Database System(CSE 235)

Lab 04

CSE 235(02) Database System		
DATE	22.11.26	
NAME	2018136121	조원석
	2018136111	장영진
	2019136154	이제윤
	2018136003	강희권

INDEX

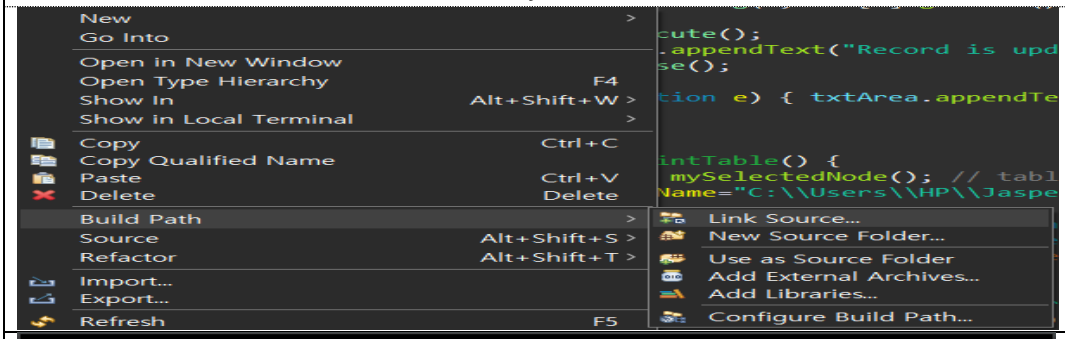
- **Tasks-1 : develop Java Application for your Database**
    - Establish a connection between your data base and Java Application using JDBC driver
    - Create an appropriate GUI using Java language
        - display information about your database (tables, functions, stored procedures etc)
        - navigate (see) records (data) from each table in your database
        - run functions to insert, update and delete records for each table in your database
        - print reports (data from different tables)
- **Tasks-2: Submit project report**
    - Introduction
    - Database Design
    - Server side Programming (Procedure, functions, triggers etc)
    - Client side Programming (Java Application)
- **Tasks-3: Prepare a video presentation for the project (5-7 minuets)**

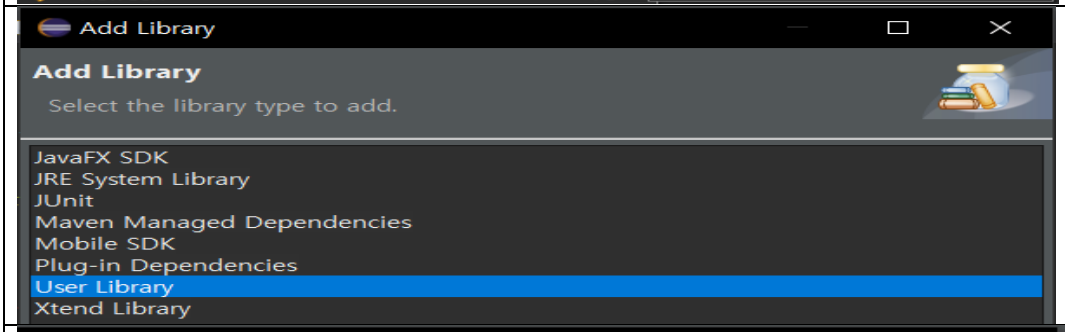
Tasks-1 : develop Java Application for your Database

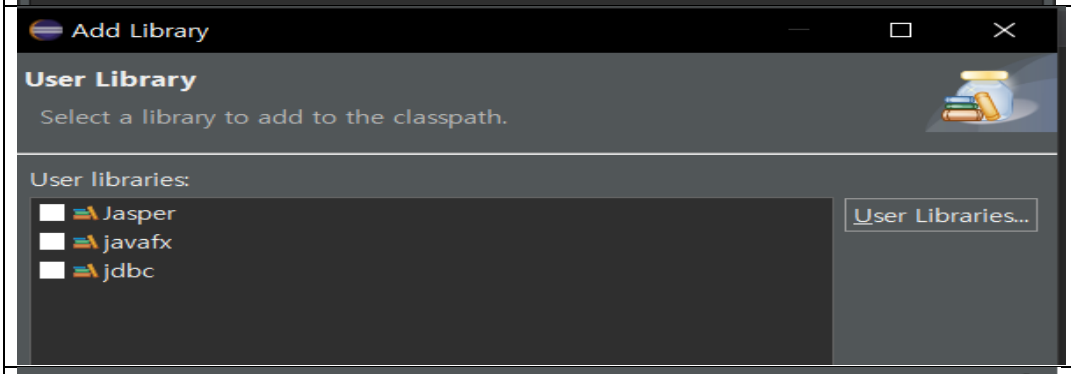
1. **Establish a connection between your database and Java Application using JDBC driver**

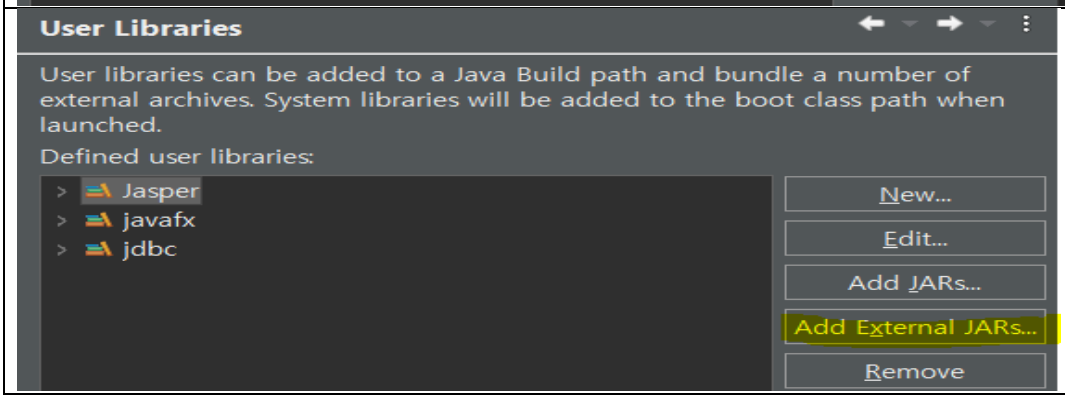
1.1 Adding User Library

Build Path -> Add Libraries -> User Library -> Add External JARs...



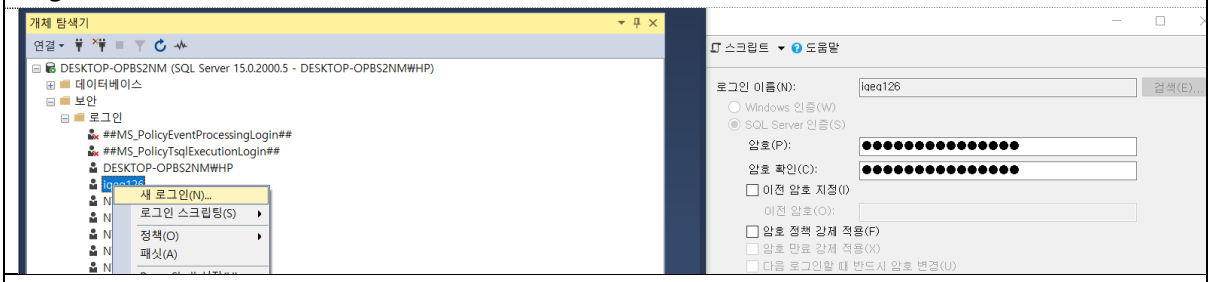






1.2 Make connection

Login set



MyConnection.java

```
public static Connection makeConnection() {
        . . .
        // load and register the Driver
        // using "encrypt=false" to avoid ssh authorization
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        String url = "jdbc:sqlserver://DESKTOP-
        OPBS2NM:1433;databaseName=RewardCrowdFunding;encrypt=false";
        // Get connection using URL through Driver Manager
        con = DriverManager.getConnection(url, "iqeq126", "1320");
        . . .
```

DBGUI.java / Metadata.java

```
public class DBGUI extends Application or public class Metadata {
. . .
private Connection con = MyConnection.makeConnection();
. . .
```

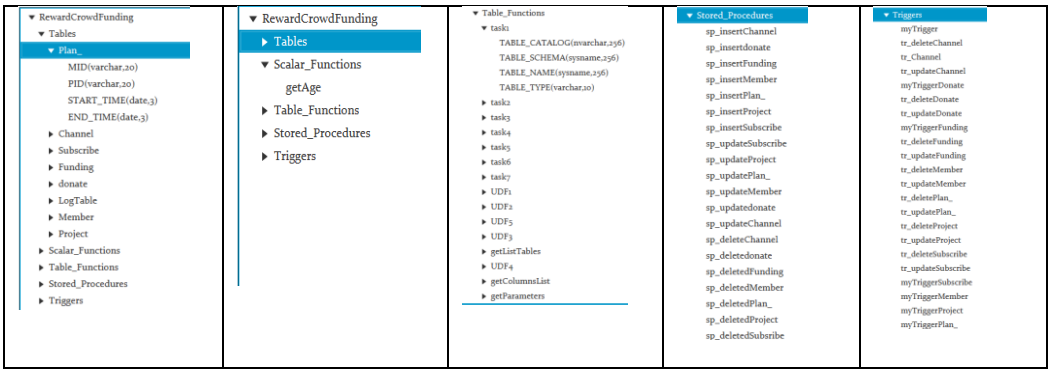
1.3 Result

2. Create an appropriate GUI using Java language

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

[Press <ctrl> & <click>]

..₩Video₩Task1.2.A.mp4



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

[Press <ctrl> & <click>]

..₩Video₩Task1.2.B.mp4

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

[Press <ctrl> & <click>], [Sample_Channel_Table]

..₩Video₩Task1.2.C.mp4

D. print reports (data from different tables)

[Press <ctrl> & <click>]

..₩Video₩Task1.2.D.mp4

Tasks-2: Submit project report

1. Introduction

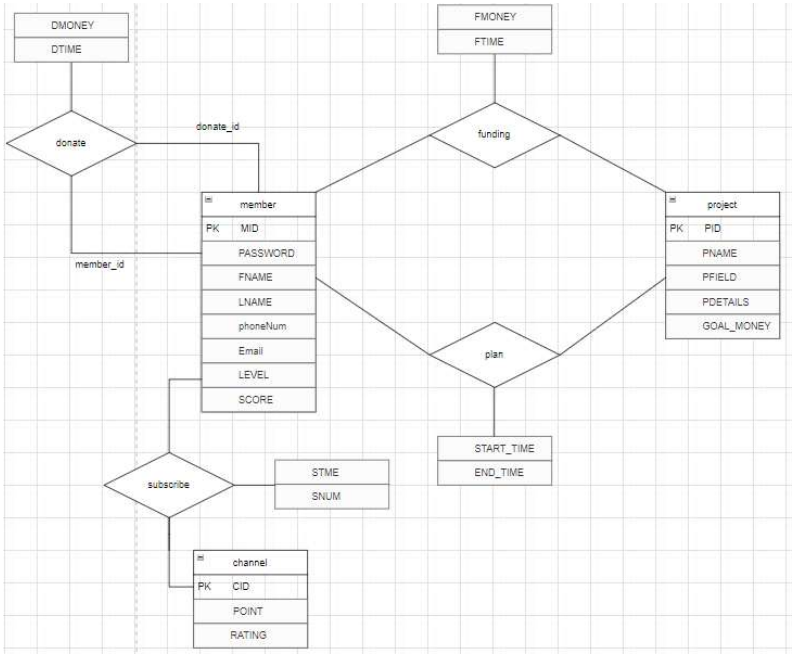
We took a class on the database. We took the following class that Database Design, Server side Programming, and Client side Programming. And we created our own projects for each assignment.

We came up with a crowdfunding database for a specific project. This database is about being donated by the people working on the project.

So, Let me introduce our own Database.

