 애플리케이션의 개발 및 제작에 적합한 무료 SQL Server 버전입니다.

[지금 다운로드하기 >](#)

- Microsoft SQL Server 2016 Express 다운로드 또는
- Microsoft SQL Server 2017 Express 다운로드
- <https://www.microsoft.com/ko-kr/sql-server/sql-server-downloads>



Attribute data types

• In MS-SQL SERVER...

Data type	Length	Description
bigint	8	Integer from -2^{63} (-9 223 372 036 854 775 808) to $2^{63}-1$ (9 223 372 036 854 775 807).
int	4	Integer from -2^{31} (-2 147 483 648) to $2^{31}-1$ (2 147 483 647).
smallint	2	Integer from -2^{15} (-32 768) to $2^{15}-1$ (32 767).
tinyint	1	Integer from 0 to 255.
bit	1 bit	Integer 0 or 1.
decimal(precision, scale)	5-17	Numeric data type with fixed precision and scale (accuracy 1-38, 18 by default and scale 0-p, 0 by default).
numeric	5-17	Same as data type 'decimal'.
money	8	Financial data type from -2^{63} (-922 337 203 685 477.5808) to $2^{63}-1$ (922 337 203 685 477.5807) with the precision of one ten-thousandth unit.
smallmoney	4	Financial data type from -2^{31} (-214 748.3648) to $2^{31}-1$ (214 748.3647) with the precision of one ten-thousandth unit.
float(n)	4-8	Numeric data type with float precision, where n is the number of mantis bits (1-24, accuracy of 7 digits, size of 4 bytes and 25-53, accuracy of 15 digits and size of 8 bytes).
real	4	Numeric data type with float precision that is defined as a float(24).
datetime	8	Data type representing date and time from 1.1.1753 to 31.12.9999 with precision about 3ms. Values are rounded to .000, .003 and .007.
smalldatetime	4	Data type representing date and time from 1.1.1900 to 6.6.2079 with precision of 1min. Values up to 29.998 are rounded down and values from 29.999 are rounded down to the nearest minute.

Attribute data types (cont'd.)

• In MS-SQL SERVER ...

Data type	Length	Description
char	n	Char string of fixed length and max. length of 8000 chars.
varchar	n	Char string of variable length and max. length of 8000 chars.
text	n	Char string of variable length and max. length of $2^{31}-1$ (2 147 483 647) chars.
nchar	2*n	Unicode char string of fixed length and max. length of 4000 chars.
nvarchar	2*n	Unicode char string of variable length and max. length of 4000 chars.
ntext	2*n	Unicode char string of variable length and max. length of $2^{30}-1$ (1 073 741 823) chars.
binary	n+4	Binary data of fixed length and max. length of 8000 bytes.
varbinary	n+4	Binary data of variable length and max. length of 8000 bytes.
image	n	Binary data of variable length and max. length of $2^{31}-1$ (2 147 483 647) bytes.
cursor		For storing the reference to cursors in a variable or in a procedure (no for CREATE TABLE).
sql_variant		For storing value of another type (no text , ntext , image , timestamp , sql_variant) of max. length to 8016 bytes. ODBC doesn't fully support this data type.
table		For storing the query result for the later usage.
timestamp	8+4	Data type generates automatically binary numbers, unique in the database, used mostly to the rows identification. There can be only column of this data type in the table.
uniqueidentifier		Data type for storing GUID (new by means of the NEWID function or existing from the string in the form xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, for example 6F9619FF-8B86-D011-B42D-00C04FC964FF).

Insert commands for EMP

use companyDB;

insert into EMP(EMPNO, EMPNAME, TITLE, MANAGER, SALARY, DNO)
values (2106, '김창섭', '대리', 1003, 2500000, 2);

insert into EMP(EMPNO, EMPNAME, TITLE, MANAGER, SALARY, DNO)
values (2106, '박영권', '과장', 4377, 3000000, 1);

insert into EMP(EMPNO, EMPNAME, TITLE, MANAGER, SALARY, DNO)
values (2106, '이수민', '부장', 4377, 4000000, 3);

insert into EMP(EMPNO, EMPNAME, TITLE, MANAGER, SALARY, DNO)
values (2106, '조민희', '과장', 4377, 3000000, 2);

insert into EMP(EMPNO, EMPNAME, TITLE, MANAGER, SALARY, DNO)
values (2106, '최종철', '사원', 3011, 1500000, 3);

insert into EMP(EMPNO, EMPNAME, TITLE, MANAGER, SALARY, DNO)
values (2106, '김상원', '사원', 3426, 1500000, 1);

insert into EMP(EMPNO, EMPNAME, TITLE, MANAGER, SALARY, DNO)
values (2106, '이성래', '사장', null, 5000000, 2);

Insert commands for DEPT

use companyDB;

insert into DEPT(DEPTNO, DEPTNAME, FLOOR) values (1, '영업', 8);
insert into DEPT(DEPTNO, DEPTNAME, FLOOR) values (2, '기획', 10);
insert into DEPT(DEPTNO, DEPTNAME, FLOOR) values (3, '개발', 9);
insert into DEPT(DEPTNO, DEPTNAME, FLOOR) values (4, '총무', 7);

Sample DB

EMPLOYEE

EMPNO	EMPNAME	TITLE	MANAGER	SALARY	DNO
2106	김창섭	대리	1003	2500000	2
3426	박영권	과장	4377	3000000	1
3011	이수민	부장	4377	4000000	3
1003	조민희	과장	4377	3000000	2
3427	최종철	사원	3011	1500000	3
1365	김상원	사원	3426	1500000	1
4377	이성래	사장	^	5000000	2

