

# REPORT



## 2022-2 Database Systems Lab04

<b>Subject</b>	Database Systems
<b>Class</b>	01
<b>Professor</b>	Muhammad Tariq Mahmood
<b>Team member</b>	2020136129 최수연 2020136033 김지민 2020120110 이채린
<b>Submission date</b>	2022.11.26

# Table of Contents

<b>1. Introduction</b>	<b>-----3</b>
------------------------	---------------

<b>2. Database Design</b>	<b>-----4</b>
---------------------------	---------------

<b>3. Server side programming &amp; Client side programming</b>	<b>-----10</b>
---	----------------

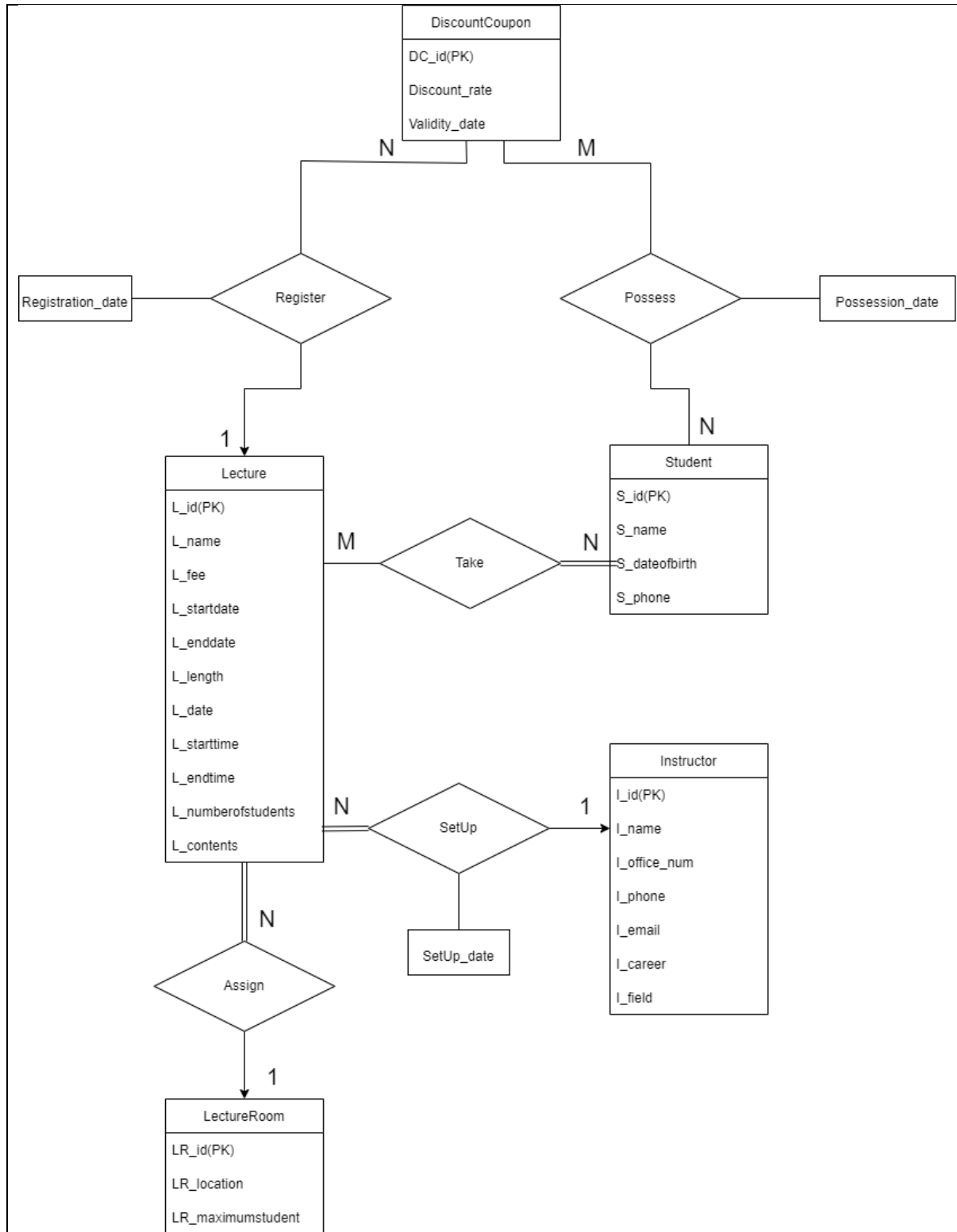
<b>4. Prepare a video presentation for the project(5-7 minutes)</b>	<b>-----52</b>
---	----------------

# **1. Introduction**

Based on what I learned in the database system lecture, I tried to create an application by building a database in a real server environment with the aim of improving the ability to use SQL server and eclipse and build DB. To implement this, we first created an ER diagram through conceptual design, and built our database through logical and physical design. In addition, stored procedure, trigger, and several user defined functions were implemented in SQL server, and Java GUI was implemented through JDBC and Javafx. The following is a report on this.

## 2. Database Design

- Develop conceptual data model (E-R diagram)



- **logical data model**

```
Lecture(L_id, L_name, L_fee, L_startdate, L_enddate, L_length, L_date, L_starttime, L_endtime,  
L_numberofstudents, L_contents, I_id(FK), SetUp_date, LR_id(FK))  
LectureRoom(LR_id, LR_location, LR_maximumstudent)  
Instructor(I_id, I_name, I_office_num, I_phone, I_email, I_career, I_field)  
Student(S_id, S_name, S_dateofbirth, S_phone)  
DiscountCoupon(DC_id, Discount_rate, Validity_date, L_id(FK), Registration_date)  
Take(S_id(FK), L_id(FK))  
Possess(DC_id(FK), S_id(FK), Possession_date)
```

- **physical design**

```
USE LectureDB;  
  
go  
DROP TABLE IF EXISTS Possess  
go  
  
DROP TABLE IF EXISTS Take  
go  
  
DROP TABLE IF EXISTS DiscountCoupon  
go  
  
DROP TABLE IF EXISTS Student  
go  
  
DROP TABLE IF EXISTS Lecture  
go  
  
DROP TABLE IF EXISTS Instructor  
go  
  
DROP TABLE IF EXISTS LectureRoom  
go  
  
CREATE TABLE LectureRoom (
```

```

LR_id VARCHAR(10) NOT NULL,
LR_location VARCHAR(50),
LR_maximumstudent INT DEFAULT 0,
CONSTRAINT pk_LR_id PRIMARY KEY (LR_id)
);

CREATE TABLE Instructor (
I_id VARCHAR(10) NOT NULL,
I_name VARCHAR(30) NOT NULL,
I_office_num VARCHAR(10) NOT NULL,
I_phone VARCHAR(16) NOT NULL,
I_email VARCHAR(40) NOT NULL,
I_career VARCHAR(2000) NOT NULL,
I_field VARCHAR(50) NOT NULL,
CONSTRAINT pk_I_id PRIMARY KEY(I_id),
CONSTRAINT unique_I_OPE UNIQUE(I_office_num, I_phone, I_email)
);

CREATE TABLE Lecture (
L_id VARCHAR(10) NOT NULL,
L_name VARCHAR(30) NOT NULL,
L_fee INT DEFAULT 0,
L_startdate DATE NOT NULL,
L_enddate DATE NOT NULL,
L_length INT,
L_date VARCHAR(10) NOT NULL,
L_starttime TIME NOT NULL,
L_endtime TIME NOT NULL,
L_numberofstudents INT DEFAULT 0,
L_contents VARCHAR(2000),
I_id VARCHAR(10) NOT NULL,
LR_id VARCHAR(10) NOT NULL,
SetUp_date DATE NOT NULL,
CONSTRAINT pk_L_id PRIMARY KEY(L_id),
CONSTRAINT fk_I_idL FOREIGN KEY(I_id) REFERENCES Instructor(I_id) ON DELETE CASCADE,
CONSTRAINT fk_LR_idL FOREIGN KEY(LR_id) REFERENCES LectureRoom(LR_id) ON DELETE CASCADE,
CONSTRAINT check_L_startdate CHECK(L_startdate <= L_enddate),