2. Database Design

Introduce our database design. Our database's ER Diagram.



List of entity and attribute

We proceeded with Logical Design with this ER diagram.

Explained entity and attribute of the database that proceeded to Logical Design.

- Project

  project that will receive crowdfunding. It has PID, PNAME, PField, PDetails, goalMoney.

  PID is primary key.

열 이름	데이터 형식	Null 허용
PID	varchar(10)	☐
PName	varchar(20)	☐
PField	varchar(20)	☐
PDetails	varchar(22)	☐
goalMoney	int	☐
		☐

- Member

  Member is the object of funding. It has member id, password, phoneNumber, eMail, and funding Score. MID is primary key.

열 이름	데이터 형식	Null 허용
MID	varchar(20)	☐
PW	varchar(30)	☐
fname	varchar(20)	☐
lname	varchar(20)	☐
phoneNum	varchar(22)	☐
eMail	varchar(30)	☑
Score	int	☐
		☐

- Funding

  Related to funding. Receives the MID, PID as a foreign key. It has attribute of a funding money, funding time. MID and PID is primary key.

열 이름	데이터 형식	Null 허용
MID	varchar(20)	☐
PID	varchar(10)	☐
FMoney	int	☐
FTIME	date	☐

- Plan_

Plan is funding with members. Receives the MID, PID as a foreign key. It has attribute of a start time, end time. MID and PID is primary key.

열 이름	데이터 형식	Null 허용
MID	varchar(20)	☐
PID	varchar(10)	☐
START_TIME	date	☐
END_TIME	date	☐
		☐

- Donate

Donate is an object related to donation. Receives the member_id as a foreign key. It has attribute of donate id, donate time. Donated_id is primary key.

열 이름	데이터 형식	Null 허용
donated_id	varchar(10)	☐
member_id	varchar(20)	☐
DTIME	date	☐
		☐

- Channel

Channel is platform with subscribe (like youtube channel). Point is the number of human subscribed to the channel. Rating is the rank based on the number of Point. Channel Id is primary key.

열 이름	데이터 형식	Null 허용
CID	varchar(10)	☐
POINT	int	☐
RATING	char(1)	☐
		☐

- Subscribe

  Subscribe is entity which channel a particular member subscribed to. CID, MID is foreign key.

열 이름	데이터 형식	Null 허용
CID	varchar(10)	☐
MID	varchar(20)	☐
STIME	date	☐
SNUM	int	☐

- LogTable

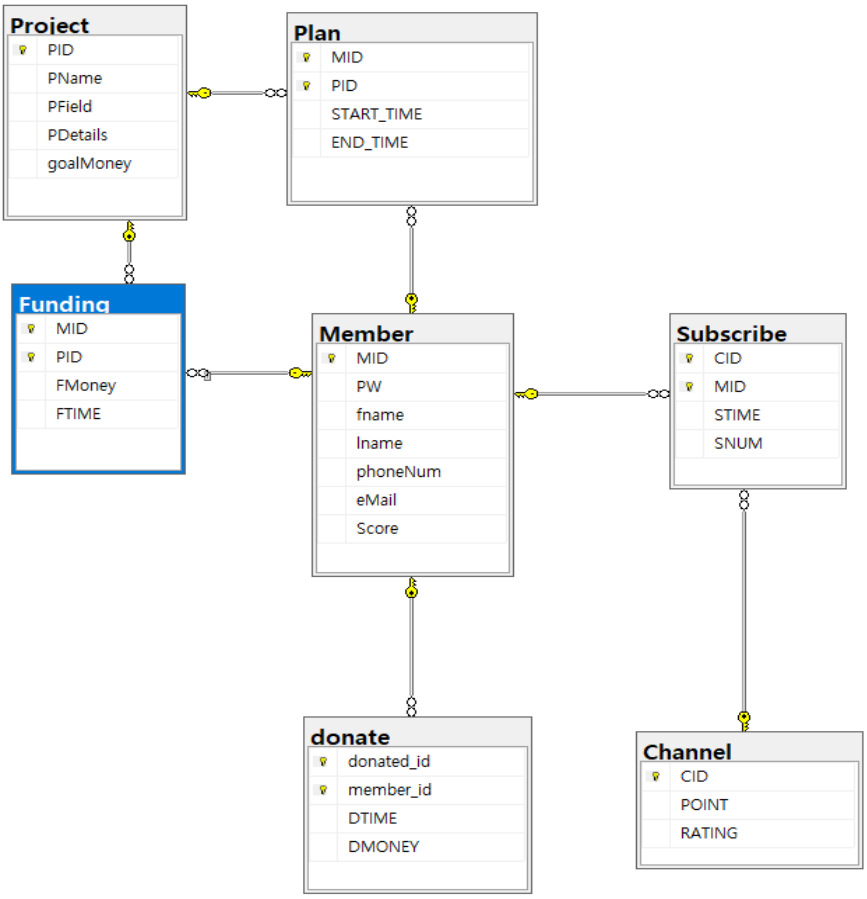
  Logdata has logdata, logUser, logTableName, logPeration, logRecord

DESKTOP-OPBS2NM...- dbo.LogTable | SQLQuery7.sql -...-OPBS2NM₩HI

열 이름	데이터 형식	Null 허용
logdate	datetime	☐
logUser	varchar(30)	☑
logTableName	varchar(50)	☑
logOperation	varchar(30)	☑
logRecord	varchar(MAX)	☑
		☐

- Full Table

  below is the overall table diagram and connection relationship



2.1 Table Create Quary

Member

```
CREATE TABLE Member(
MID VARCHAR(20) NOT NULL,   -- NOT NULL Constraint
PW VARCHAR(30) NOT NULL,    -- NOT NULL Constraint
fname VARCHAR(20) NOT NULL,    -- NOT NULL Constraint
lname VARCHAR(20) NOT NULL,         -- NOT NULL Constraint
phoneNum VARCHAR(22) NOT NULL,    -- NOT NULL Constraint
eMail VARCHAR(30) NULL,
Score INT NOT NULL ,                -- NOT NULL Constraint
PRIMARY KEY (MID), -- PRIMARY Key Constraint
-- CHECK (branchNo like 'B%') -- CHECK Constraint
);
go
```

Project

```
CREATE TABLE   Project(
PID VARCHAR(10) NOT NULL,          -- NOT NULL Constraint
PName VARCHAR(20) NOT NULL,          -- NOT NULL Constraint
PField VARCHAR(20) NOT NULL,      -- NOT NULL Constraint
PDetails VARCHAR(22) NOT NULL,          -- NOT NULL Constraint
goalMoney INT NOT NULL,              -- NOT NULL Constraint
PRIMARY KEY (PID), -- PRIMARY Key Constraint
CHECK (goalMoney >= 100000)
);
go
```

Funding

```
CREATE TABLE Funding(
MID VARCHAR(20) NOT NULL,          -- NOT NULL Constraint
PID VARCHAR(10) NOT NULL,          -- NOT NULL Constraint
FMoney INT NOT NULL,               -- NOT NULL Constraint
FTIME DATE DEFAULT GETDATE() NOT NULL,      -- DEFAULT, NOT NULL Constraint
PRIMARY KEY (MID, PID), -- PRIMARY Key Constraint
foreign key (MID) REFERENCES Member(MID),   -- FORENIGN Key Constraint
foreign key (PID) REFERENCES Project(PID)    -- FORENIGN Key Constraint
);
go
```

Plan_

```
CREATE TABLE Plan_(
MID VARCHAR(20) NOT NULL,
PID VARCHAR(10) NOT NULL,
START_TIME DATE NOT NULL,
END_TIME DATE NOT NULL,
PRIMARY KEY(MID,PID),
foreign key (MID) REFERENCES Member(MID),   -- FORENIGN Key Constraint
foreign key (PID) REFERENCES Project(PID),     -- FORENIGN Key Constraint
CHECK ( END_TIME > START_TIME),
CHECK ( START_TIME > GETDATE())
);
go
```

Channel

```
CREATE TABLE Channel(
CID VARCHAR(10) NOT NULL PRIMARY KEY,
POINT INT NOT NULL DEFAULT 0,
RATING CHAR(1) NOT NULL DEFAULT 'C' CHECK(RATING IN('S', 'A', 'B', 'C'))
);
go
```

Subscribe

```
CREATE TABLE Subscribe(
CID VARCHAR(10) NOT NULL,
MID VARCHAR(20) NOT NULL,
STIME DATE DEFAULT GETDATE() NOT NULL,
SNUM INT NOT NULL,
PRIMARY KEY(CID, MID),-- primary key constraint
foreign key (CID) REFERENCES Channel(CID),     -- foreign key constraint
foreign key (MID) REFERENCES Member (MID),     -- foreign key constraint
CHECK (SNUM >= 0),
);
go
```

donate

```
CREATE TABLE   donate(
donated_id VARCHAR(10) NOT NULL,
member_id VARCHAR(20) NOT NULL,
DTIME DATE DEFAULT GETDATE() NOT NULL,
PRIMARY KEY (donated_id, member_id),          -- primary key constraint
FOREIGN KEY (member_id) REFERENCES Member(MID)          -- foreign key constraint
);
go
```

LogTable : update / delete Trigger's Log store

```
CREATE TABLE LogTable(
        logdate datetime NOT NULL,
        logUser varchar(30) NULL,
        logTableName varchar(50) NULL,
        logOperation varchar(30) NULL,
        logRecord varchar(max) NULL
)
go
```

3. Server side Programming

3.1 Procedure

Explanation : This procedures is the procedure responsible for deletion at each table.

```
create proc sp_deleteChannel @c1 varchar(10)

as
begin
delete from dbo.Channel
where CID = @c1;
end

create proc sp_deletedonate @c1 int, @c2 varchar(20)

as
begin
delete from dbo.donate
where donated_id = @c1 AND member_id = @c2;
end

create proc sp_deletedFunding @c1 varchar(20), @c2 varchar(20)

as
begin
delete from dbo.Funding
where MID = @c1 AND PID = @c2;
end

create proc sp_deletedMember @c1 varchar(20)

as
begin
delete from dbo.Member
where MID = @c1;
end

create proc sp_deletedPlan_ @c1 varchar(20), @c2 varchar(20)
```

```
as
begin
delete from dbo.Plan_
where MID = @c1 AND PID = @c2;
end

create proc sp_deletedProject @c1 varchar(20)

as
begin
delete from dbo.Project
where PID = @c1;
end

create proc sp_deletedSubsribe @c1 varchar(10), @c2 varchar(20)

as
begin
delete from dbo.Subscribe
where CID = @c1 AND MID = @c2;
end
```

Explanation : These procedures are procedures responsible for inserting data into each table.

```
create proc sp_insertdonate
@c2 varchar(20),
@c3 date,
@c4 int
as
begin
insert into dbo.donate values (@c2,@c3,@c4)
end

create proc sp_insertFunding
@c1 varchar(20),
@c2 varchar(20),
@c3 int,
@c4 date
as
```

```
begin
insert into dbo.Funding values (@c1,@c2,@c3,@c4)
end

create proc sp_insertMember
@c1 varchar(20),
@c2 varchar(30),
@c3 varchar(20),
@c4 varchar(20),
@c5 varchar(22),
@c6 varchar(30),
@c7 int
as
begin
insert into dbo.Member values (@c1,@c2,@c3,@c4, @c5, @c6, @c7)
end

create proc sp_insertPlan_
@c1 varchar(20),
@c2 varchar(20),
@c3 date,
@c4 date
as
begin
insert into dbo.Plan_ values (@c1,@c2,@c3,@c4)
end

create proc sp_insertProject
@c1 varchar(20),
@c2 varchar(20),
@c3 varchar(20),
@c4 varchar(50),
@c5 int
as
begin
insert into dbo.Project values (@c1,@c2,@c3,@c4,@c5)
end
```

```
create proc sp_insertSubscribe
@c1 varchar(10),
@c2 varchar(20),
@c3 date,
@c4 int
as
begin
insert into dbo.Subscribe values (@c1,@c2,@c3,@c4)
end
```