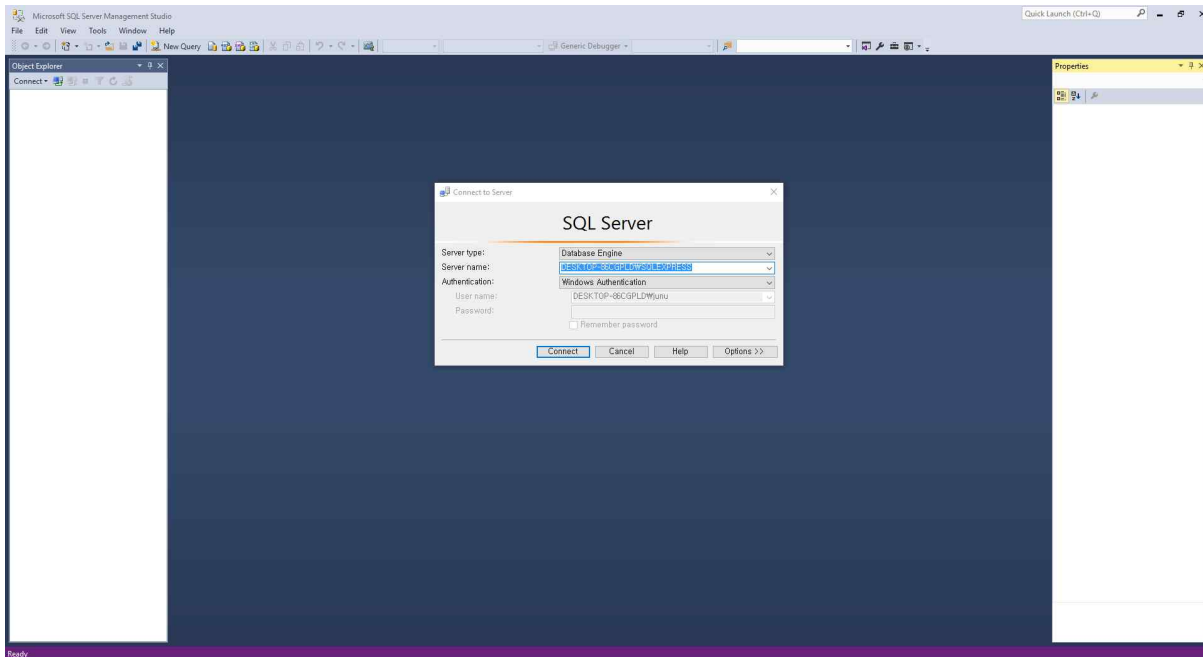
DEPARTMENT

DEPTNO	DEPTNAME	FLOOR
1	영업	8
2	기획	10
3	개발	9
4	총무	7

[그림 4.8] 관계 데이터베이스 상태

Microsoft SQL server

- Program execution
 - [start button] → [program] → [Microsoft SQL Server]
→ [Microsoft SQL Server Management Studio]



Basic screen configuration

- It consists of one menu bar for simple selection of commands and four component windows.

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The top menu bar includes File, Edit, Tools, Window, and Help. The main workspace is the Query Editor, which contains a SQL query: `SELECT * FROM EMPLOYEE;`. The left pane shows the Object Explorer with a tree view of the database structure, including System Databases, Database Snapshots, COMPANY_DB, Database Diagrams, Tables, System Tables, File Tables, and various user-defined tables like dbo.DEPARTMENT, dbo.EMPLOYEE, and dbo.LOCATION. The bottom pane is split into two sections: Results and Messages. The Results section shows a table with columns SSN, Super_SSN, Fname, Minit, Lname, Bdate, Address, Sex, Salary, and Dno, containing four rows of employee data. The Messages section shows a green status bar indicating 'Query executed successfully.' The right pane displays the Properties window for the current connection, showing details such as Connection name, Connection details, and Server information.

Query editor

Menu bar

Property page

object Explorer

Results / messages page

	SSN	Super_SSN	Fname	Minit	Lname	Bdate	Address	Sex	Salary	Dno
1	00000000	00000000	Mack	H	Ruddy	1980-05-17 00:00:00.000	610 Fondren, Houston, TX	M	55000	1
2	20170001	00000000	John	B	Smith	1965-01-09 00:00:00.000	731 Fondren, Houston, TX	M	30000	5
3	20170002	00000000	Mae	L	Kim	1995-05-22 00:00:00.000	381 Rouston, Houston, TX	F	15000	5
4	20170003	00000000	Kane	P	Wood	1994-12-01 00:00:00.000	555, Rouston, Houston, TX	M	15000	5