```

```

CONSTRAINT unique_L_name UNIQUE(L_name)
);

CREATE TABLE Student (
S_id VARCHAR(10) NOT NULL,
S_name VARCHAR(30) NOT NULL,
S_dateofbirth DATE,
S_phone VARCHAR(16) NOT NULL,
CONSTRAINT pk_S_id PRIMARY KEY(S_id),
CONSTRAINT unique_S_phone UNIQUE(S_phone)
);

CREATE TABLE DiscountCoupon (
DC_id INT NOT NULL,
Discount_rate INT NOT NULL,
Validity_date date NOT NULL,
L_id VARCHAR(10) NOT NULL,
Registration_date DATE NOT NULL,
CONSTRAINT pk_DC_id PRIMARY KEY(DC_id),
CONSTRAINT fk_L_idD FOREIGN KEY(L_id) REFERENCES Lecture(L_id) ON DELETE CASCADE,
CONSTRAINT check_Discount_rate CHECK(Discount_rate > 0 AND Discount_rate <= 100)
);

CREATE TABLE Take (
S_id VARCHAR(10) NOT NULL,
L_id VARCHAR(10) NOT NULL,
take_date DATE NOT NULL,
CONSTRAINT pk_SL_id PRIMARY KEY(S_id, L_id),
CONSTRAINT fk_S_idT FOREIGN KEY(S_id) REFERENCES Student(S_id) ON DELETE CASCADE,
CONSTRAINT fk_L_idT FOREIGN KEY(L_id) REFERENCES Lecture(L_id) ON DELETE CASCADE
);

CREATE TABLE Possess (
DC_id INT NOT NULL,
S_id VARCHAR(10) NOT NULL,
Possession_date date NOT NULL,
CONSTRAINT pk_DCS_id PRIMARY KEY(DC_id, S_id),
CONSTRAINT fk_DC_idP FOREIGN KEY(DC_id) REFERENCES DiscountCoupon(DC_id) ON DELETE

```

```
CASCADE,  
CONSTRAINT fk_S_idP FOREIGN KEY(S_id) REFERENCES Student(S_id) ON DELETE CASCADE  
);
```

```
use LectureDB  
go
```

```
INSERT INTO LectureRoom VALUES ('LR01', '201', '20');  
INSERT INTO LectureRoom VALUES ('LR02', '302', '10');  
INSERT INTO LectureRoom VALUES ('LR03', '409', '20');  
INSERT INTO LectureRoom VALUES ('LR04', '802', '10');  
INSERT INTO LectureRoom VALUES ('LR05', '404', '10');
```

```
Select * from LectureRoom
```

```
INSERT INTO Instructor VALUES ('I01', 'Jeemin', '101', '010-1234-5678',  
'119cloud@koreatech.ac.kr', 'professor 3 years', 'Korean literature');  
INSERT INTO Instructor VALUES ('I02', 'Sooyeon', '102', '010-1004-8282',  
'sooddong@koreatech.ac.kr', 'Doctoral degree', 'Mathematics');  
INSERT INTO Instructor VALUES ('I03', 'Charin', '104', '010-5858-1234', 'chacha@koreatech.ac.kr',  
'High school teacher 2 years', 'Chemical engineering');  
INSERT INTO Instructor VALUES ('I04', 'Miri', '105', '010-4242-0000', 'abc@naver.com', '3 years of  
teaching experience', 'Biotechnology');  
INSERT INTO Instructor VALUES ('I05', 'May', '106', '010-0101-0101', 'haha@gmail.com', 'Master  
degree', 'English literature');
```

```
Select * from Instructor
```

```
INSERT INTO Lecture VALUES ('L01', 'Math', '100000', '2022-10-01', '2022-12-16', '3', 'MON',  
'09:00:00', '12:00:00', '10', 'Differential and integral calculus', 'I01', 'LR01', '2022-09-30');  
INSERT INTO Lecture VALUES ('L02', 'Korean', '80000', '2022-08-26', '2022-10-01', '2', 'TUE',  
'13:30:00', '15:30:00', '8', 'Grammar', 'I02', 'LR02', '2022-08-01');  
INSERT INTO Lecture VALUES ('L03', 'Biology', '90000', '2022-10-01', '2022-12-15', '3', 'WED',  
'10:00:00', '13:00:00', '10', 'Anatomy', 'I03', 'LR03', '2022-09-30');  
INSERT INTO Lecture VALUES ('L04', 'Chemistry', '90000', '2022-10-10', '2022-12-15', '2', 'THU',  
'12:00:00', '14:00:00', '9', 'Chemical formula', 'I04', 'LR04', '2022-09-30');  
INSERT INTO Lecture VALUES ('L05', 'English', '100000', '2022-8-10', '2022-10-01', '1', 'FRI',  
'13:00:00', '14:00:00', '7', 'Conversational English', 'I05', 'LR05', '2022-08-01');
```



Select \* from Lecture

INSERT INTO Student VALUES ('2020126602', 'Yuri', '2001-01-01', '010-4241-1249');

INSERT INTO Student VALUES ('2021341234', 'Youheon', '2002-04-05', '010-1357-1113');

INSERT INTO Student VALUES ('2017332122', 'Rin', '1998-08-12', '010-2468-1012');

INSERT INTO Student VALUES ('2019126223', 'Siyeon', '2000-02-27', '010-1357-9111');

INSERT INTO Student VALUES ('2020341331', 'Rose', '2001-10-01', '010-1119-1119');

Select \* from Student

INSERT INTO DiscountCoupon VALUES (10674, 30, '2022-12-31', 'L01', '2022-09-14');

INSERT INTO DiscountCoupon VALUES (86723, 15, '2022-12-31', 'L02', '2022-09-20');

INSERT INTO DiscountCoupon VALUES (65420, 20, '2022-12-31', 'L03', '2022-09-25');

INSERT INTO DiscountCoupon VALUES (97463, 25, '2022-12-31', 'L04', '2022-09-28');

INSERT INTO DiscountCoupon VALUES (36753, 40, '2022-12-31', 'L05', '2022-09-30');

Select \* from DiscountCoupon

INSERT INTO Take VALUES ('2020126602', 'L01', '2022-11-10');

INSERT INTO Take VALUES ('2021341234', 'L02', '2022-09-04');