Explanation : These procedures serve to update data in the table.

```
use RewardCrowdfunding
go

create proc sp_updateChannel
@c1 varchar(10),
@c2 int,
@c3 char(1)
as
begin
update dbo.Channel set POINT = @c2, RATING = @c3
where CID = @c1;
end

create proc sp_updatedonate
@c1 int,
@c2 varchar(20),
@c3 date,
@c4 int
as
begin
update dbo.donate set DTIME = @c3, DMONEY = @c4
where donated_id = @c1 AND member_id = @c2;
end

create proc sp_updateFunding
@c1 varchar(20),
@c2 varchar(20),
@c3 int,
```

```
@c4 date
as
begin
update dbo.Funding set FMoney = @c3, FTIME = @c4
where MID = @c1 AND PID = @c2;
end

create proc sp_updateMember
@c1 varchar(20),
@c2 varchar(30),
@c3 varchar(20),
@c4 varchar(20),
@c5 varchar(22),
@c6 varchar(30),
@c7 int
as
begin
update dbo.Member set PW = @c2, fname = @c3, lname = @c4, phoneNum = @c5, eMail =
@c6, Score = @c7
where MID = @c1;
end

create proc sp_updatePlan_
@c1 varchar(20),
@c2 varchar(20),
@c3 date,
@c4 date
as
begin
update dbo.Plan_ set START_TIME = @c3, END_TIME = @c4
where MID = @c1 AND PID = @c2;
end

create proc sp_updateProject
@c1 varchar(20),
@c2 varchar(20),
@c3 varchar(20),
@c4 varchar(50),
```

```
@c5 int
as
begin
update dbo.Project set PName = @c2, PField = @c3, PDetails = @c4, goalMoney = @c5
where PID = @c1
end

create proc sp_updateSubscribe
@c1 varchar(10),
@c2 varchar(20),
@c3 date,
@c4 int
as
begin
update dbo.Subscribe set STIME = @c3, SNUM = @c4
where CID = @c1 AND MID = @c2;
end
```

3.2 Functions

Explanation : This function returns a list of columns in a table.

```
create function [dbo].[getColumnsList](@tname as 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.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(@tname)
```

```
select * from getColumnsList('Member');
```

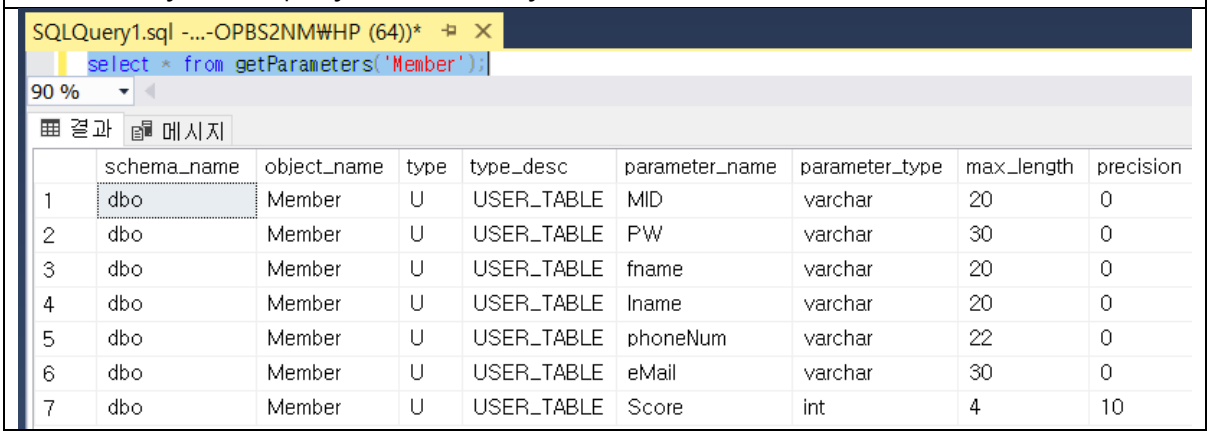
90 %

결과 메시지

	schema_name	object_name	type	type_desc	column_name	column_type	max_length
1	dbo	Member	U	USER_TABLE	MID	varchar	20
2	dbo	Member	U	USER_TABLE	PW	varchar	30
3	dbo	Member	U	USER_TABLE	fname	varchar	20
4	dbo	Member	U	USER_TABLE	lname	varchar	20
5	dbo	Member	U	USER_TABLE	phoneNum	varchar	22
6	dbo	Member	U	USER_TABLE	eMail	varchar	30
7	dbo	Member	U	USER_TABLE	Score	int	4

Explanation : This function returns a Parameters in a table

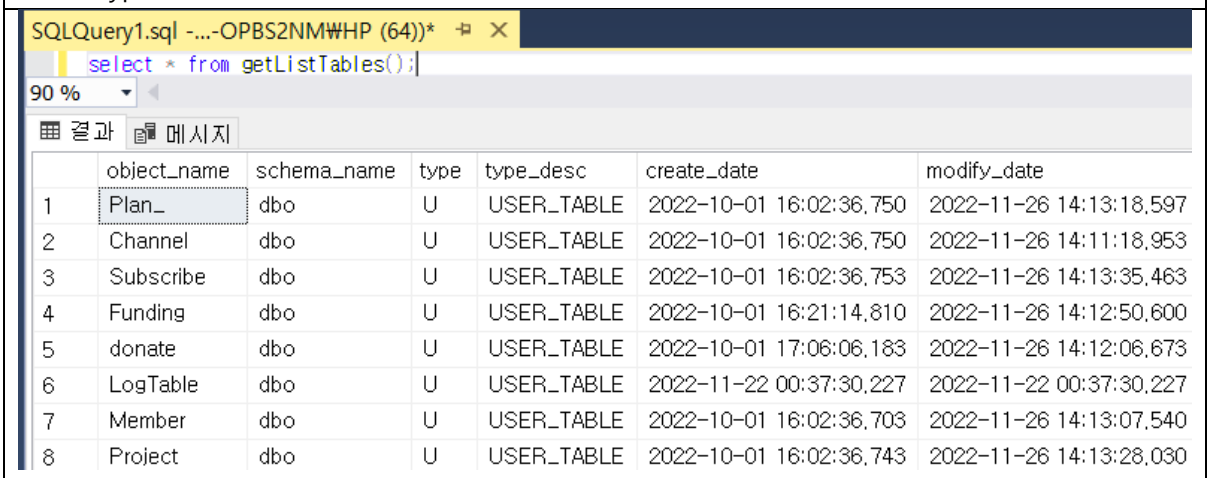
```
create function [dbo].[getParameters](@tname as varchar(30))
returns table
as return
select SCHEMA_NAME(schema_id) AS schema_name
,o.name AS object_name, o.type, o.type_desc
,p.name AS parameter_name
,TYPE_NAME(p.user_type_id) AS parameter_type
,p.max_length
,p.precision

FROM sys.objects AS o, sys.columns AS p

WHERE o.object_id = p.object_id and o.object_id = OBJECT_ID(@tname)
```



	schema_name	object_name	type	type_desc	parameter_name	parameter_type	max_length	precision
1	dbo	Member	U	USER_TABLE	MID	varchar	20	0
2	dbo	Member	U	USER_TABLE	PW	varchar	30	0
3	dbo	Member	U	USER_TABLE	fname	varchar	20	0
4	dbo	Member	U	USER_TABLE	lname	varchar	20	0
5	dbo	Member	U	USER_TABLE	phoneNum	varchar	22	0
6	dbo	Member	U	USER_TABLE	eMail	varchar	30	0
7	dbo	Member	U	USER_TABLE	Score	int	4	10

Explanation : This function returns a list of table

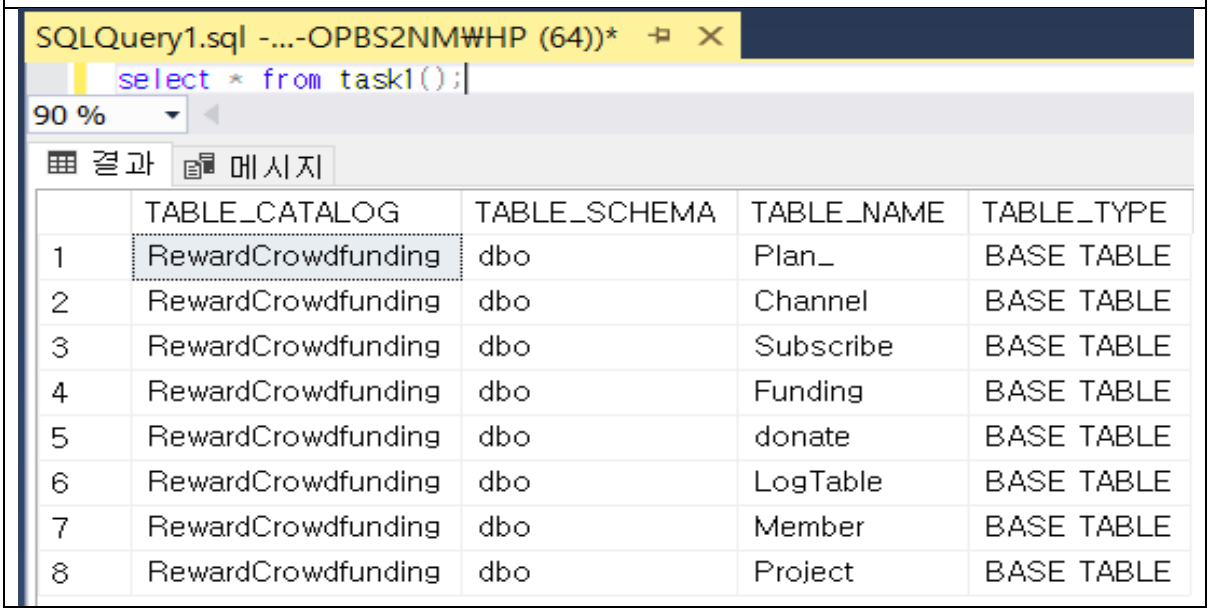
```
create function [dbo].[getListTables]()
returns table
as
return SELECT name AS object_name, SCHEMA_NAME(schema_id) AS schema_name
,type, type_desc, create_date, modify_date
FROM sys.objects
where type = 'U'
```



	object_name	schema_name	type	type_desc	create_date	modify_date
1	Plan_	dbo	U	USER_TABLE	2022-10-01 16:02:36.750	2022-11-26 14:13:18.597
2	Channel	dbo	U	USER_TABLE	2022-10-01 16:02:36.750	2022-11-26 14:11:18.953
3	Subscribe	dbo	U	USER_TABLE	2022-10-01 16:02:36.753	2022-11-26 14:13:35.463
4	Funding	dbo	U	USER_TABLE	2022-10-01 16:21:14.810	2022-11-26 14:12:50.600
5	donate	dbo	U	USER_TABLE	2022-10-01 17:06:06.183	2022-11-26 14:12:06.673
6	LogTable	dbo	U	USER_TABLE	2022-11-22 00:37:30.227	2022-11-22 00:37:30.227
7	Member	dbo	U	USER_TABLE	2022-10-01 16:02:36.703	2022-11-26 14:13:07.540
8	Project	dbo	U	USER_TABLE	2022-10-01 16:02:36.743	2022-11-26 14:13:28.030

Explanation : This function returns information from the generated table.