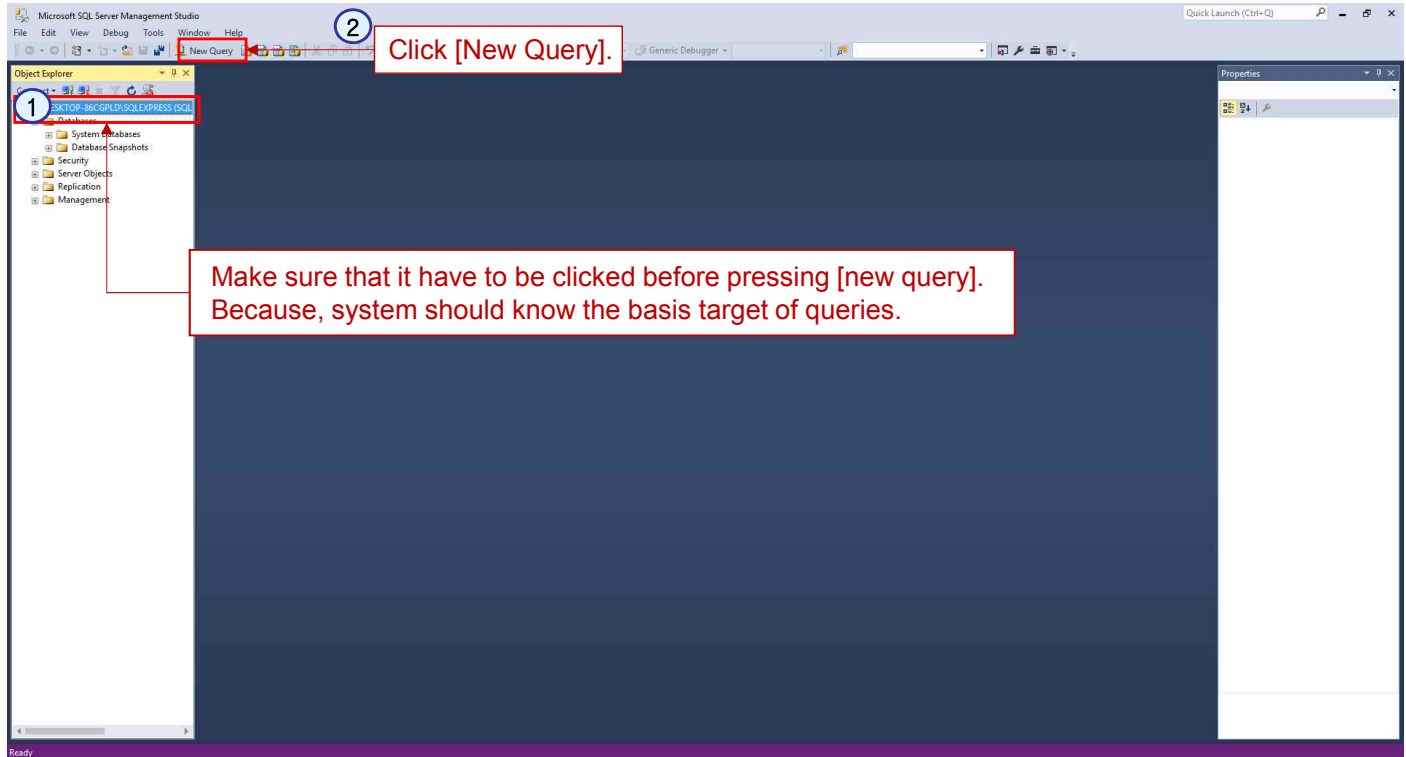
Basic screen configuration (cont'd.)

Division	Function
Menu bars	<ul style="list-style-type: none">➤ The buttons to help modeling works are gathered.➤ It can be located anywhere user want.
Query editor	<ul style="list-style-type: none">➤ By using the Database Engine Query Editor in SQL Server Management Studio you can write and edit queries as scripts.➤ You use scripts when you have to process Windows System commands and Transact-SQL statements in the same script.
Object Explorer	<ul style="list-style-type: none">➤ SQL Server Management Studio provides features for managing objects in instances of the Database Engine, Analysis Services, Integration Services, and Reporting Services.
Results / messages page	<ul style="list-style-type: none">➤ You can see results of query, messages.
Property page	<ul style="list-style-type: none">➤ Property page dialog boxes in Microsoft SQL Server Management Studio all use a common format displaying information with expanding and collapsing categories.➤ The fields shown depend on the particular property➤ Properties shown in gray are read-only.➤ Categorized and Alphabetic buttons are near the top of each property page.

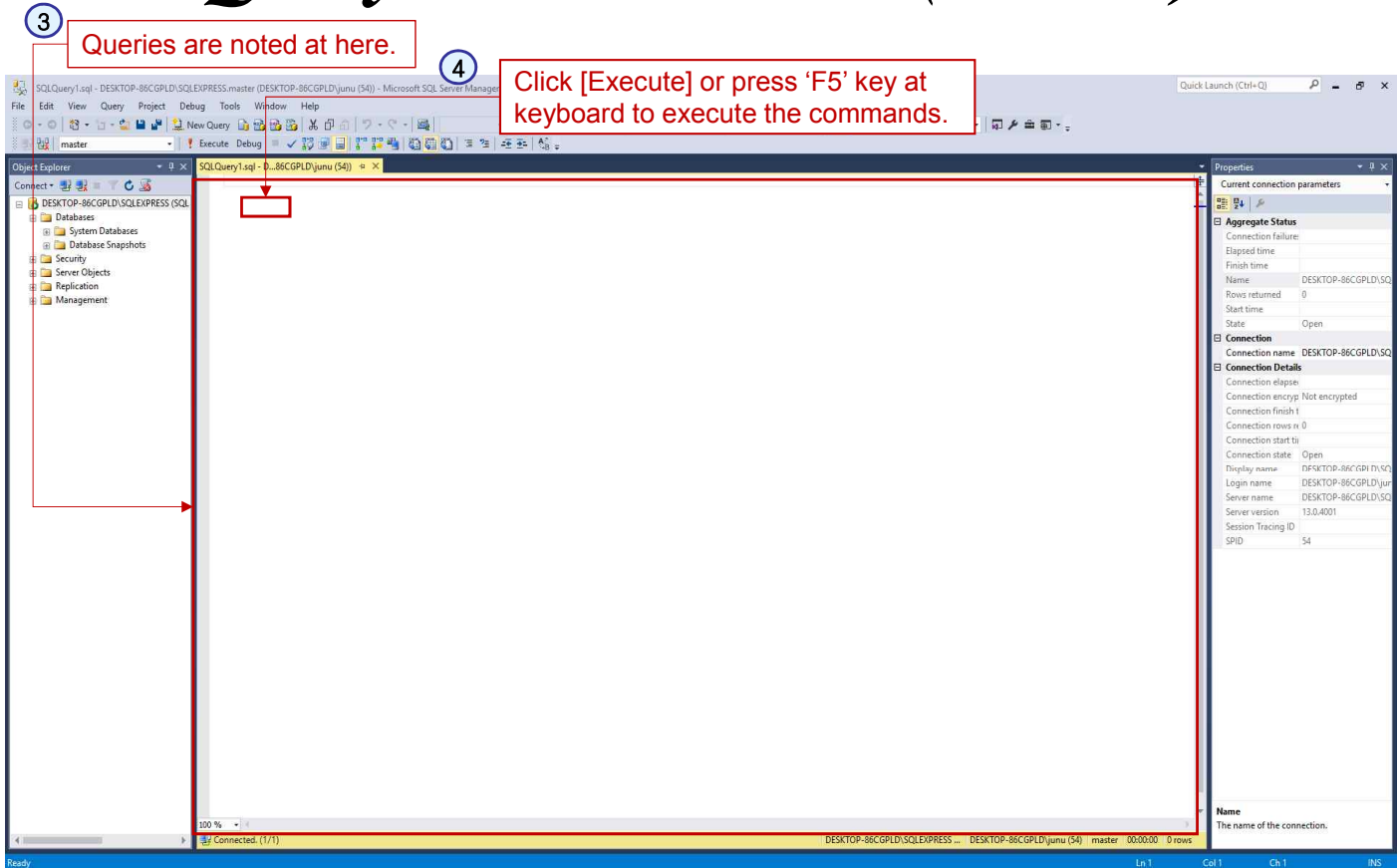
References : <https://docs.microsoft.com/en-us/sql/ssms/use-sql-server-management-studio>