INSERT INTO Take VALUES ('2017332122', 'L03', '2022-10-27');

INSERT INTO Take VALUES ('2019126223', 'L04', '2022-11-23');

INSERT INTO Take VALUES ('2020341331', 'L05', '2022-09-15');

Select \* from Take

INSERT INTO Possess VALUES (10674, '2020126602', '2022-07-28');

INSERT INTO Possess VALUES (86723, '2021341234', '2022-08-25');

INSERT INTO Possess VALUES (65420, '2017332122', '2022-07-15');

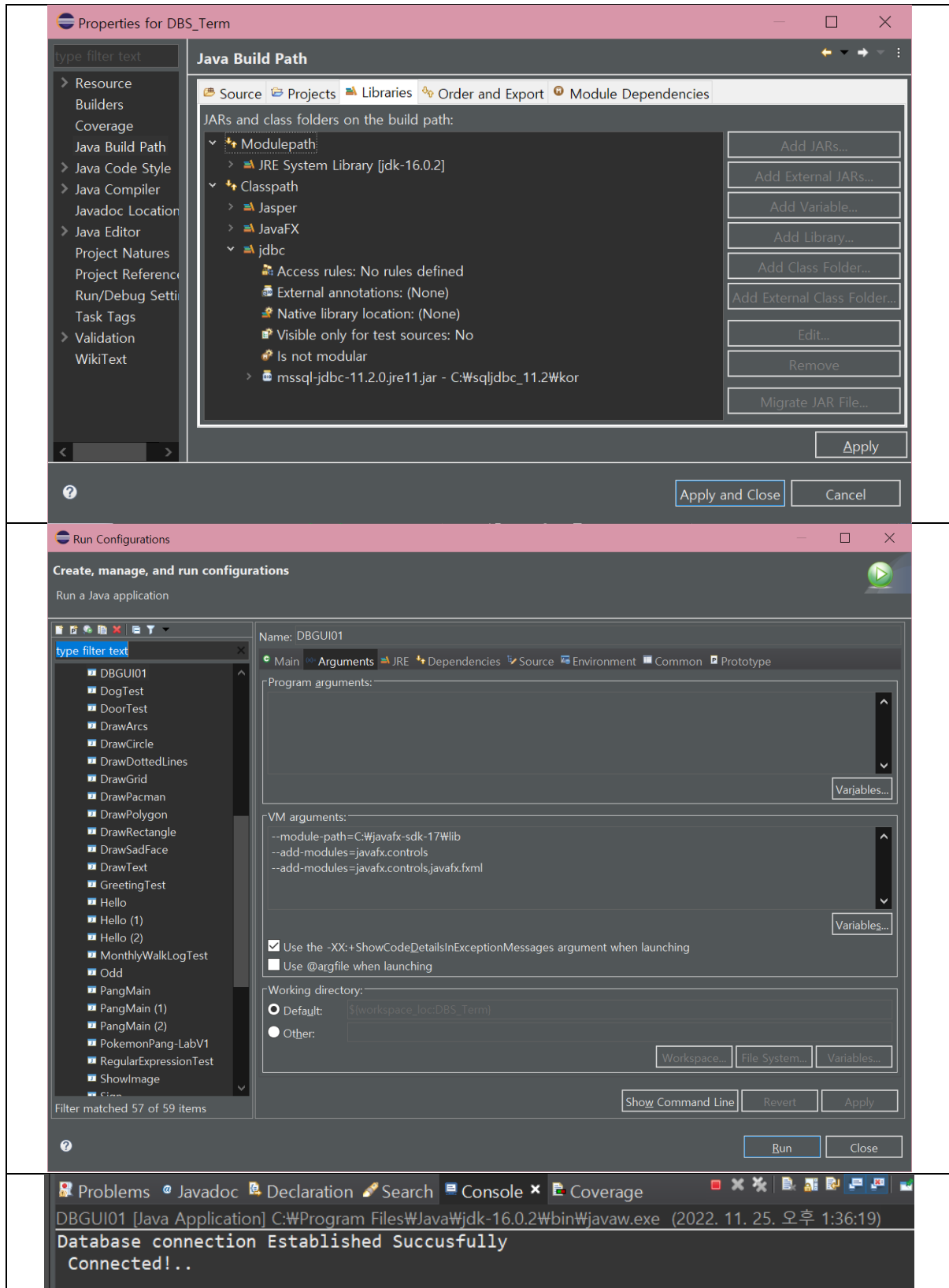
INSERT INTO Possess VALUES (97463, '2019126223', '2022-08-03');

INSERT INTO Possess VALUES (36753, '2020341331', '2022-09-16');

Select \* from Possess

### 3. Server side programming & Client side programming

1. Establish a connection between your data base and Java Application using JDBC driver



Lecture

clear

save

update

delete

print

search

▼ Lecture

▼ Tables

DiscountCoupon

Instructor

Lecture

LectureRoom

Possess

Student

Take

► Reports

Exit

About

DC_id	Discount_rate	Validity_date	L_id	Registration_date
10674	30	2022-12-31	L01	2022-09-14
36753	40	2022-12-31	L05	2022-09-30
65420	20	2022-12-31	L03	2022-09-25
86723	15	2022-12-31	L02	2022-09-20
97463	25	2022-12-31	L04	2022-09-28

DC\_id

10674

Discount\_rate

30

Validity\_date

2022-12-31

L\_id

L01

Registration\_date

2022-09-14

Result a connection between our data base and Java Application using JDBC driver

2. Create an appropriate GUI using Java language. Your GUI should facilitate user to perform basic operations on database. User should be able to

- display information about your database (tables, functions, stored procedures etc)

The screenshot shows a Java GUI window titled "Database Meta Data". The main heading is "About my Database". On the left, there is a tree view under the "Lecture" folder, listing various database objects: Tables (LogTable, LectureRoom, Instructor, Lecture, Student, DiscountCoupon, Take, Possess), Scalar\_Functions (CalculateArea, GetAge, getSum, rectangleArea, reverseCustName), Table\_Functions, Stored\_Procedures, and Triggers. The right side of the GUI displays a list of database metadata in a key-value format, each with a text input field for the value.