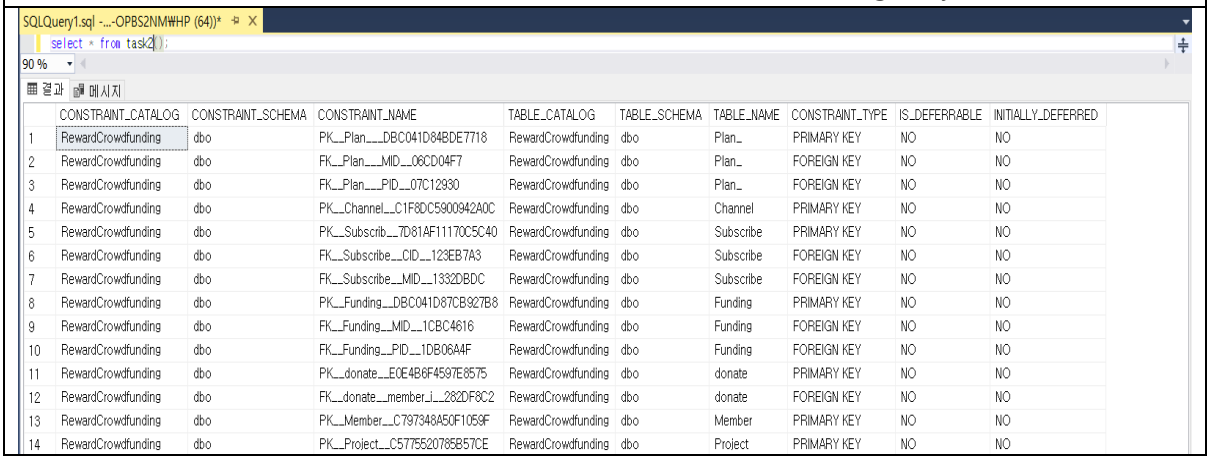
```
create function [dbo].[task1]()
returns table
return select * from INFORMATION_SCHEMA.TABLES
```



	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	RewardCrowdfunding	dbo	Plan_	BASE TABLE
2	RewardCrowdfunding	dbo	Channel	BASE TABLE
3	RewardCrowdfunding	dbo	Subscribe	BASE TABLE
4	RewardCrowdfunding	dbo	Funding	BASE TABLE
5	RewardCrowdfunding	dbo	donate	BASE TABLE
6	RewardCrowdfunding	dbo	LogTable	BASE TABLE
7	RewardCrowdfunding	dbo	Member	BASE TABLE
8	RewardCrowdfunding	dbo	Project	BASE TABLE

Explanation : This function returns all Primary-keys and Foreign-Keys

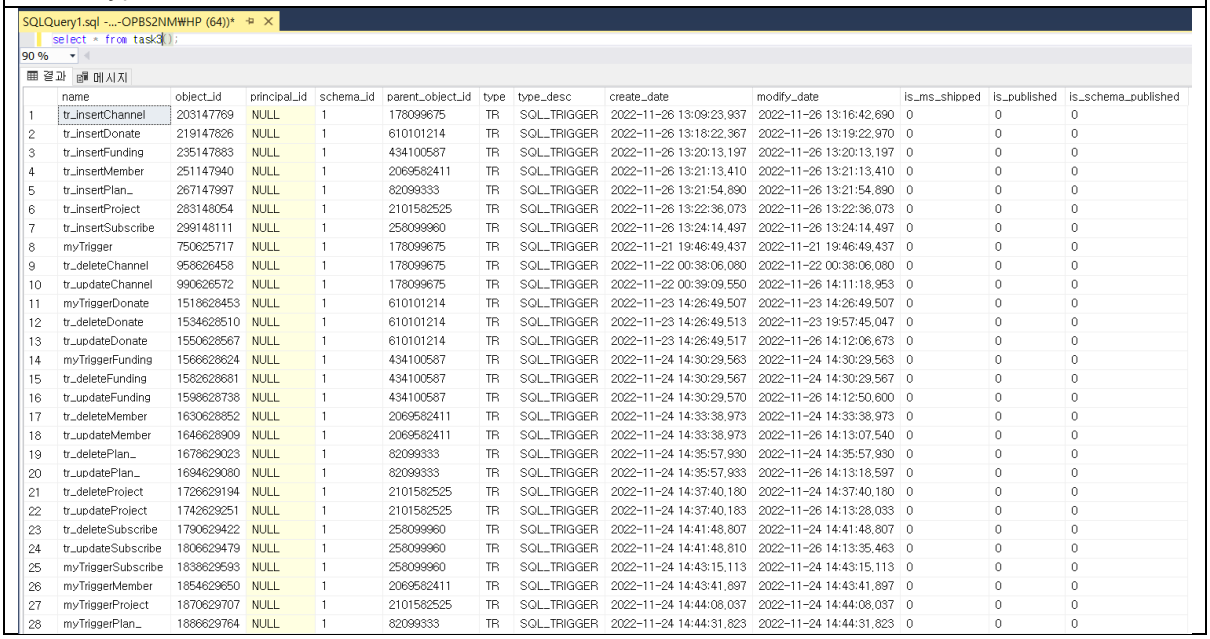
```
create function [dbo].[task2]()
returns table
return select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
where CONSTRAINT_TYPE='PRIMARY KEY'or CONSTRAINT_TYPE='foreign key'
```



	CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	IS_DEFERRABLE	INITIALLY_DEFERRED
1	RewardCrowdfunding	dbo	PK__Plan___DBC041D84BDE7718	RewardCrowdfunding	dbo	Plan_	PRIMARY KEY	NO	NO
2	RewardCrowdfunding	dbo	FK__Plan___MID__06CD04F7	RewardCrowdfunding	dbo	Plan_	FOREIGN KEY	NO	NO
3	RewardCrowdfunding	dbo	FK__Plan___PID__07C12930	RewardCrowdfunding	dbo	Plan_	FOREIGN KEY	NO	NO
4	RewardCrowdfunding	dbo	PK__Channel__C1F8DC5900942A0C	RewardCrowdfunding	dbo	Channel	PRIMARY KEY	NO	NO
5	RewardCrowdfunding	dbo	PK__Subscrib__7D81AF11170C5C40	RewardCrowdfunding	dbo	Subscribe	PRIMARY KEY	NO	NO
6	RewardCrowdfunding	dbo	FK__Subscribe__CID__123EB7A3	RewardCrowdfunding	dbo	Subscribe	FOREIGN KEY	NO	NO
7	RewardCrowdfunding	dbo	FK__Subscribe__MID__1332DBDC	RewardCrowdfunding	dbo	Subscribe	FOREIGN KEY	NO	NO
8	RewardCrowdfunding	dbo	PK__Funding__DBC041D87CB927B8	RewardCrowdfunding	dbo	Funding	PRIMARY KEY	NO	NO
9	RewardCrowdfunding	dbo	FK__Funding__MID__1CBC4616	RewardCrowdfunding	dbo	Funding	FOREIGN KEY	NO	NO
10	RewardCrowdfunding	dbo	FK__Funding__PID__1DB06A4F	RewardCrowdfunding	dbo	Funding	FOREIGN KEY	NO	NO
11	RewardCrowdfunding	dbo	PK__donate__E0E4B6F4597E8575	RewardCrowdfunding	dbo	donate	PRIMARY KEY	NO	NO
12	RewardCrowdfunding	dbo	FK__donate__member_i__282DF8C2	RewardCrowdfunding	dbo	donate	FOREIGN KEY	NO	NO
13	RewardCrowdfunding	dbo	PK__Member__C797348A50F1059F	RewardCrowdfunding	dbo	Member	PRIMARY KEY	NO	NO
14	RewardCrowdfunding	dbo	PK__Project__C5775520785B57CE	RewardCrowdfunding	dbo	Project	PRIMARY KEY	NO	NO

Explanation : This function returns all user-defined triggers

```
create function [dbo].[task3]()
returns table
return select * from sys.objects
where type = 'TR'
```