Query editor window

- Query editor window can be created by following steps

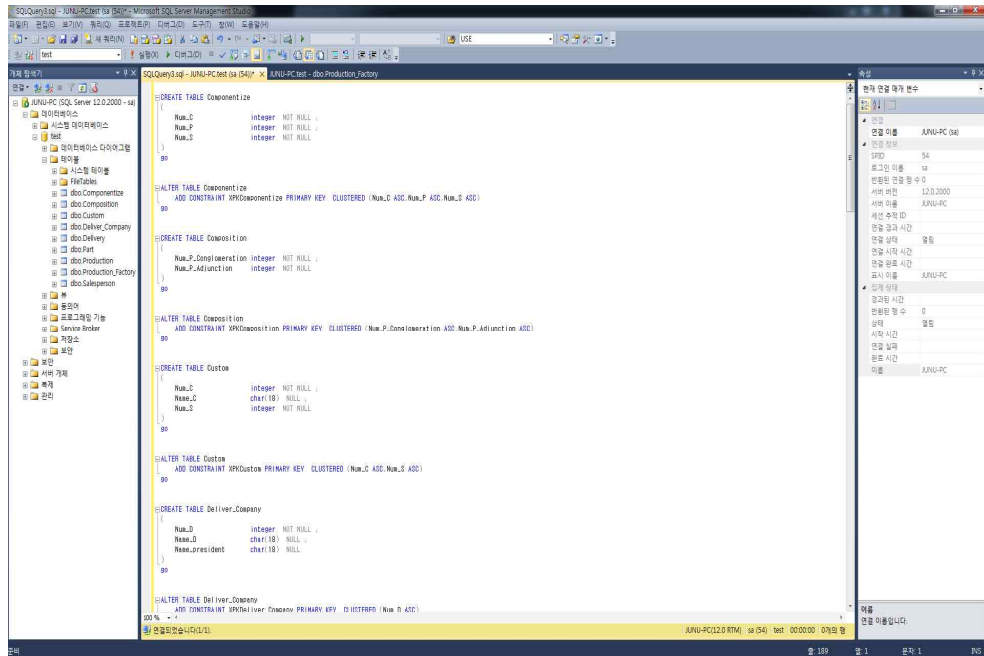


Query editor window (cont'd.)