Property	Value
Database name	LectureDB
Number of Tables	8
Number of Scalar Valued Functions	5
Number of Table-Valued Functions	10
Number of Stored procedures	21
Number of Triggers	21
DBMS name:	Microsoft SQL Server
DBMS Version:	15.00.2000
JDBC Driver name:	Microsoft JDBC Driver 11.2 for SQL Server
JDBC Driver Version:	11.2.0.0

<ul style="list-style-type: none"> <li>Functions <ul style="list-style-type: none"> <li>Table-valued Functions <ul style="list-style-type: none"> <li>dbo.getColumnList</li> <li>dbo.getFunctionList</li> <li>dbo.getParameters</li> <li>dbo.getPKFK</li> <li>dbo.getProcedure</li> <li>dbo.getRCtable</li> <li>dbo.getTableList</li> <li>dbo.getTriggers1</li> <li>dbo.getTriggers2</li> <li>dbo.getUDtrigger</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Stored Procedures <ul style="list-style-type: none"> <li>System Stored Procedures <ul style="list-style-type: none"> <li>dbo.sp_addDiscountCoupon</li> <li>dbo.sp_addInstructor</li> <li>dbo.sp_addLecture</li> <li>dbo.sp_addLectureRoom</li> <li>dbo.sp_addPossess</li> <li>dbo.sp_addStudent</li> <li>dbo.sp_addTake</li> <li>dbo.sp_deleteDiscountCoupon</li> <li>dbo.sp_deleteInstructor</li> <li>dbo.sp_deleteLecture</li> <li>dbo.sp_deleteLectureRoom</li> <li>dbo.sp_deletePossess</li> <li>dbo.sp_deleteStudent</li> <li>dbo.sp_deleteTake</li> <li>dbo.sp_updateDiscountCoupon</li> <li>dbo.sp_updateInstructor</li> <li>dbo.sp_updateLecture</li> <li>dbo.sp_updateLectureRoom</li> <li>dbo.sp_updatePossess</li> <li>dbo.sp_updateStudent</li> <li>dbo.sp_updateTake</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>LectureDB <ul style="list-style-type: none"> <li>Database Diagrams</li> <li>Tables <ul style="list-style-type: none"> <li>System Tables</li> <li>FileTables</li> <li>External Tables</li> <li>Graph Tables</li> <li>dbo.DiscountCoupon</li> <li>dbo.Instructor</li> <li>dbo.Lecture</li> <li>dbo.LectureRoom</li> <li>dbo.LogTable</li> <li>dbo.Possess</li> <li>dbo.Student</li> <li>dbo.Take</li> </ul> </li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>dbo.LectureRoom <ul style="list-style-type: none"> <li>Columns</li> <li>Keys</li> <li>Constraints</li> <li>Triggers <ul style="list-style-type: none"> <li>myTrigger_LR</li> <li>tr_deleteLectureRoom</li> <li>tr_updateLectureRoom</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>dbo.DiscountCoupon <ul style="list-style-type: none"> <li>Columns</li> <li>Keys</li> <li>Constraints</li> <li>Triggers <ul style="list-style-type: none"> <li>myTrigger_D</li> <li>tr_deleteDiscountCoupon</li> <li>tr_updateDiscountCoupon</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>dbo.Instructor <ul style="list-style-type: none"> <li>Columns</li> <li>Keys</li> <li>Constraints</li> <li>Triggers <ul style="list-style-type: none"> <li>myTrigger_I</li> <li>tr_deleteInstructor</li> <li>tr_updateInstructor</li> </ul> </li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>dbo.Lecture <ul style="list-style-type: none"> <li>Columns</li> <li>Keys</li> <li>Constraints</li> <li>Triggers <ul style="list-style-type: none"> <li>myTrigger_L</li> <li>tr_deleteLecture</li> <li>tr_updateLecture</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>dbo.Possess <ul style="list-style-type: none"> <li>Columns</li> <li>Keys</li> <li>Constraints</li> <li>Triggers <ul style="list-style-type: none"> <li>myTrigger_P</li> <li>tr_deletePossess</li> <li>tr_updatePossess</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>dbo.Student <ul style="list-style-type: none"> <li>Columns</li> <li>Keys</li> <li>Constraints</li> <li>Triggers <ul style="list-style-type: none"> <li>myTrigger_S</li> <li>tr_deleteStudent</li> <li>tr_updateStudent</li> </ul> </li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>dbo.Take <ul style="list-style-type: none"> <li>Columns</li> <li>Keys</li> <li>Constraints</li> <li>Triggers <ul style="list-style-type: none"> <li>myTrigger_T</li> <li>tr_deleteTake</li> <li>tr_updateTake</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Functions <ul style="list-style-type: none"> <li>Table-valued Functions</li> <li>Scalar-valued Functions <ul style="list-style-type: none"> <li>dbo.CalculateArea</li> <li>dbo.GetAge</li> <li>dbo.getSum</li> <li>dbo.rectangleArea</li> <li>dbo.reverseCustName</li> </ul> </li> </ul> </li> </ul>	

### [Scalar-valued Functions]

```
use LectureDB
go

IF OBJECT_ID('getSum') IS NOT NULL DROP FUNCTION dbo.getSum;
GO

create function getSum(@a int=10, @b int=10, @c int=10)
returns int
as
begin
declare @d as int
set @d=@a+@b+@c
return @d
end;
go

select dbo.getSum(13, 5, 17)
-----

use LectureDB
go

IF OBJECT_ID('dbo.CalculateArea') IS NOT NULL DROP FUNCTION dbo.CalculateArea;
GO

create function dbo.CalculateArea(@radius as float)
returns float
as
begin
return PI()* power(@radius,2);
end;
go

declare @p as float
set @p=5.8
select dbo.CalculateArea(@p)
-----

use LectureDB
```

```

go

IF OBJECT_ID('dbo.GetAge') IS NOT NULL DROP FUNCTION dbo.GetAge;
GO

CREATE FUNCTION dbo.GetAge(@birthdate AS DATE)
RETURNS INT
AS
BEGIN
RETURN DATEDIFF(year, @birthdate, sysdatetime());
END;
GO

-----

use LectureDB
go

IF OBJECT_ID('dbo.rectangleArea') IS NOT NULL DROP FUNCTION dbo.rectangleArea;
GO

CREATE FUNCTION dbo.rectangleArea(@Width int, @Height int)
RETURNS int
AS
BEGIN
RETURN ( @Width * @Height )
END
GO

select dbo.rectangleArea(4,8)

-----

use LectureDB
go

IF OBJECT_ID('dbo.reverseCustName') IS NOT NULL DROP FUNCTION dbo.reverseCustName;
GO

CREATE FUNCTION dbo.reverseCustName(@string varchar(100))
RETURNS varchar(100)
AS

```

```
BEGIN  
DECLARE @custName varchar(100)  
set @custName = REVERSE(@string)  
RETURN @custName  
END  
go
```



### [Table-valued Functions]

```
use LectureDB
go

IF OBJECT_ID ('dbo.getTableList') is not null drop function dbo.getTableList;
GO
Create function dbo.getTableList()
Returns table
As
Return
Select * From sys.objects
Where type = 'U'
GO
Select * from dbo.getTableList()
-----

use LectureDB
go

IF OBJECT_ID ('dbo.getPKFK') is not null drop function dbo.getPKFK;
GO
Create function dbo.getPKFK()
Returns table
As
Return
select i.table_name, i.column_name, o.type
from sys.objects as o, information_schema.key_column_usage as i
where i.constraint_name = o.name and (o.type = 'PK' or o.type = 'F')
GO
Select * from dbo.getPKFK()
-----

use LectureDB
go

IF OBJECT_ID ('dbo.getUDtrigger') is not null drop function dbo.getUDtrigger;
GO
Create function dbo.getUDtrigger()
Returns table
As
```

```

Return
Select type, name
From sys.objects
Where type = 'tr'
GO
Select * from dbo.getUDtrigger()
-----

use LectureDB
go

IF OBJECT_ID ('dbo.getRCtable') is not null drop function dbo.getRCtable;
GO
Create function dbo.getRCtable ()
Returns table
As
Return
Select t.name, s.row_count
From sys.tables t, sys.dm_db_partition_stats s
Where t.object_id = s.object_id and t.type_desc = 'USER_TABLE'
GO
Select * from dbo.getRCtable()
-----

use LectureDB
go

IF OBJECT_ID ('dbo.getFunctionList') is not null drop function dbo.getFunctionList;
GO
Create function dbo.getFunctionList ()
Returns table
As
Return
Select name as function_name, schema_name(schema_id) as schema_name, type, type_desc
From sys.objects
Where type_desc like '%function%'
GO
Select * from dbo.getFunctionList()
-----

use LectureDB