SQLQuery1.sql -...-OPBS2NM₩HP (64))*

```
select * from task3();
```

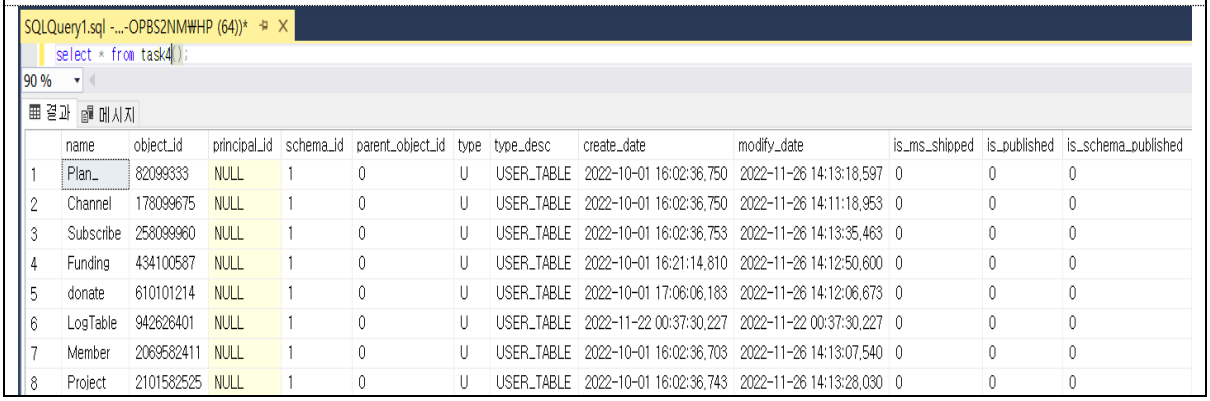
90 %

결과 메시지

	name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published	is_schema_published
1	tr_insertChannel	203147769	NULL	1	178099675	TR	SQL_TRIGGER	2022-11-26 13:09:23.937	2022-11-26 13:16:42.690	0	0	0
2	tr_insertDonate	219147826	NULL	1	610101214	TR	SQL_TRIGGER	2022-11-26 13:18:22.367	2022-11-26 13:19:22.970	0	0	0
3	tr_insertFunding	235147883	NULL	1	434100587	TR	SQL_TRIGGER	2022-11-26 13:20:13.197	2022-11-26 13:20:13.197	0	0	0
4	tr_insertMember	251147940	NULL	1	2069582411	TR	SQL_TRIGGER	2022-11-26 13:21:13.410	2022-11-26 13:21:13.410	0	0	0
5	tr_insertPlan_	267147997	NULL	1	82099333	TR	SQL_TRIGGER	2022-11-26 13:21:54.890	2022-11-26 13:21:54.890	0	0	0
6	tr_insertProject	283148054	NULL	1	2101582525	TR	SQL_TRIGGER	2022-11-26 13:22:36.073	2022-11-26 13:22:36.073	0	0	0
7	tr_insertSubscribe	299148111	NULL	1	258099960	TR	SQL_TRIGGER	2022-11-26 13:24:14.497	2022-11-26 13:24:14.497	0	0	0
8	myTrigger	750625717	NULL	1	178099675	TR	SQL_TRIGGER	2022-11-21 19:46:49.437	2022-11-21 19:46:49.437	0	0	0
9	tr_deleteChannel	958626458	NULL	1	178099675	TR	SQL_TRIGGER	2022-11-22 00:38:06.080	2022-11-22 00:38:06.080	0	0	0
10	tr_updateChannel	990626572	NULL	1	178099675	TR	SQL_TRIGGER	2022-11-22 00:39:09.550	2022-11-26 14:11:18.953	0	0	0
11	myTriggerDonate	1518628453	NULL	1	610101214	TR	SQL_TRIGGER	2022-11-23 14:26:49.507	2022-11-23 14:26:49.507	0	0	0
12	tr_deleteDonate	1534628510	NULL	1	610101214	TR	SQL_TRIGGER	2022-11-23 14:26:49.513	2022-11-23 19:57:45.047	0	0	0
13	tr_updateDonate	1550628567	NULL	1	610101214	TR	SQL_TRIGGER	2022-11-23 14:26:49.517	2022-11-26 14:12:06.673	0	0	0
14	myTriggerFunding	1566628624	NULL	1	434100587	TR	SQL_TRIGGER	2022-11-24 14:30:29.563	2022-11-24 14:30:29.563	0	0	0
15	tr_deleteFunding	1582628681	NULL	1	434100587	TR	SQL_TRIGGER	2022-11-24 14:30:29.567	2022-11-24 14:30:29.567	0	0	0
16	tr_updateFunding	1598628738	NULL	1	434100587	TR	SQL_TRIGGER	2022-11-24 14:30:29.570	2022-11-26 14:12:50.600	0	0	0
17	tr_deleteMember	1630628852	NULL	1	2069582411	TR	SQL_TRIGGER	2022-11-24 14:33:38.973	2022-11-24 14:33:38.973	0	0	0
18	tr_updateMember	1646628909	NULL	1	2069582411	TR	SQL_TRIGGER	2022-11-24 14:33:38.973	2022-11-26 14:13:07.540	0	0	0
19	tr_deletePlan_	1678629023	NULL	1	82099333	TR	SQL_TRIGGER	2022-11-24 14:35:57.930	2022-11-24 14:35:57.930	0	0	0
20	tr_updatePlan_	1694629080	NULL	1	82099333	TR	SQL_TRIGGER	2022-11-24 14:35:57.933	2022-11-26 14:13:18.597	0	0	0
21	tr_deleteProject	1726629194	NULL	1	2101582525	TR	SQL_TRIGGER	2022-11-24 14:37:40.180	2022-11-24 14:37:40.180	0	0	0
22	tr_updateProject	1742629251	NULL	1	2101582525	TR	SQL_TRIGGER	2022-11-24 14:37:40.183	2022-11-26 14:13:28.033	0	0	0
23	tr_deleteSubscribe	1790629422	NULL	1	258099960	TR	SQL_TRIGGER	2022-11-24 14:41:48.807	2022-11-24 14:41:48.807	0	0	0
24	tr_updateSubscribe	1806629479	NULL	1	258099960	TR	SQL_TRIGGER	2022-11-24 14:41:48.810	2022-11-26 14:13:35.463	0	0	0
25	myTriggerSubscribe	1838629593	NULL	1	258099960	TR	SQL_TRIGGER	2022-11-24 14:43:15.113	2022-11-24 14:43:15.113	0	0	0
26	myTriggerMember	1854629650	NULL	1	2069582411	TR	SQL_TRIGGER	2022-11-24 14:43:41.897	2022-11-24 14:43:41.897	0	0	0
27	myTriggerProject	1870629707	NULL	1	2101582525	TR	SQL_TRIGGER	2022-11-24 14:44:08.037	2022-11-24 14:44:08.037	0	0	0
28	myTriggerPlan_	1886629764	NULL	1	82099333	TR	SQL_TRIGGER	2022-11-24 14:44:31.823	2022-11-24 14:44:31.823	0	0	0

Explanation : This function returns row-count for all user-defined tables in the database

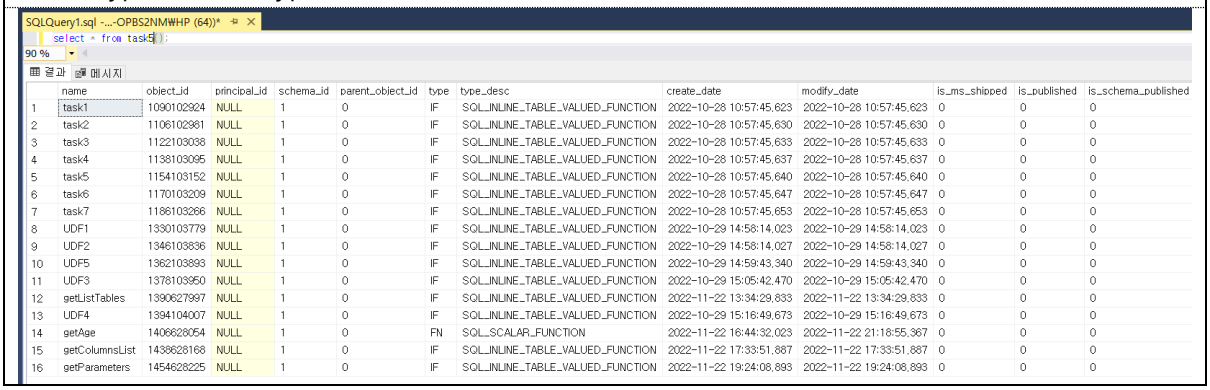
```
create function [dbo].[task4]()
returns table
return select * from sys.objects
where type = 'U'
```



	name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published	is_schema_published
1	Plan_	82099333	NULL	1	0	U	USER_TABLE	2022-10-01 16:02:36.750	2022-11-26 14:13:18.597	0	0	0
2	Channel	178099675	NULL	1	0	U	USER_TABLE	2022-10-01 16:02:36.750	2022-11-26 14:11:18.953	0	0	0
3	Subscribe	258099960	NULL	1	0	U	USER_TABLE	2022-10-01 16:02:36.753	2022-11-26 14:13:35.463	0	0	0
4	Funding	434100587	NULL	1	0	U	USER_TABLE	2022-10-01 16:21:14.810	2022-11-26 14:12:50.600	0	0	0
5	donate	610101214	NULL	1	0	U	USER_TABLE	2022-10-01 17:06:06.183	2022-11-26 14:12:06.673	0	0	0
6	LogTable	942626401	NULL	1	0	U	USER_TABLE	2022-11-22 00:37:30.227	2022-11-22 00:37:30.227	0	0	0
7	Member	2069582411	NULL	1	0	U	USER_TABLE	2022-10-01 16:02:36.703	2022-11-26 14:13:07.540	0	0	0
8	Project	2101582525	NULL	1	0	U	USER_TABLE	2022-10-01 16:02:36.743	2022-11-26 14:13:28.030	0	0	0

Explanation : This function returns the list of all user defined functions.

```
create function [dbo].[task5]()
returns table
return select * from sys.objects
where type = 'IF' or type ='FN'
```



SQLQuery1.sql -...-OPBS2NM₩HP (64))*

```
select * from task5();
```

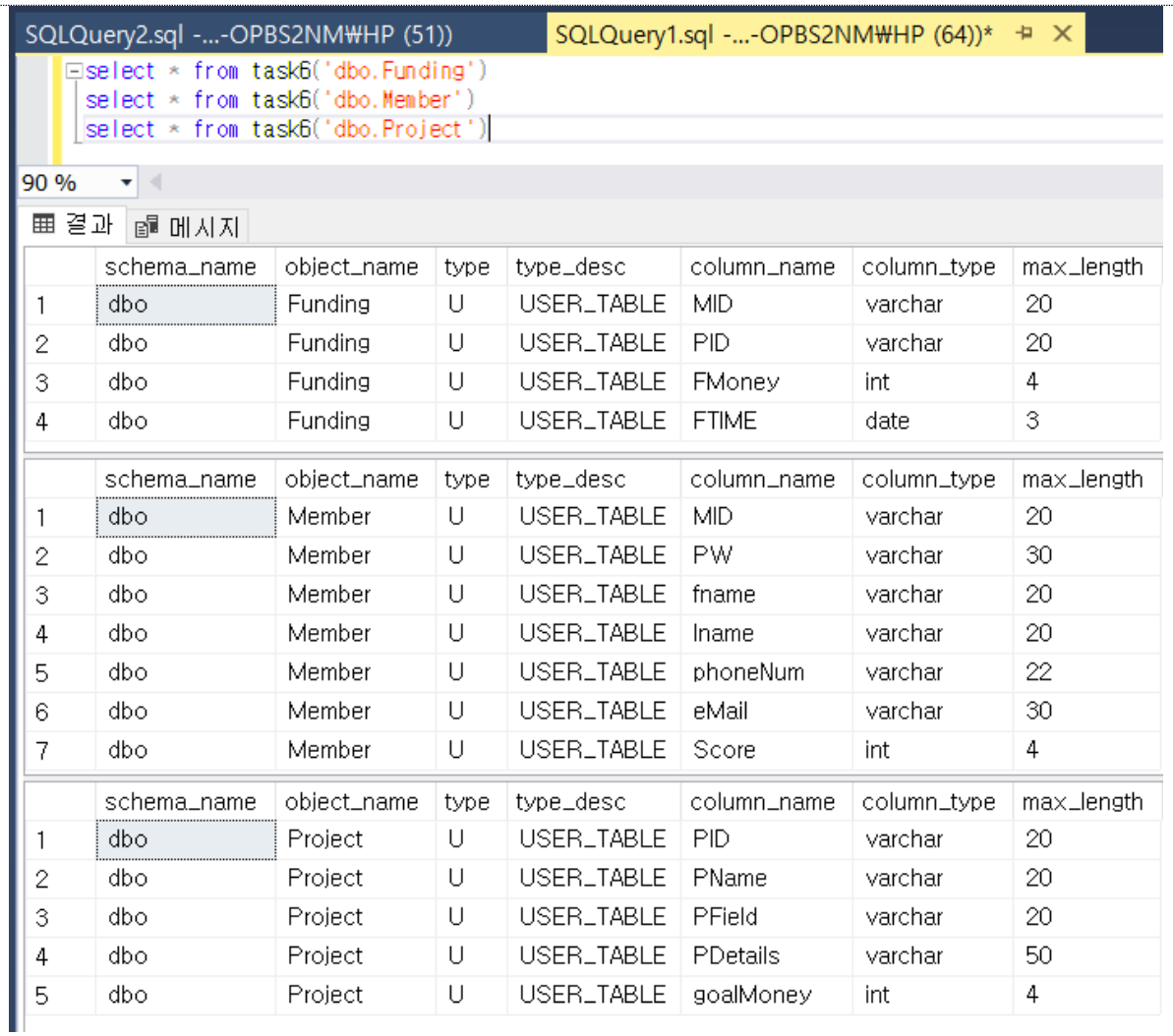
90 %

결과 메시지

	name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published	is_schema_published
1	task1	1090102924	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-28 10:57:45.623	2022-10-28 10:57:45.623	0	0	0
2	task2	1106102981	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-28 10:57:45.630	2022-10-28 10:57:45.630	0	0	0
3	task3	1122103038	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-28 10:57:45.633	2022-10-28 10:57:45.633	0	0	0
4	task4	1138103095	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-28 10:57:45.637	2022-10-28 10:57:45.637	0	0	0
5	task5	1154103152	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-28 10:57:45.640	2022-10-28 10:57:45.640	0	0	0
6	task6	1170103209	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-28 10:57:45.647	2022-10-28 10:57:45.647	0	0	0
7	task7	1186103266	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-28 10:57:45.653	2022-10-28 10:57:45.653	0	0	0
8	UDF1	1330103779	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-29 14:58:14.023	2022-10-29 14:58:14.023	0	0	0
9	UDF2	1346103836	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-29 14:58:14.027	2022-10-29 14:58:14.027	0	0	0
10	UDF5	1362103893	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-29 14:59:43.340	2022-10-29 14:59:43.340	0	0	0
11	UDF3	1378103950	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-29 15:05:42.470	2022-10-29 15:05:42.470	0	0	0
12	getListTables	1390627997	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-11-22 13:34:29.833	2022-11-22 13:34:29.833	0	0	0
13	UDF4	1394104007	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-10-29 15:16:49.673	2022-10-29 15:16:49.673	0	0	0
14	getAge	1406628054	NULL	1	0	FN	SQL_SCALAR_FUNCTION	2022-11-22 16:44:32.023	2022-11-22 21:18:55.367	0	0	0
15	getColumnsList	1438628168	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-11-22 17:33:51.887	2022-11-22 17:33:51.887	0	0	0
16	getParameters	1454628225	NULL	1	0	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	2022-11-22 19:24:08.893	2022-11-22 19:24:08.893	0	0	0

Explanation : This function takes a table name as input and returns the column names with their types

```
create function [dbo].[task6](@frname as varchar(30))

returns table
as return
select SCHEMA_NAME(schema_id) AS schema_name,
o.name as object_name, o.type,o.type_desc
,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(@frname)
```