How to use SSMS (SQL server management studio) tool

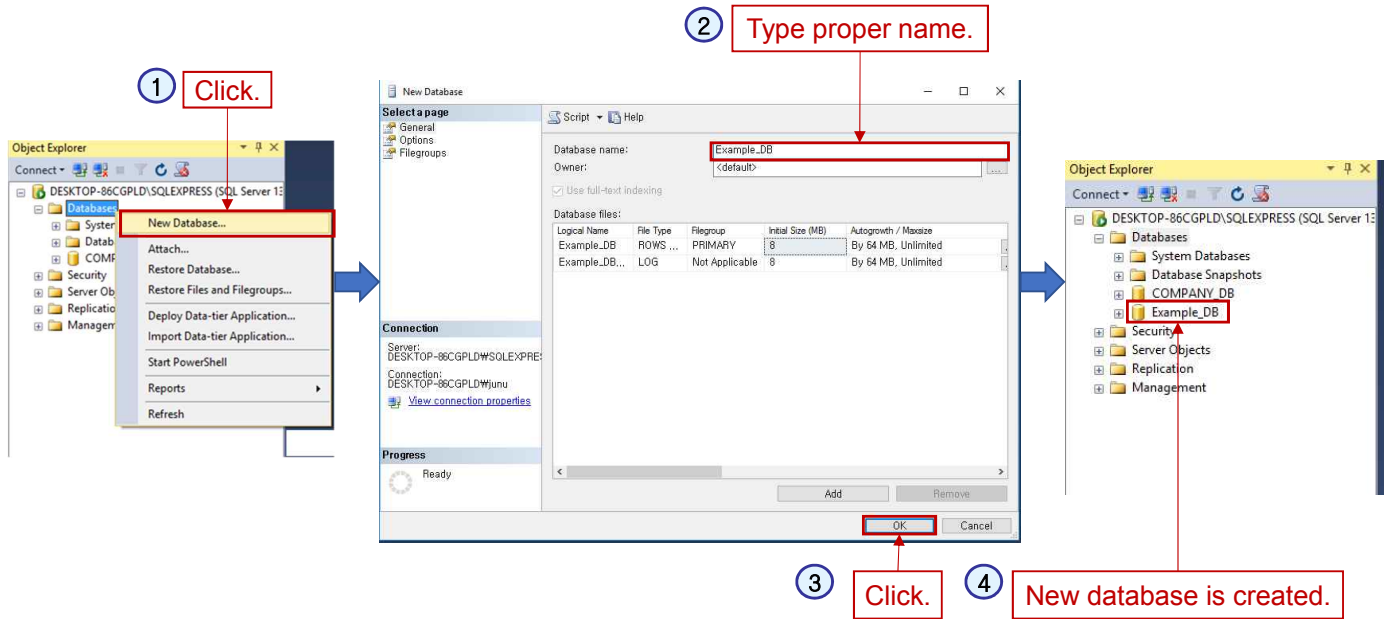
- Using GUI



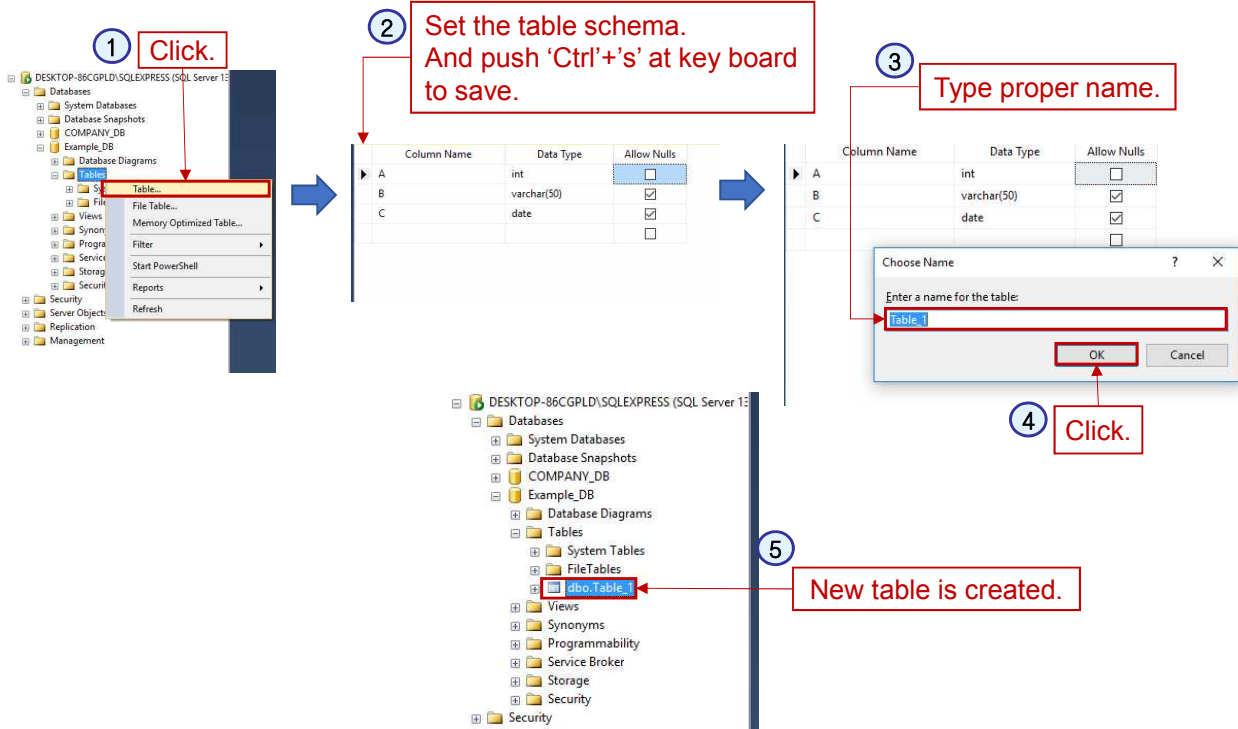
- Management 기능과 쿼리분석기가 통합된 관리 툴
- SSMS의 기능
 - 서버관리, 데이터베이스 생성/변경, DB백업 및 복구, 데이터 전송 및 변환



Create database

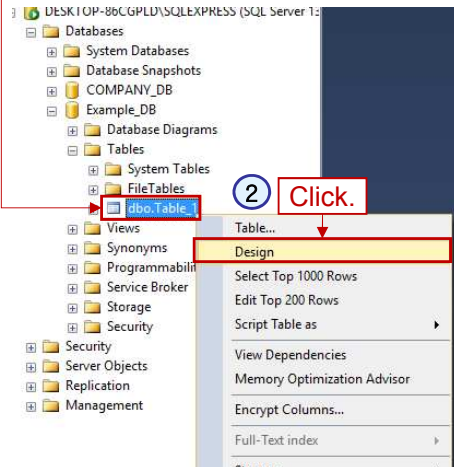


Create table



Modify table design

1 Click right-side button.



2 Click.

3 Modify table schema.

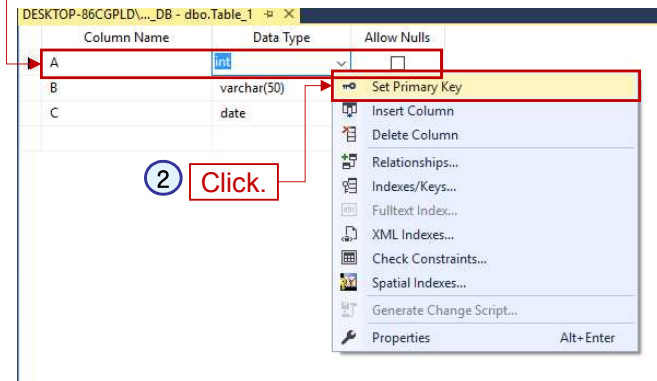


DESKTOP-86CGPLD\..._DB - dbo.Table_1

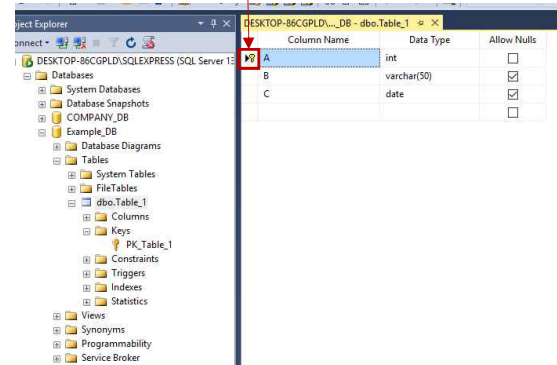
Column Name	Data Type	Allow Nulls
A	int	<input type="checkbox"/>
B	varchar(50)	<input checked="" type="checkbox"/>
C	date	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Set primary key

- 1 Click right-side button anywhere in row of attribute that you want to set primary key.