```

```

go
IF OBJECT_ID ('dbo.getColumnList') is not null drop function dbo.getColumnList;
go

Create function dbo.getColumnList (@cname varchar(40))
Returns table
As
Return
select schema_name (schema_id) as schema_name, o.name as object_name, o.type, o.type_desc,
c.column_id, c.name as column_name, type_name(c.user_type_id) as column_type, c.max_length
from sys.objects as o, sys.columns as c
where o.object_id = c.object_id and o.object_id = OBJECT_ID(@cname)
go
select * from dbo.getColumnList('Instructor')
-----

use LectureDB
go

IF OBJECT_ID ('dbo.getParameters') is not null drop function dbo.getParameters;
go

Create function dbo.getParameters (@fname varchar(40))
Returns table
As
Return
select schema_name (schema_id) as schema_name, o.name as object_name, o.type, o.type_desc,
p.parameter_id, p.name as parameter_name, type_name(p.user_type_id) as parameter_type,
p.max_length, p.precision
from sys.objects as o, sys.parameters as p
where o.object_id = p.object_id and o.object_id = OBJECT_ID(@fname)
go
select * from dbo.getParameters('getColumnList')
-----

use LectureDB
go

IF OBJECT_ID ('dbo.getProcedure') is not null drop function dbo.getProcedure;
GO

```

```
Create function dbo.getProcedure()
Returns table
As
Return
Select * From sys.objects
Where type = 'P'
GO
Select * from dbo.getProcedure()
```

#### [Table-valued Functions getTriggers]

```
use LectureDB
go

/*시스템 트리거의 TR 확인*/
create function getTriggers1()
returns table

return
select name, type, type_desc, create_date from sys.triggers
where type = 'TR'
go

/*시스템 객체의 TR 확인*/
create function getTriggers2()
returns table

return
select name, type, type_desc, create_date from sys.objects
where type = 'TR'
go

select * from sys.triggers
select * from sys.objects

-- 실행 후 'Programmability > Functions > Table-valued Functions' 에서 트리거 확인
```

### [StoredProcedures\_delete]

```
use LectureDB
go

Create procedure sp_deleteLectureRoom
@c1 varchar(10),
@c2 varchar(50),
@c3 int
As
Begin
delete from dbo.LectureRoom
           where LR_id=@c1
End
Go

Create procedure sp_deleteInstructor
@c1 varchar(10),
@c2 varchar(30),
@c3 varchar(10),
@c4 varchar(16),
@c5 varchar(40),
@c6 varchar(2000),
@c7 varchar(50)
As
Begin
delete from dbo.Instructor
           where I_id=@c1
End
Go

Create procedure sp_deleteLecture
@c1 varchar(10),
@c2 varchar(30),
@c3 int,
@c4 date,
@c5 date,
@c6 int,
@c7 varchar(10),
```

```

@c8 time,
@c9 time,
@c10 int,
@c11 varchar(2000),
@c12 varchar(10),
@c13 varchar(10),
@c14 date
As
Begin
delete from dbo.Lecture
            where L_id=@c1
End
Go

Create procedure sp_deleteStudent
@c1 varchar(10),
@c2 varchar(30),
@c3 date,
@c4 varchar(16)
As
Begin
delete from dbo.Student
            where S_id=@c1
End
Go

Create procedure sp_deleteDiscountCoupon
@c1 int,
@c2 int,
@c3 date,
@c4 varchar(10),
@c5 date
As
Begin
delete from dbo.DiscountCoupon
            where DC_id=@c1
End
Go

```

```
Create procedure sp_deleteTake
@c1 varchar(10),
@c2 varchar(10),
@c3 date
As
Begin
delete from dbo.Take
            where S_id=@c1 and L_id=@c2
End
Go

Create procedure sp_deletePossess
@c1 int,
@c2 varchar(10),
@c3 date
As
Begin
delete from dbo.Possess
            where DC_id=@c1 and S_id=@c2
End
Go
```

### [StoredProcedures\_insert]

```
use LectureDB
go

Create procedure sp_addLectureRoom
@c1 varchar(10),
@c2 varchar(50),
@c3 int
As
Begin
Insert into dbo.LectureRoom values(@c1, @c2, @c3)
End
Go

Create procedure sp_addInstructor
@c1 varchar(10),
@c2 varchar(30),
@c3 varchar(10),
@c4 varchar(16),
@c5 varchar(40),
@c6 varchar(2000),
@c7 varchar(50)
As
Begin
Insert into dbo.Instructor values(@c1, @c2, @c3, @c4, @c5, @c6, @c7)
End
Go

Create procedure sp_addLecture
@c1 varchar(10),
@c2 varchar(30),
@c3 int,
@c4 date,
@c5 date,
@c6 int,
@c7 varchar(10),
@c8 time,
@c9 time,
```



```
@c10 int,  
@c11 varchar(2000),  
@c12 varchar(10),  
@c13 varchar(10),  
@c14 date  
As  
Begin  
Insert into dbo.Lecture values(@c1, @c2, @c3, @c4, @c5, @c6, @c7, @c8, @c9, @c10, @c11,  
@c12,  
@c13, @c14)  
End  
Go  
  
Create procedure sp_addStudent  
@c1 varchar(10),  
@c2 varchar(30),  
@c3 date,  
@c4 varchar(16)  
As  
Begin  
Insert into dbo.Student values(@c1, @c2, @c3, @c4)  
End  
Go  
  
Create procedure sp_addDiscountCoupon  
@c1 int,  
@c2 int,  
@c3 date,  
@c4 varchar(10),  
@c5 date  
As  
Begin  
Insert into dbo.DiscountCoupon values(@c1, @c2, @c3, @c4, @c5)  
End  
Go  
  
Create procedure sp_addTake  
@c1 varchar(10),
```

```
@c2 varchar(10),  
@c3 date  
As  
Begin  
Insert into dbo.Take values(@c1, @c2, @c3)  
End  
Go  
  
Create procedure sp_addPossess  
@c1 int,  
@c2 varchar(10),  
@c3 date  
As  
Begin  
Insert into dbo.Possess values(@c1, @c2, @c3)  
End  
Go
```