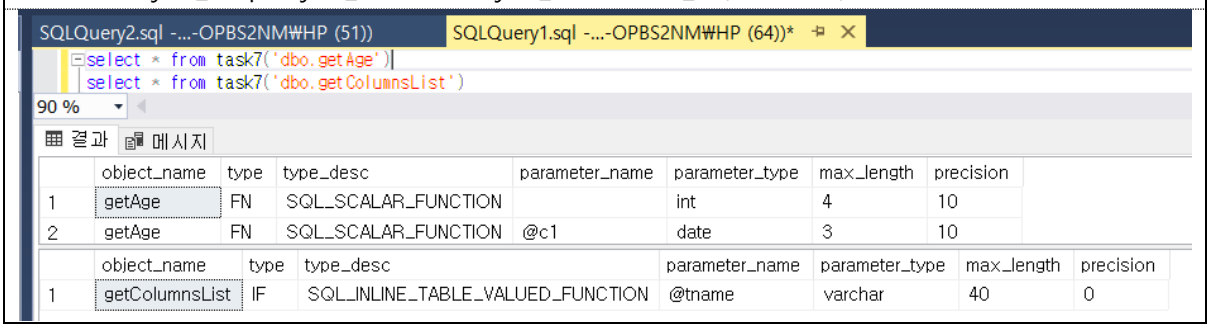
	schema_name	object_name	type	type_desc	column_name	column_type	max_length
1	dbo	Funding	U	USER_TABLE	MID	varchar	20
2	dbo	Funding	U	USER_TABLE	PID	varchar	20
3	dbo	Funding	U	USER_TABLE	FMoney	int	4
4	dbo	Funding	U	USER_TABLE	FTIME	date	3

	schema_name	object_name	type	type_desc	column_name	column_type	max_length
1	dbo	Member	U	USER_TABLE	MID	varchar	20
2	dbo	Member	U	USER_TABLE	PW	varchar	30
3	dbo	Member	U	USER_TABLE	fname	varchar	20
4	dbo	Member	U	USER_TABLE	lname	varchar	20
5	dbo	Member	U	USER_TABLE	phoneNum	varchar	22
6	dbo	Member	U	USER_TABLE	eMail	varchar	30
7	dbo	Member	U	USER_TABLE	Score	int	4

	schema_name	object_name	type	type_desc	column_name	column_type	max_length
1	dbo	Project	U	USER_TABLE	PID	varchar	20
2	dbo	Project	U	USER_TABLE	PName	varchar	20
3	dbo	Project	U	USER_TABLE	PField	varchar	20
4	dbo	Project	U	USER_TABLE	PDetails	varchar	50
5	dbo	Project	U	USER_TABLE	goalMoney	int	4

Explanation :: This function takes a function name as input and returns input parameters for that function with their types

```
create function [dbo].[task7](@frname as varchar(30))
returns table
return select o.name as object_name
,o.type,o.type_desc,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(@frname)
```

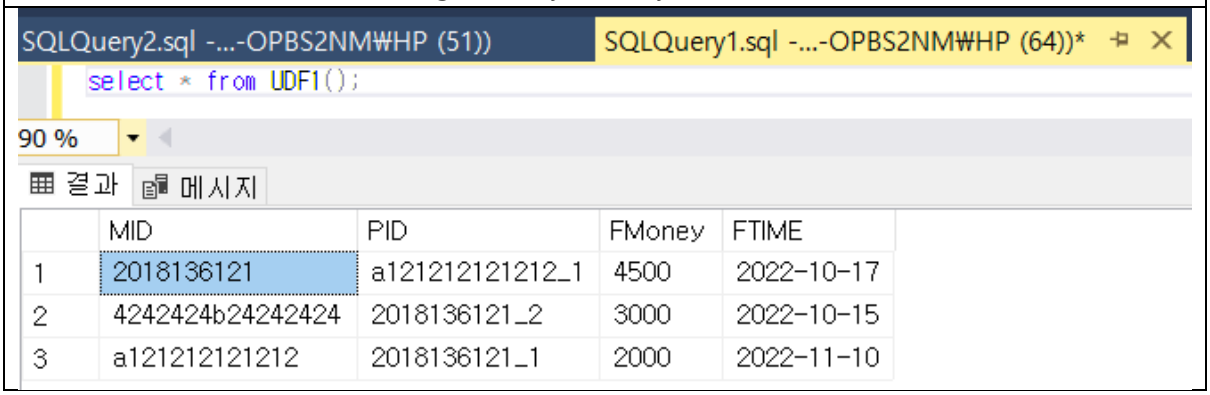


	object_name	type	type_desc	parameter_name	parameter_type	max_length	precision
1	getAge	FN	SQL_SCALAR_FUNCTION		int	4	10
2	getAge	FN	SQL_SCALAR_FUNCTION	@c1	date	3	10

	object_name	type	type_desc	parameter_name	parameter_type	max_length	precision
1	getColumnsList	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	@tname	varchar	40	0

Explanation : This function shows the three most expensive funding products.

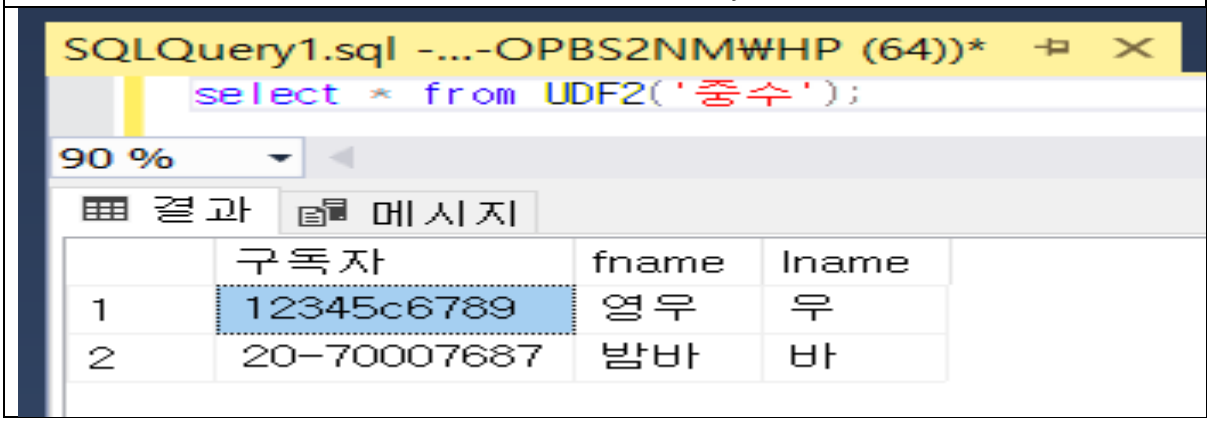
```
create function [dbo].[UDF1]()

returns table
return select TOP(3) * from Funding order by FMoney desc
```



	MID	PID	FMoney	FTIME
1	2018136121	a121212121212_1	4500	2022-10-17
2	4242424b24242424	2018136121_2	3000	2022-10-15
3	a121212121212	2018136121_1	2000	2022-11-10

Explanation : This function outputs subscribers who subscribe to a particular channel.

```
create function [dbo].[UDF2](@myID varchar(10))
returns table
return select MID as '구독자', fname, lname from Member
where MID in (select MID from Subscribe where CID= @myID)
```



Explanation : This function outputs the member ID that funding the project.

```
create function [dbo].[UDF3](@name varchar(40))
returns table
return select MID as '펀딩한 사람' from Funding
where PID = (select PID from   Project where PName = @name)
```

SQLQuery1.sql -...-OPBS2NM₩HP (64))*

```
select * from UDF3('자동코딩기기');
select * from UDF3('어디로든문');

select * from funding;
select * from project;
```

90 %

결과 메시지

	펀딩한 사람
1	2018136121

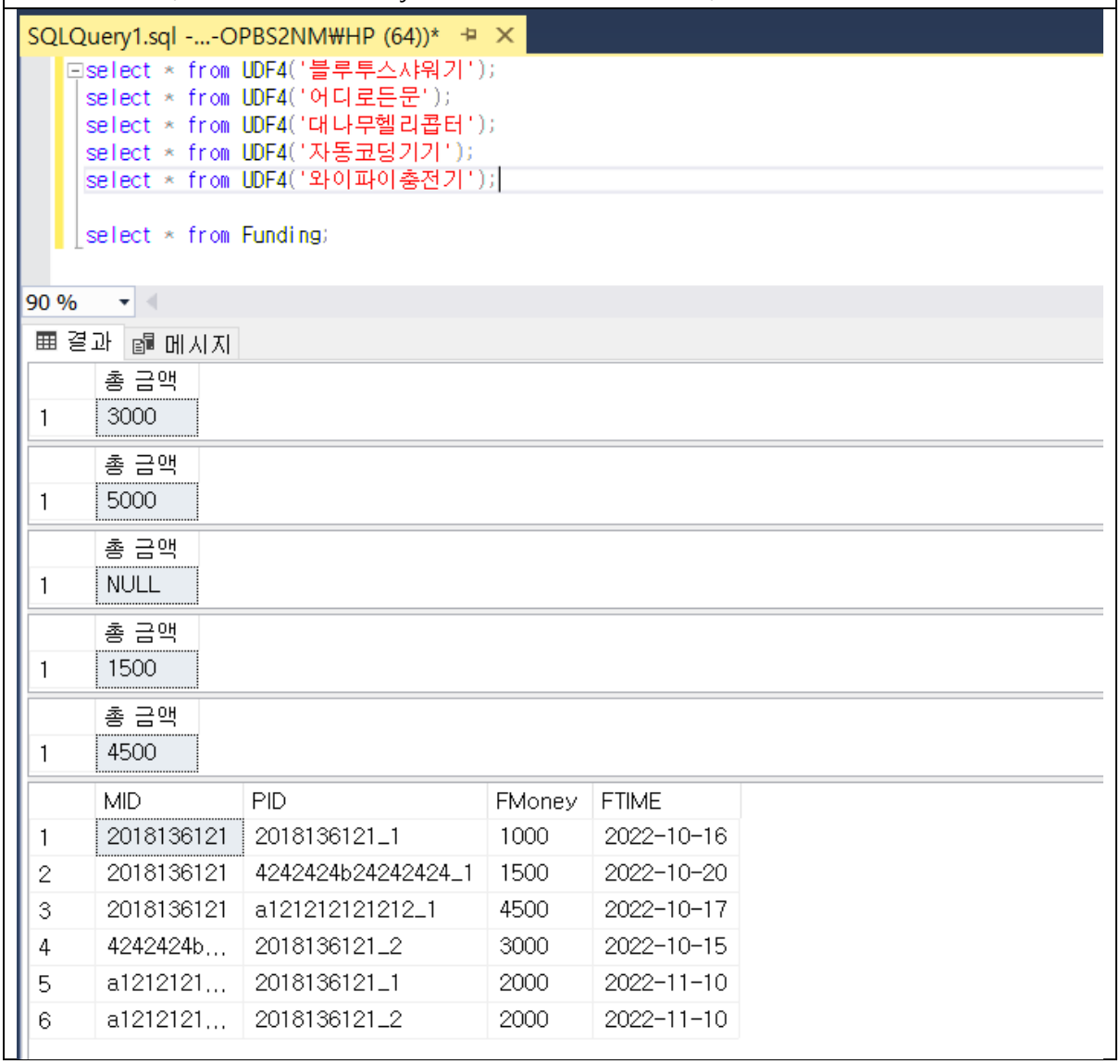
	펀딩한 사람
1	4242424b24242424
2	a121212121212

	MID	PID	FMoney	FTIME
1	2018136121	2018136121_1	1000	2022-10-16
2	2018136121	4242424b24242424_1	1500	2022-10-20
3	2018136121	a121212121212_1	4500	2022-10-17
4	4242424b24242424	2018136121_2	3000	2022-10-15
5	a121212121212	2018136121_1	2000	2022-11-10
6	a121212121212	2018136121_2	2000	2022-11-10