- 3 You can see key mark.



Set foreign key

- How to connect attribute 'A_' of 'Table_2' into attribute 'A' of 'Table_1':

DESKTOP-86CGPLD\..._DB - dbo.Table_2			DESKTOP-86CGPLD\..._DB - dbo.Table_1		
Column Name	Data Type	Allow Nulls	Column Name	Data Type	Allow Nulls
A	int	<input type="checkbox"/>			
B	varchar(50)	<input checked="" type="checkbox"/>			
C	date	<input checked="" type="checkbox"/>			
		<input type="checkbox"/>			

DESKTOP-86CGPLD\..._DB - dbo.Table_2			DESKTOP-86CGPLD\..._D		
Column Name	Data Type	Allow Nulls	Column Name	Data Type	Allow Nulls
D	int	<input type="checkbox"/>			
E	nchar(10)	<input checked="" type="checkbox"/>			
F	nchar(10)	<input checked="" type="checkbox"/>			
A_	int	<input type="checkbox"/>			
		<input type="checkbox"/>			

The attributes in FK have the same domain(s) as the primary key attributes PK.

Set foreign key (cont'd.)

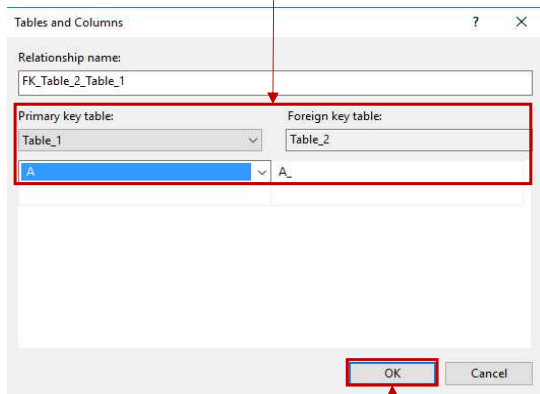
The screenshot illustrates the steps to create a foreign key relationship in SQL Server Enterprise Manager:

- Click right-side button [Keys] of table that you want to set foreign key.** The 'Keys' button for 'dbo.Table_2' is highlighted.
- Click.** The 'New Foreign Key...' option in the context menu is highlighted.
- Click.** The 'Tables And Columns Specification' tab in the 'Foreign Key Relationships' dialog is highlighted.

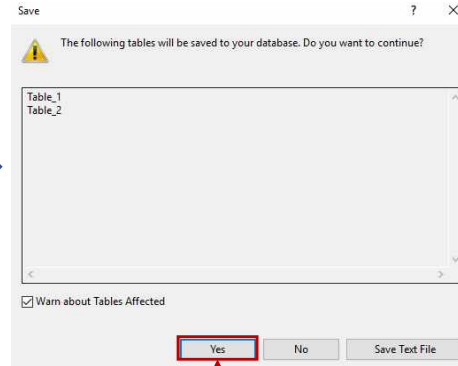
The 'Foreign Key Relationships' dialog shows the relationship 'FK_Table_2_Table_2*' being created between 'Table_2' and 'Table_2'. The 'Tables And Columns Specification' tab is active, showing the foreign key base table as 'Table_2', foreign key columns as 'D', primary/unique key base as 'Table_2', and primary/unique key columns as 'D'. The 'Identity' tab shows the relationship name as 'FK_Table_2_Table_2'.

Set foreign key (cont'd.)

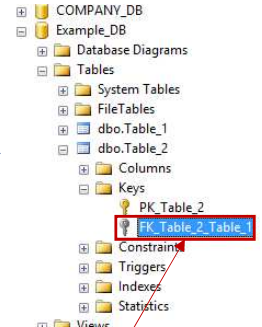
4 Set [Primary key table] and [Foreign key table].



5 Click.

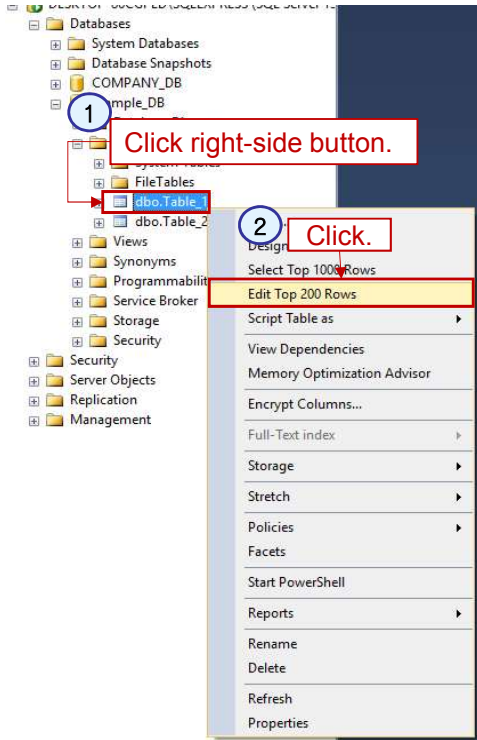


6 Click.



7 You can see the foreign key.

Insert instances



3 Type to insert instance.

	A	B	C
	1	AA	2017-03-01
▶▶	NULL	NULL	NULL

Insert instances (cont'd.)