### [StoredProcedures\_update]

```
use LectureDB
go

Create procedure sp_updateLectureRoom
@c1 varchar(10),
@c2 varchar(50),
@c3 int
As
Begin
Update dbo.LectureRoom
        set LR_location=@c2, LR_maximumstudent=@c3
        where LR_id=@c1
End
Go

Create procedure sp_updateInstructor
@c1 varchar(10),
@c2 varchar(30),
@c3 varchar(10),
@c4 varchar(16),
@c5 varchar(40),
@c6 varchar(2000),
@c7 varchar(50)
As
Begin
Update dbo.Instructor
        set I_name=@c2, I_office_num=@c3, I_phone=@c4, I_email=@c5,
I_career=@c6, I_field=@c7
        where I_id=@c1
End
Go

Create procedure sp_updateLecture
@c1 varchar(10),
@c2 varchar(30),
@c3 int,
@c4 date,
```

```

@c5 date,
@c6 int,
@c7 varchar(10),
@c8 time,
@c9 time,
@c10 int,
@c11 varchar(2000),
@c12 varchar(10),
@c13 varchar(10),
@c14 date
As
Begin
Update dbo.Lecture
        set    L_name=@c2,    L_fee=@c3,    L_startdate=@c4,    L_enddate=@c5,
L_length=@c6, L_date=@c7, L_starttime=@c8, L_endtime=@c9, L_numberofstudents=@c10,
L_contents=@c11, L_id=@c12, LR_id=@c13, SetUp_date=@c14
        where L_id=@c1
End
Go

Create procedure sp_updateStudent
@c1 varchar(10),
@c2 varchar(30),
@c3 date,
@c4 varchar(16)
As
Begin
Update dbo.Student
        set S_name=@c2, S_dateofbirth=@c3, S_phone=@c4
        where S_id=@c1
End
Go

Create procedure sp_updateDiscountCoupon
@c1 int,
@c2 int,
@c3 date,
@c4 varchar(10),

```

```

@c5 date
As
Begin
Update dbo.DiscountCoupon
        set Discount_rate=@c2, Validity_date=@c3, L_id=@c4, Registration_date=@c5
        where DC_id=@c1

End
Go

Create procedure sp_updateTake
@c1 varchar(10),
@c2 varchar(10),
@c3 date
As
Begin
Update dbo.Take
        set take_date=@c3
        where S_id=@c1 and L_id=@c2

End
Go

Create procedure sp_updatePossess
@c1 int,
@c2 varchar(10),
@c3 date
As
Begin
Update dbo.Possess
        set Possession_date=@c3
        where DC_id=@c1 and S_id=@c2

End
Go

```

### [myTrigger(DML\_Trigger)]

```
use LectureDB
go

/*DML Trigger*/
create trigger myTrigger_D
on DiscountCoupon
after insert, delete, update
as
begin
print 'there is a change in DiscountCoupon table '
end
go

create trigger myTrigger_I
on Instructor
after insert, delete, update
as
begin
print 'there is a change in Instructor table '
end
go

create trigger myTrigger_L
on Lecture
after insert, delete, update
as
begin
print 'there is a change in Lecture table '
end
go

create trigger myTrigger_LR
on LectureRoom
after insert, delete, update
as
begin
print 'there is a change in LectureRoom table '
```

```

end
go

create trigger myTrigger_P
on Possess
after insert, delete, update
as
begin
print 'there is a change in Possess table '
end
go

create trigger myTrigger_S
on Student
after insert, delete, update
as
begin
print 'there is a change in Student table '
end
go

create trigger myTrigger_T
on Take
after insert, delete, update
as
begin
print 'there is a change in Take table '
end
go

-----

/*쿼리문 실행 및 결과 확인*/
insert into LectureRoom values('LR06', '410', '30')
update LectureRoom set LR_maximumstudent=LR_maximumstudent-10 where LR_id='LR06'
delete from LectureRoom where LR_id='LR06'

select * from LectureRoom
select * from sys.trigger_events

```

```
select * from sys.triggers
```

```
SELECT TE.*  
FROM sys.trigger_events AS TE  
JOIN sys.triggers AS T ON T.object_id =TE.object_id  
WHERE T.parent_class = 0 AND T.name = 'myTrigger_LR';  
go
```



### [TableTrigger(DML\_Trigger) 1 delete]

```
-- Trigger to log the deleted records /*테이블의 레코드를 삭제할 때 LogTable 에 내용 기록*/
/*
use DreamHomeDB
go

create trigger tr_deleteBranch
on dbo.Branch
for delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Branch';
set @c4 = 'delete';
select @c5 = branchNo + ', ' + street + ', ' + city + ', ' + postcode from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go
*/
-----

use LectureDB
go

create trigger tr_deleteLectureRoom
on dbo.LectureRoom
for delete
as
begin
declare
@c1 datetime,
```

```

@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.LectureRoom';
set @c4 = 'delete';
select @c5 = LR_id + ', ' + LR_location + ', ' + LR_maximumstudent from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

/*다음 쿼리문을 통해 LogTable 에 대한 트리거가 잘 실행되는지 확인*/
--insert into LectureRoom values('LR06', '410', '30')
--update LectureRoom set LR_maximumstudent=LR_maximumstudent-10 where LR_id='LR06'
--delete from LectureRoom where LR_id='LR06'
-- select * from LogTable
-----

use LectureDB
go

create trigger tr_deleteDiscountCoupon
on dbo.DiscountCoupon
for delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.DiscountCoupon';
set @c4 = 'delete';

```

```

select @c5 = DC_id + ', ' + Discount_rate + ', ' + convert(varchar, Validity_date) + ', ' + L_id + ', ' + convert(varchar, Registration_date) from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_deleteInstructor
on dbo.Instructor
for delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Instructor';
set @c4 = 'delete';
select @c5 = I_id + ', ' + I_name + ', ' + I_office_num + ', ' + I_phone + ', ' + I_email + ', ' + I_career + ', ' + I_field from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_deleteLecture
on dbo.Lecture
for delete

```

```

as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Lecture';
set @c4 = 'delete';
select @c5 = L_id + ', ' + L_name + ', ' + L_fee + ', ' + convert(varchar, L_startdate) + ', ' +
convert(varchar, L_enddate) + ', ' + L_length + ', ' + L_date + ', ' + convert(varchar, L_starttime) +
', ' + convert(varchar, L_endtime) + ', ' + L_numberofstudents + ', ' + L_contents + ', ' + L_id + ',
' + LR_id + ', ' + convert(varchar, SetUp_date) from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_deletePossess
on dbo.Possess
for delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Possess';

```

```

set @c4 = 'delete';
select @c5 = DC_id + ', ' + S_id + ', ' + convert(varchar, Possession_date) from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_deleteStudent
on dbo.Student
for delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Student';
set @c4 = 'delete';
select @c5 = S_id + ', ' + S_name + ', ' + convert(varchar, S_dateofbirth) + ', ' + S_phone from
deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_deleteTake
on dbo.Take
for delete

```

```
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Take';
set @c4 = 'delete';
select @c5 = S_id + ', ' + L_id + ', ' + convert(varchar, take_date) from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go
```

### [TableTrigger(DML\_Trigger) 2 update]

```
--Trigger to log the modified records /*테이블의 레코드를 업데이트할 때 LogTable 에 내용 기록*/
use LectureDB
go

create trigger tr_updateLectureRoom
on dbo.LectureRoom
for update
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = suser_name();
set @c3 = 'dbo.LectureRoom';
set @c4 = 'update';
select @c5 = LR_id + ', ' + LR_location + ', ' + LR_maximumstudent from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_updateDiscountCoupon
on dbo.DiscountCoupon
for update
as
begin
declare
@c1 datetime,
@c2 varchar(30),
```

```

@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.DiscountCoupon';
set @c4 = 'update';
select @c5 = DC_id + ', ' + Discount_rate + ', ' + convert(varchar, Validity_date) + ', ' + L_id + ', ' + convert(varchar, Registration_date) from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_updateInstructor
on dbo.Instructor
for update
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Instructor';
set @c4 = 'update';
select @c5 = I_id + ', ' + I_name + ', ' + I_office_num + ', ' + I_phone + ', ' + I_email + ', ' + I_career + ', ' + I_field from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