	PID	PName	PField	PDetails	goalMoney
1	2018136121_1	블루투스샤워기	샤워용품	선이 없이 샤워를 할 수 있음	120000
2	2018136121_2	어디로든문	운송	어디든지 갈 수 있음	7440000
3	2018136121_3	대나무헬리콥터	비행	머리에 달면 하늘을 날 수 있다	900000
4	4242424b24242424_1	자동코딩기기	전자기기	스스로 코드를 만들어주는 기계	120000
5	a121212121212_1	와이파이충전기	전기기기	와이파이를 저장해놨다가 사...	231000

Explanation : This function calculates the sum of the funds for the entered project.

```
create function [dbo].[UDF4](@name varchar(40)) returns table
as
return select sum(FMoney) as '총 금액' from Funding
where PID in ( select PID from Project where PName=@name)
```



	총 금액
1	3000

	총 금액
1	5000

	총 금액
1	NULL

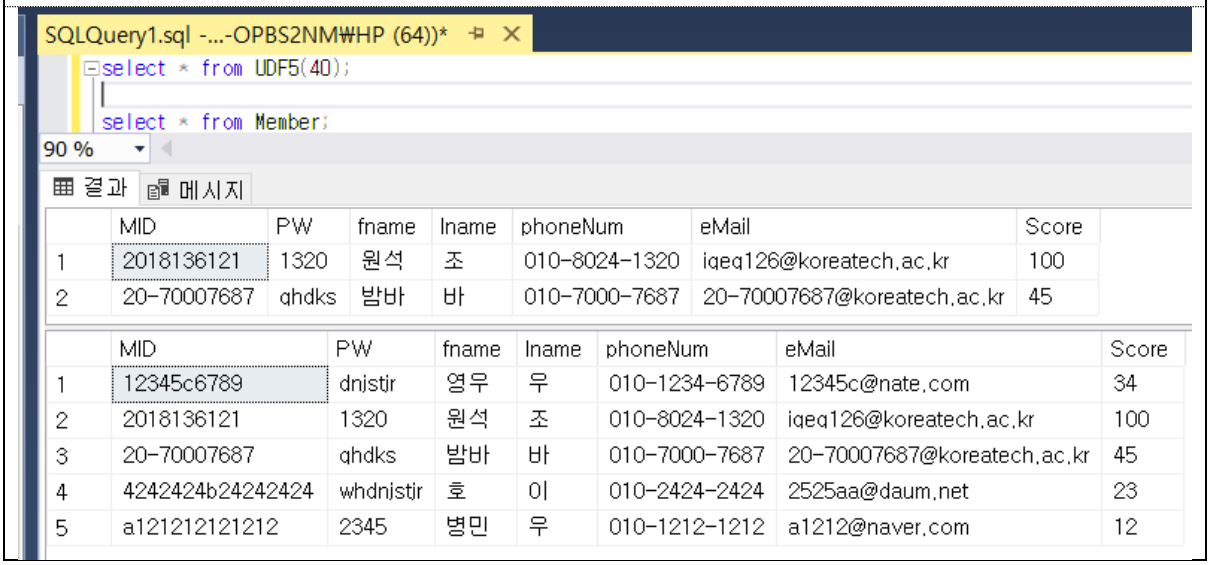
	총 금액
1	1500

	총 금액
1	4500

	MID	PID	FMoney	FTIME
1	2018136121	2018136121_1	1000	2022-10-16
2	2018136121	4242424b24242424_1	1500	2022-10-20
3	2018136121	a121212121212_1	4500	2022-10-17
4	4242424b...	2018136121_2	3000	2022-10-15
5	a1212121...	2018136121_1	2000	2022-11-10
6	a1212121...	2018136121_2	2000	2022-11-10

Explanation : This function searches for members with n points or more.

```
create function [dbo].[UDF5](@num int)-
returns table
as
return select * from Member
where Score >= @num
```



	MID	PW	fname	lname	phoneNum	eMail	Score
1	2018136121	1320	원석	조	010-8024-1320	iqeq126@koreatech.ac.kr	100
2	20-70007687	qhdks	밤바	바	010-7000-7687	20-70007687@koreatech.ac.kr	45

	MID	PW	fname	lname	phoneNum	eMail	Score
1	12345c6789	dnjstjr	영우	우	010-1234-6789	12345c@nate.com	34
2	2018136121	1320	원석	조	010-8024-1320	iqeq126@koreatech.ac.kr	100
3	20-70007687	qhdks	밤바	바	010-7000-7687	20-70007687@koreatech.ac.kr	45
4	4242424b24242424	whdnjstjr	호	이	010-2424-2424	2525aa@daum.net	23
5	a121212121212	2345	병민	우	010-1212-1212	a1212@naver.com	12

3.3 Trigger

3.3.1 Create Trigger

We created 4 triggers each for 7 tables, excluding logTable

We will explain the Channel table as an example..

Explanation : This prints message that change table

```
CREATE trigger [dbo].[myTrigger]
on [dbo].[Channel]
after insert, delete, update
as
begin
print 'there is a change in Channel table '
end
```

Explanation : This is delete trigger

```
CREATE trigger [dbo].[tr_deleteChannel]
on [dbo].[Channel]
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.Channel';
set @c4 = 'delete';
select @c5 = CID + ', ' + convert(varchar, POINT) + ', ' + RATING from deleted;

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

Explanation : This is insert trigger

```
CREATE trigger [dbo].[tr_insertChannel]
on [dbo].[Channel]
for insert
as
```

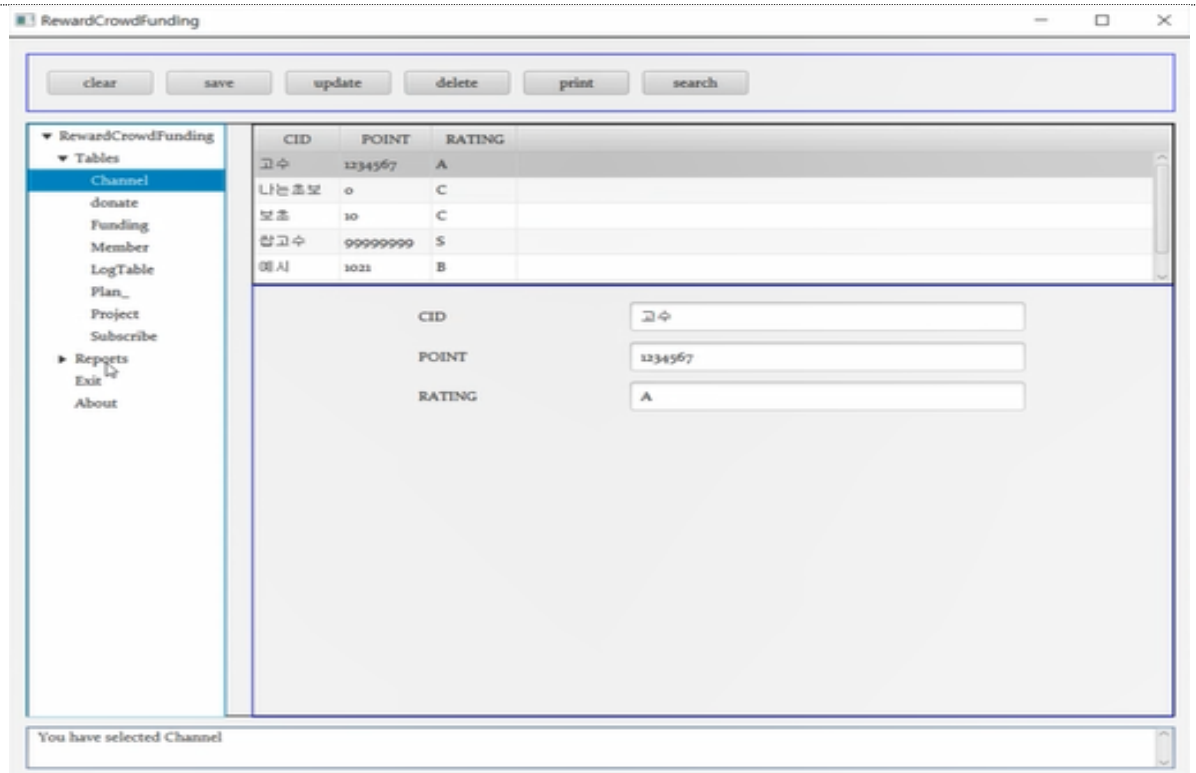
```
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = suser_name();
set @c3 = 'dbo.Channel';
set @c4 = 'insert';
select @c5 = CID + ', ' + convert(varchar, POINT) + ', ' + RATING from inserted;