- When attribute 'A_' of 'Table_2' refers attribute 'A' of 'Table_1':

refers

	A	B	C
1	AA		2017-03-01
2	NULL	NULL	NULL

	D	E	F	A_
1		CC	BB	1
2	NULL	NULL	NULL	NULL

	D	E	F	A_
1		CC	BB	1
2		AC	QA	5
*	NULL	NULL	NULL	NULL

Microsoft SQL Server Management Studio

No row was updated.

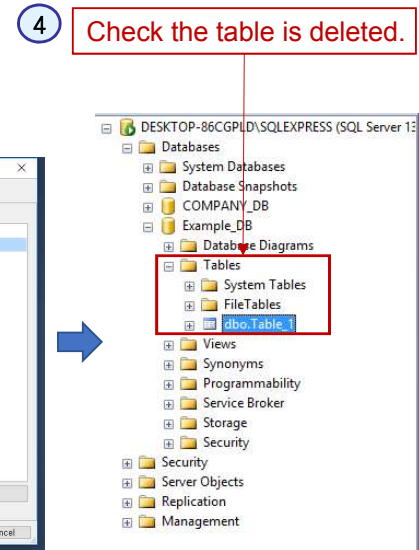
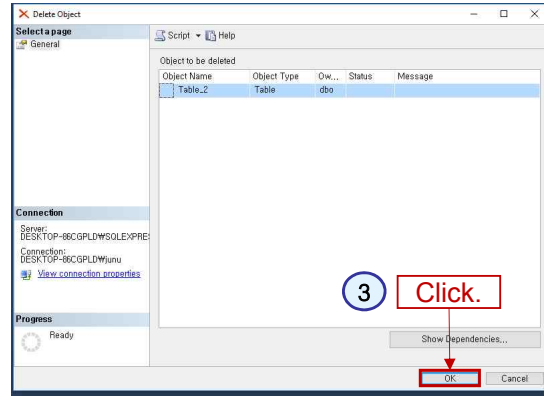
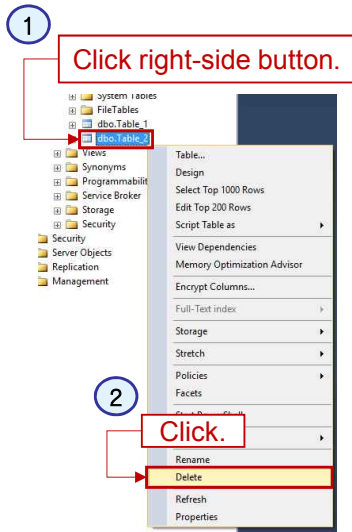
Error occurs!
Insert statement conflicts with foreign key constraint rule!

Correct the errors and retry or press ESC to cancel the change(s).

OK Help

Conflict referential integrity constraint.
Value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2) or is NULL.

Delete table



System view (catalogs)

The screenshot illustrates the steps to view all tables in a specific database using a system view. It shows the SQL Server Enterprise Manager interface with the 'System Views' folder expanded under 'Example_DB'. A red box highlights the 'Views' folder, and another red box highlights the 'System Views' folder. A blue arrow points from the 'System Views' folder to the 'INFORMATION_SCHEMA.TABLES' view. A red box highlights the 'INFORMATION_SCHEMA.TABLES' view, and a blue arrow points from it to the 'Results' pane. The 'Results' pane shows a single row of data for the 'Example_DB' database.

1 Open [Views]→[System Views].

2 Click.

3 Can see all the tables in 'Example_DB'.

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
Example_DB	dbo	Table_1	BASE TABLE

How to use SSMS (SQL server management studio) tool

- Using DDL commands.

Create database

```
CREATE DATABASE database_name
```

```
[ ON
```

```
  [ < filespec > [ ,...n ] ]
```

```
  [ , < filegroup > [ ,...n ] ]
```

```
]
```

```
[ LOG ON { < filespec > [ ,...n ] } ]
```

```
[ COLLATE collation_name ]
```

```
[ FOR LOAD | FOR ATTACH ]
```

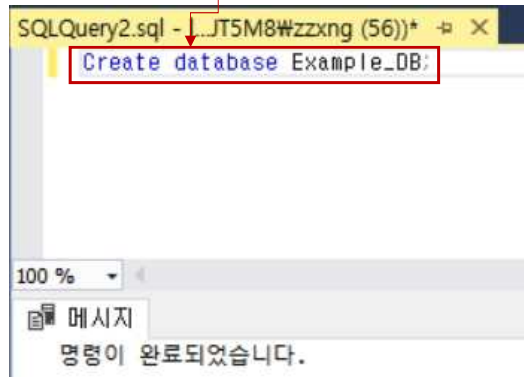
데이터베이스의 데이터 부분이 저장되는 데이터 파일 정의

데이터베이스의 로그가 저장되는 로그파일을 정의

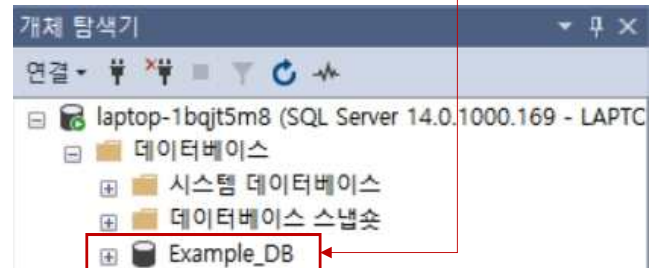
데이터 정렬 정의

이전 버전과의 호환성을 위해 지원되는 절

① Type to create database



② New database is created.