```



```

-----
use LectureDB
go

create trigger tr_updateLecture
on dbo.Lecture
for update
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Lecture';
set @c4 = 'update';
select @c5 = L_id + ', ' + L_name + ', ' + L_fee + ', ' + convert(varchar, L_startdate) + ', ' +
convert(varchar, L_enddate) + ', ' + L_length + ', ' + L_date + ', ' + convert(varchar, L_starttime) +
', ' + convert(varchar, L_endtime) + ', ' + L_numberofstudents + ', ' + L_contents + ', ' + L_id + ',
' + LR_id + ', ' + convert(varchar, SetUp_date) from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go
-----

use LectureDB
go

create trigger tr_updatePossess
on dbo.Possess
for update
as
begin
declare
@c1 datetime,

```

```

@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Possess';
set @c4 = 'update';
select @c5 = DC_id + ', ' + S_id + ', ' + convert(varchar, Possession_date) from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_updateStudent
on dbo.Student
for update
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Student';
set @c4 = 'update';
select @c5 = S_id + ', ' + S_name + ', ' + convert(varchar, S_dateofbirth) + ', ' + S_phone from
deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

```

```
-----  
use LectureDB  
go  
  
create trigger tr_updateTake  
on dbo.Take  
for update  
as  
begin  
declare  
@c1 datetime,  
@c2 varchar(30),  
@c3 varchar(50),  
@c4 varchar(30),  
@c5 varchar(max);  
set @c1 = getdate();  
set @c2 = SUSER_NAME();  
set @c3 = 'dbo.Take';  
set @c4 = 'update';  
select @c5 = S_id + ', ' + L_id + ', ' + convert(varchar, take_date) from deleted;  
  
insert into LogTable values(@c1, @c2, @c3, @c4, @c5);  
end  
go
```

### [TableTrigger(DML\_Trigger) 3 delete\_update]

```
/*테이블의 레코드를 삭제 및 업데이트할 때 LogTable 에 내용 기록*/
use LectureDB
go

create trigger tr_LectureRoom
on dbo.LectureRoom
for update, delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = suser_name();
set @c3 = 'dbo.LectureRoom';
set @c4 = 'update/delete';
select @c5 = LR_id + ', ' + LR_location + ', ' + LR_maximumstudent from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_DiscountCoupon
on dbo.DiscountCoupon
for update, delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
```

```

@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.DiscountCoupon';
set @c4 = 'update/delete';
select @c5 = DC_id + ', ' + Discount_rate + ', ' + Validity_date + ', ' + L_id + ', ' + Registration_date
from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_Instructor
on dbo.Instructor
for update, delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Instructor';
set @c4 = 'update/delete';
select @c5 = I_id + ', ' + I_name + ', ' + I_office_num + ', ' + I_phone + ', ' + I_email + ', ' +
I_career + ', ' + I_field from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

```

```

use LectureDB
go

create trigger tr_Lecture
on dbo.Lecture
for update, delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Lecture';
set @c4 = 'update/delete';
select @c5 = L_id + ', ' + L_name + ', ' + L_fee + ', ' + L_startdate + ', ' + L_enddate + ', ' +
L_length + ', ' + L_date + ', ' + L_starttime + ', ' + L_endtime + ', ' + L_numberofstudents + ', ' +
L_contents + ', ' + L_id + ', ' + LR_id + ', ' + SetUp_date from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go
-----

use LectureDB
go

create trigger tr_Possess
on dbo.Possess
for update, delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),

```

```

@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Possess';
set @c4 = 'update/delete';
select @c5 = DC_id + ', ' + S_id + ', ' + Possession_date from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

create trigger tr_Student
on dbo.Student
for update, delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Student';
set @c4 = 'update/delete';
select @c5 = S_id + ', ' + S_name + ', ' + S_dateofbirth + ', ' + S_phone from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go

-----

use LectureDB
go

```

```
create trigger tr_Take
on dbo.Take
for update, delete
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = SUSER_NAME();
set @c3 = 'dbo.Take';
set @c4 = 'update/delete';
select @c5 = S_id + ', ' + L_id + ', ' + take_date from deleted;

insert into LogTable values(@c1, @c2, @c3, @c4, @c5);
end
go
```



- navigate (see) records (data) from each table in your database

▼ Lecture

▼ Tables

DiscountCoupon

Instructor

Lecture

LectureRoom

Possess

Student

Take

► Reports

Exit

About

DC_id	Discount_rate	Validity_date	L_id	Registration_date
10674	30	2022-12-31	Lo1	2022-09-14
36753	40	2022-12-31	Lo5	2022-09-30
65430	20	2022-12-31	Lo3	2022-09-25
86723	15	2022-12-31	Lo2	2022-09-20
97463	25	2022-12-31	Lo4	2022-09-28

DC\_id: 10674  
Discount\_rate: 30  
Validity\_date: 2022-12-31  
L\_id: Lo1  
Registration\_date: 2022-09-14

▼ Lecture

▼ Tables

DiscountCoupon

Instructor

Lecture

LectureRoom

Possess

Student

Take

► Reports

Exit

About

LR_id	L_name	L_office_num	L_phone	L_email	L_career	L_field
Lo1	Jeemin	101	010-1234-5678	jeemin@koreatech.ac.kr	professor 3 years	Korean literature
Lo2	Sooyeon	102	010-1004-8882	sooyeon@koreatech.ac.kr	Doctoral degree	Mathematic
Lo3	Charin	104	010-585-1234	charin@koreatech.ac.kr	High school teacher 1 years	Chemical en
Lo4	Miri	105	010-412-0000	miri@naver.com	3 years of teaching experience	Biotechnology

L\_id: Lo1  
L\_name: Jeemin  
L\_office\_num: 101  
L\_phone: 010-1234-5678  
L\_email: jeemin@koreatech.ac.kr  
L\_career: professor 3 years  
L\_field: Korean literature

▼ Lecture

▼ Tables

DiscountCoupon

Instructor

Lecture

LectureRoom

Possess

Student

Take

► Reports

Exit

About

L\_id: Lo4  
L\_name: Chemistry  
L\_fee: 90000  
L\_startdate: 2022-10-10  
L\_enddate: 2022-12-05  
L\_length: 3  
L\_date: THU  
L\_starttime: 12:00:00.000000000  
L\_endtime: 14:00:00.000000000  
L\_numberofstudents: 9