Create table

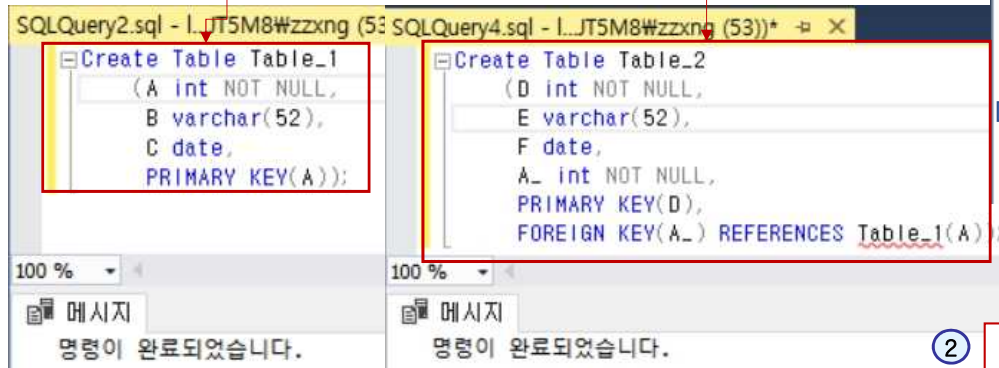
CREATE TABLE

```
[ database_name. [ owner ] . | owner. ] table_name
( ( < column_definition >
  | column_name AS computed_column_expression
  | < table_constraint > ::= [ CONSTRAINT constraint_name ] )
  [ [ { PRIMARY KEY | UNIQUE } [ ,...n ]
    ]
)
[ ON { filegroup | DEFAULT } ]
[ TEXTIMAGE_ON { filegroup | DEFAULT } ]
```

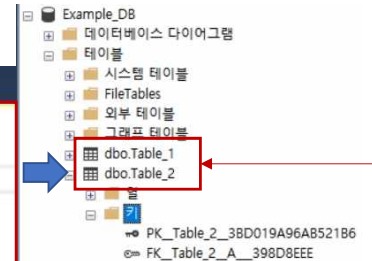
PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY 또는
CHECK 제약 조건 정의

constraint	Description
NULL NOT NULL	Null 허용 여부
PRIMARY KEY	지정한 열에 대해 고유한 인덱스 지정(1개)
UNIQUE	지정한 열에 대해 고유한 인덱스 지정(n개)
FOREIGN KEY...REFERENCES	참조 무결성 제공
ON DELETE (CASCADE NO ACTION)	부모 테이블에서 삭제시 해당 행 삭제
ON UPDATE (CASCADE NO ACTION)	부모 테이블에서 삭제시 해당 행 업데이트
CHECK	열에 입력할 수 있는 값 제한

① Type to create table



② New table is created.



Modify table design

ALTER TABLE table

{ [ALTER COLUMN column_name

{ new_data_type [(precision [, scale])]

변경된 열의 새 데이터 형식

[COLLATE < collation_name >]

열에 null 값 허용 여부를 지정

[NULL | NOT NULL]

ROWGUID 속성을 추가/삭제

[{ADD | DROP } ROWGUIDCOL]

]

| ADD

하나 이상의 열 정의 / 테이블 제약 조건 추가 지정

{ [< column_definition >]

| column_name AS computed_column_expression

} [,...n]

| [WITH CHECK | WITH NOCHECK] ADD

{ < table_constraint > } [,...n]

| DROP

테이블에서 열 / 제약 조건 제거

{ [CONSTRAINT] constraint_name

| COLUMN column } [,...n]

| { CHECK | NOCHECK } CONSTRAINT

제약조건 설정 여부 지정

{ ALL | constraint_name [,...n] }

| { ENABLE | DISABLE } TRIGGER

트리거 설정 여부 지정

{ ALL | trigger_name [,...n] }

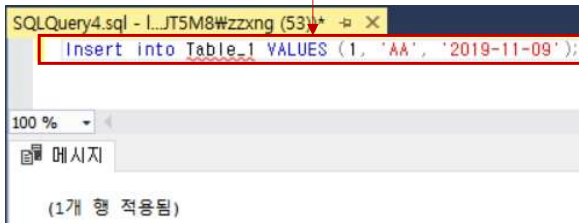
}

Insert instances

```
INSERT INTO [table_name] VALUES ('data1', 'data2', ... );  
// 테이블 생성시의 Attribute 순서와 dataType에 맞게 입력
```

```
INSERT INTO [table_name] (attr1, attr2, ... ) VALUES ('data1', ....);  
// 전체 Attribute가 아닌 특정 Attribute 에만 입력 할 경우 사용
```

① Type to insert instances



② You can see inserted instances

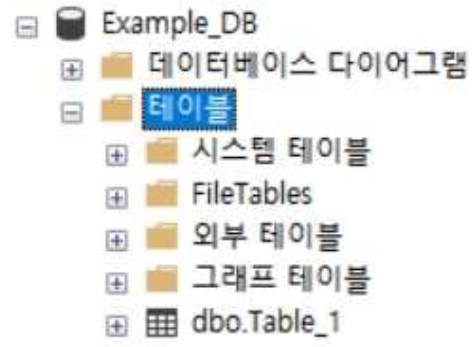
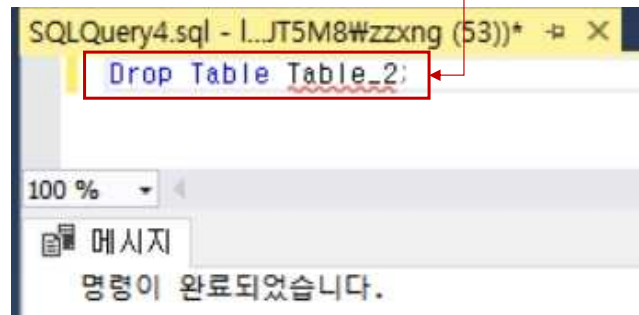
A screenshot of a table viewer showing the result of the insert operation. The table has columns A, B, and C. The first row contains the values 1, AA, and 2019-11-09. A red box highlights the entire table.

	A	B	C
1	1	AA	2019-11-09

Delete table

```
DROP TABLE table_name
```

① Type to delete table





THANK YOU



인하대학교