▼ Lecture

▼ Tables

DiscountCoupon

Instructor

Lecture

LectureRoom

Possess

Student

Take

► Reports

Exit

About

LR\_id: LR04  
LR\_location: 802  
LR\_maximumstudent: 10

▼ Lecture

▼ Tables

DiscountCoupon

Instructor

Lecture

LectureRoom

Possess

Student

Take

► Reports

Exit

About

DC\_id: 97463  
S\_id: 2019126223  
Possession\_date: 2022-08-03

▼ Lecture

▼ Tables

DiscountCoupon

Instructor

Lecture

LectureRoom

Possess

Student

Take

► Reports

Exit

About

S\_id: 2017331222  
S\_name: Rin  
S\_dateofbirth: 1998-08-12  
S\_phone: 010-1468-1012

▼ Lecture

▼ Tables

DiscountCoupon

Instructor

Lecture

LectureRoom

Possess

Student

Take

► Reports

Exit

About

S\_id: 2019126223  
L\_id: Lo4  
take\_date: 2022-11-23

- run functions to insert, update and delete records for each table in your database

S_id	S_name	S_dateofbirth	S_phone
200733123	Rin	1998-08-12	000-1468-1012
200926123	Siyoon	2000-02-27	000-1337-901
200026602	Yuri	2001-01-01	000-4340-1249
200036102	Sooyeon	2001-11-02	000-1234-5678
200034130	Rose	2001-10-01	000-100-100
200341234	Yoonheon	2002-04-05	000-1337-1013

record is added.

Save button click(insert)

S_id	S_name	S_dateofbirth	S_phone
200733123	Rin	1998-08-12	000-1468-1012
200926123	Siyoon	2000-02-27	000-1337-901
200026602	Yuri	2001-01-01	000-4340-1249
200036102	Sooyeon	2001-11-02	000-1234-5678
200034130	Rose	2001-10-01	000-100-100
200341234	Yoonheon	2002-04-05	000-1337-1013

Record is updated.

Result save

S_id	S_name	S_dateofbirth	S_phone
200733123	Rin	1998-08-12	000-1468-1012
200926123	Siyoon	2000-02-27	000-1337-901
200026602	Yuri	2001-01-01	000-4340-1249
200036102	Sooyeon	2001-11-02	000-1234-5678
200034130	Rose	2001-10-01	000-100-100
200341234	Yoonheon	2002-04-05	000-1337-1013

Record is updated.

Update button click

S_id	S_name	S_dateofbirth	S_phone
200733123	Rin	1998-08-12	000-1468-1012
200926123	Siyoon	2000-02-27	000-1337-901
200026602	Yuri	2001-01-01	000-4340-1249
200036102	Sooyeon	2001-11-02	000-1234-5678
200034130	Rose	2001-10-01	000-100-100
200341234	Yoonheon	2002-04-05	000-1337-1013

Record is deleted.

Delete button click

S_id	S_name	S_dateofbirth	S_phone
200733123	Rin	1998-08-12	000-1468-1012
200926123	Siyoon	2000-02-27	000-1337-901
200026602	Yuri	2001-01-01	000-4340-1249
200036102	Sooyeon	2001-11-02	000-1234-5678
200034130	Rose	2001-10-01	000-100-100
200341234	Yoonheon	2002-04-05	000-1337-1013

Record is deleted.

Result delete

- print reports (data from different tables)

Discount Coupon Data

ID	Discount Rate	Validity Date	Lecture ID	Registration Date
10674	30	22. 12. 31. 오전 12:00	L01	22. 9. 14. 오전 12:00
36753	40	22. 12. 31. 오전 12:00	L05	22. 9. 30. 오전 12:00
65420	20	22. 12. 31. 오전 12:00	L03	22. 9. 25. 오전 12:00
86723	15	22. 12. 31. 오전 12:00	L02	22. 9. 20. 오전 12:00
97463	25	22. 12. 31. 오전 12:00	L04	22. 9. 28. 오전 12:00

Instructor Data

ID	Name	Office Num	Phone	Email	Career	Field
I01	Jeemin	101	010-1234-5678	119kdud@koreatech.ac.kr	professor 3 years	Korean literature
I02	Sooyeon	102	010-1004-8282	sooddong@koreatech.ac.kr	Doctoral degree	Mathematics
I03	Charin	104	010-8856-1234	chacha@koreatech.ac.kr	High school teacher 2 years	Chemical engineering
I04	Miri	105	010-4242-0000	abc@naver.com	3 years of teaching	Biotechnology
I05	May	106	010-0101-0101	haha@gmail.com	Master degree	English literature

Lecture Data

ID	Name	Fee	SDate	EDate	Length	Date	STime	ETime	NoS	Contents	Instructor ID	Lecture Room ID	SetUp Date
L01	Math	100000	22. 10. 1. 오전 12:00	22. 12. 16. 오전 12:00	3	MON	70. 1. 1. 오전 9:00	70. 1. 1. 오전 12:00	10	Differential and integral	I01	LR01	22. 9. 30. 오전 12:00
L02	Korean	80000	22. 8. 28. 오전 12:00	22. 10. 1. 오전 12:00	2	TUE	70. 1. 1. 오전 3:30	70. 1. 1. 오전 8:30	8	Grammar	I02	LR02	22. 8. 1. 오전 12:00
L03	Biology	90000	22. 10. 1. 오전 12:00	22. 12. 16. 오전 12:00	3	WED	70. 1. 1. 오전 9:00	70. 1. 1. 오전 12:00	10	Anatomy	I03	LR03	22. 9. 30. 오전 12:00
L04	Chemistry	90000	22. 10. 1. 오전 12:00	22. 12. 15. 오전 12:00	2	THU	70. 1. 1. 오전 12:00	70. 1. 1. 오전 2:00	9	Chemical formula	I04	LR04	22. 9. 30. 오전 12:00
L05	English	100000	22. 8. 1. 오전 12:00	22. 10. 1. 오전 12:00	1	FRI	70. 1. 1. 오전 1:00	70. 1. 1. 오전 2:00	7	Conversational English	I05	LR05	22. 8. 1. 오전 12:00

Lecture Room Data

ID	Location	Maximum Student
LR01	201	20
LR02	302	10
LR03	409	20
LR04	802	10
LR05	404	10
LR06	203	20
LR07	206	30

Possess Data

Discount Coupon ID	Student ID	Possession Date
10674	2020126602	22. 7. 28. 오전 12:00
36753	2020341331	22. 9. 16. 오전 12:00
65420	2017332122	22. 7. 15. 오전 12:00
86723	2021341234	22. 8. 25. 오전 12:00
97463	2019126223	22. 8. 3. 오전 12:00

Student Data

ID	Name	Date Of Birth	Phone
2017332122	Rin	98. 8. 12. 오전 12:00	010-2468-1012
2019126223	Slyeon	00. 2. 27. 오전 12:00	010-1357-9111
2020126602	Yuri	01. 1. 1. 오전 12:00	010-4241-1249
2020341331	Rose	01. 10. 1. 오전 12:00	010-1119-1119
2021341234	Youheon	02. 4. 5. 오전 12:00	010-1357-1113

Take Data

Student ID	Lecture ID	Take Date
2017332122	L03	22. 10. 27. 오전 12:00
2019126223	L04	22. 11. 23. 오전 12:00
2020126602	L01	22. 11. 10. 오전 12:00
2020341331	L05	22. 9. 15. 오전 12:00
2021341234	L02	22. 9. 4. 오전 12:00

#### **4. Prepare a video presentation for the project(5-7 minutes)**

- Attaching mp4 file