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

Explanation : This is update trigger

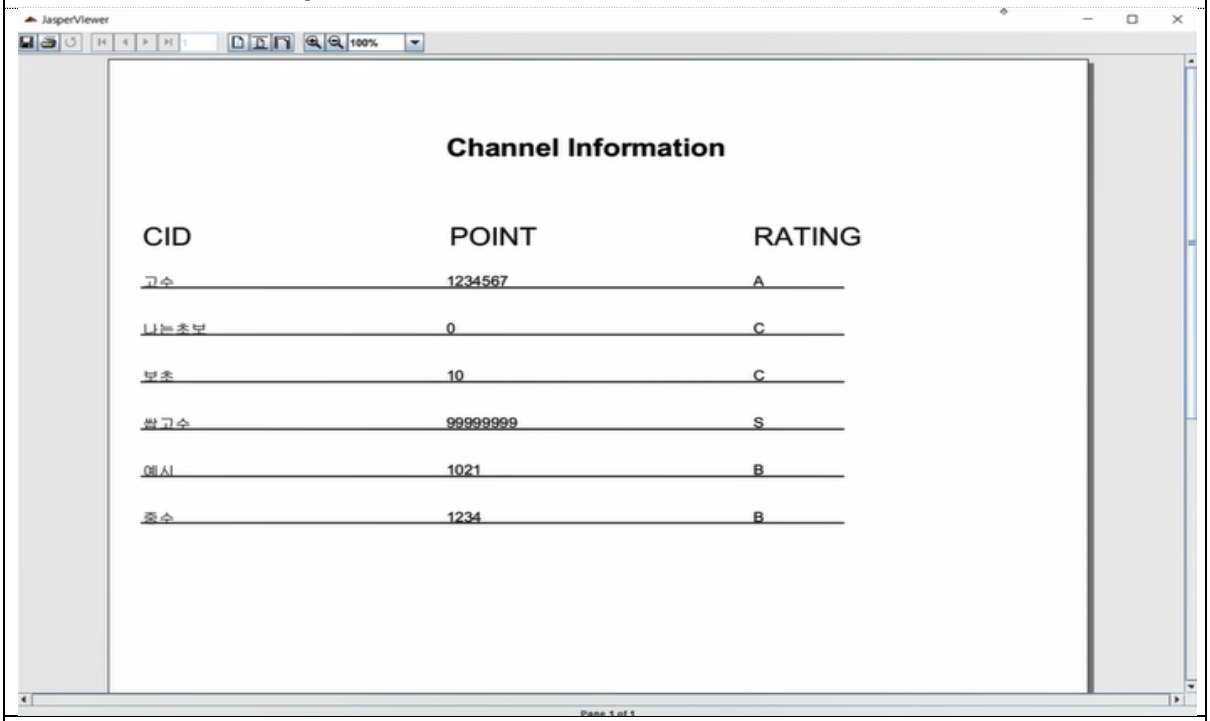
```
ALTER trigger [dbo].[tr_updateChannel]
on [dbo].[Channel] for update
as
begin
declare
@c1 datetime,
@c2 varchar(30),
@c3 varchar(50),
@c4 varchar(30),
@c5 varchar(max);
set @c1 = getdate();
set @c2 = suser_name();
set @c3 = 'dbo.Channel';
set @c4 = 'update';
select @c5 = CID + ', ' + convert(varchar, POINT) + ', ' + RATING from inserted;

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

4. Client side Programming

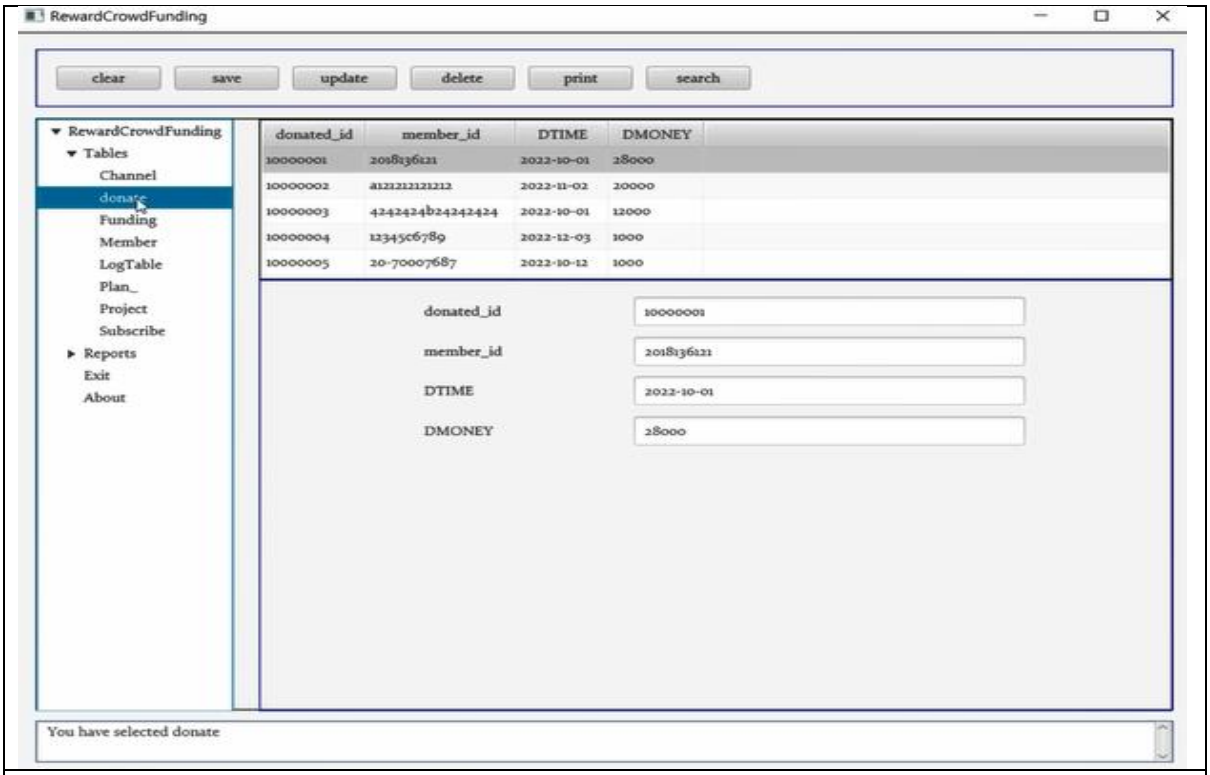
4.1 Channel Table



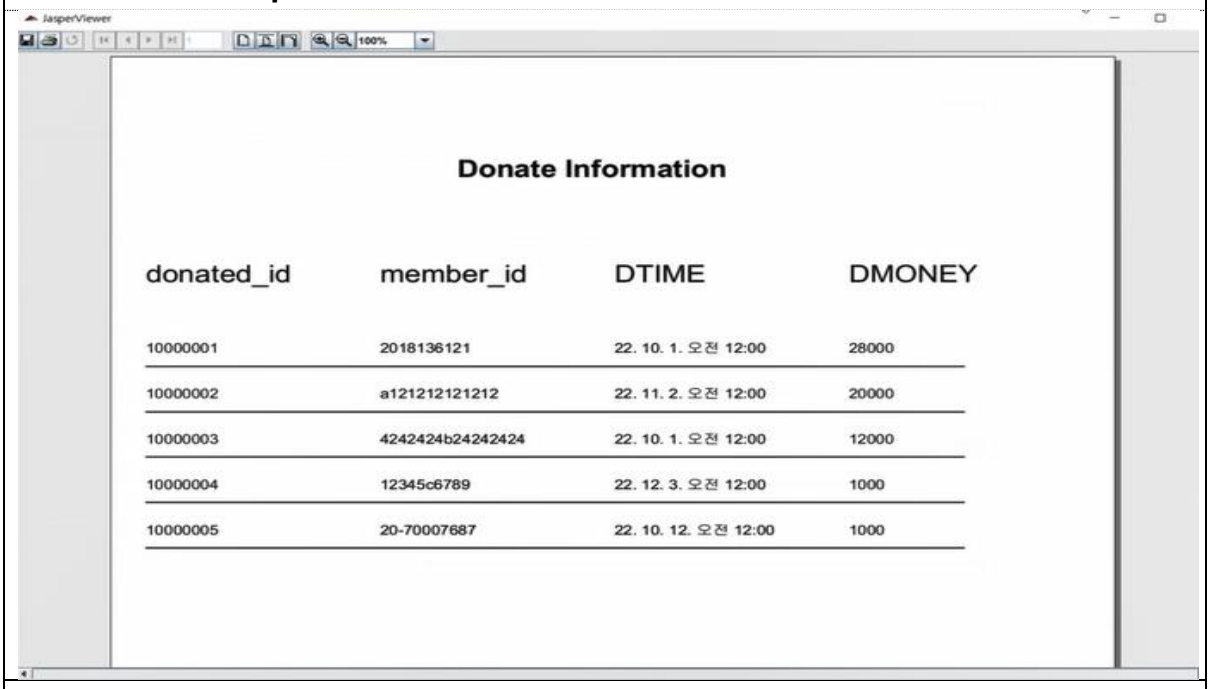
Channel Table JassperViewer



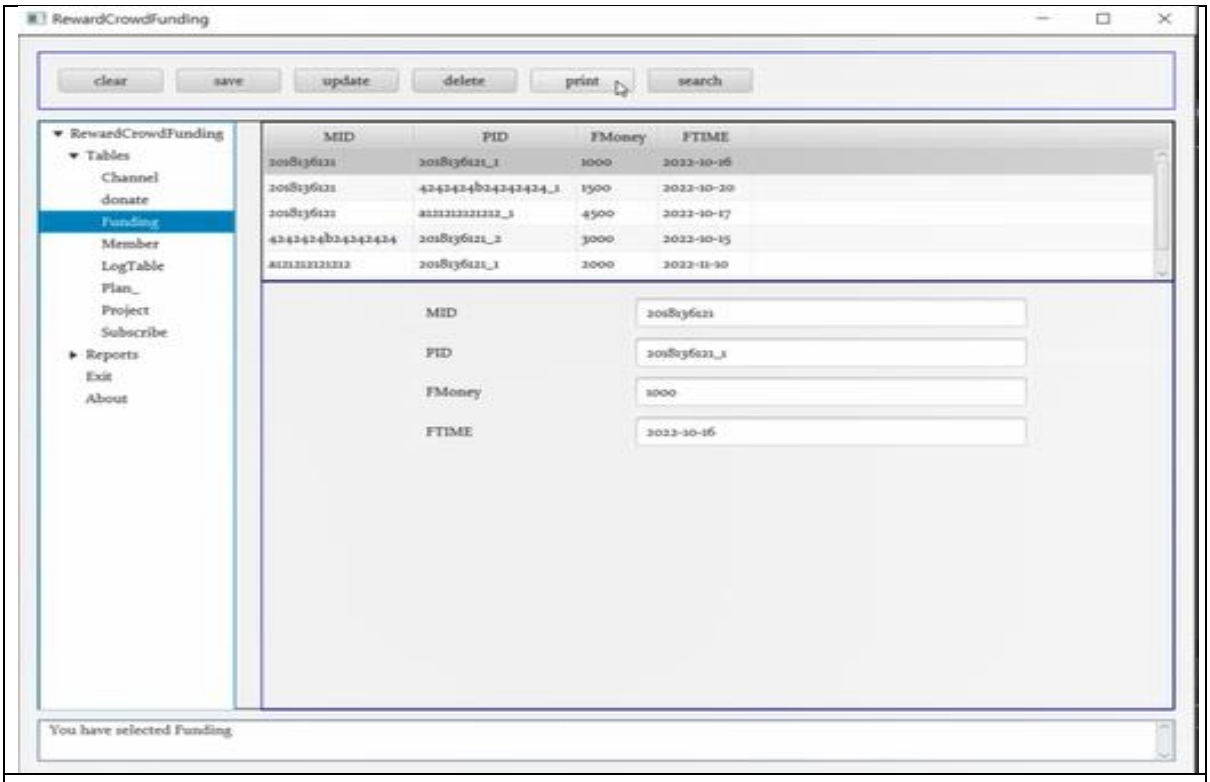
4.2 donate Table



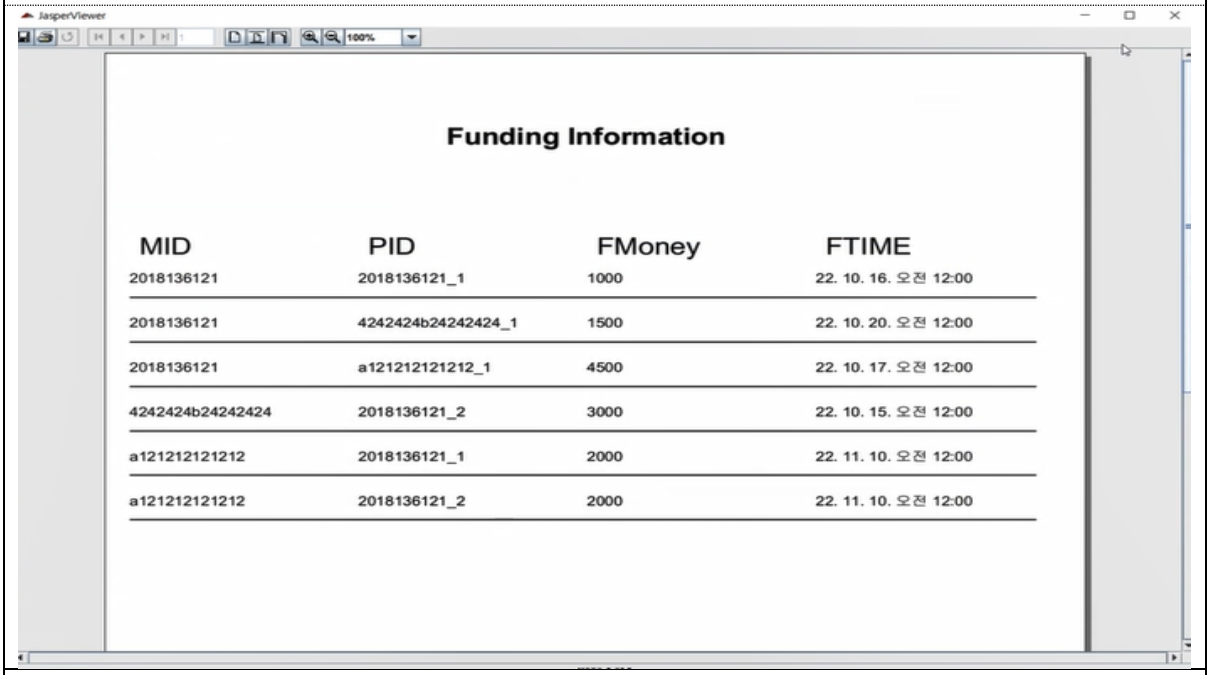
donate Table JasperViewer



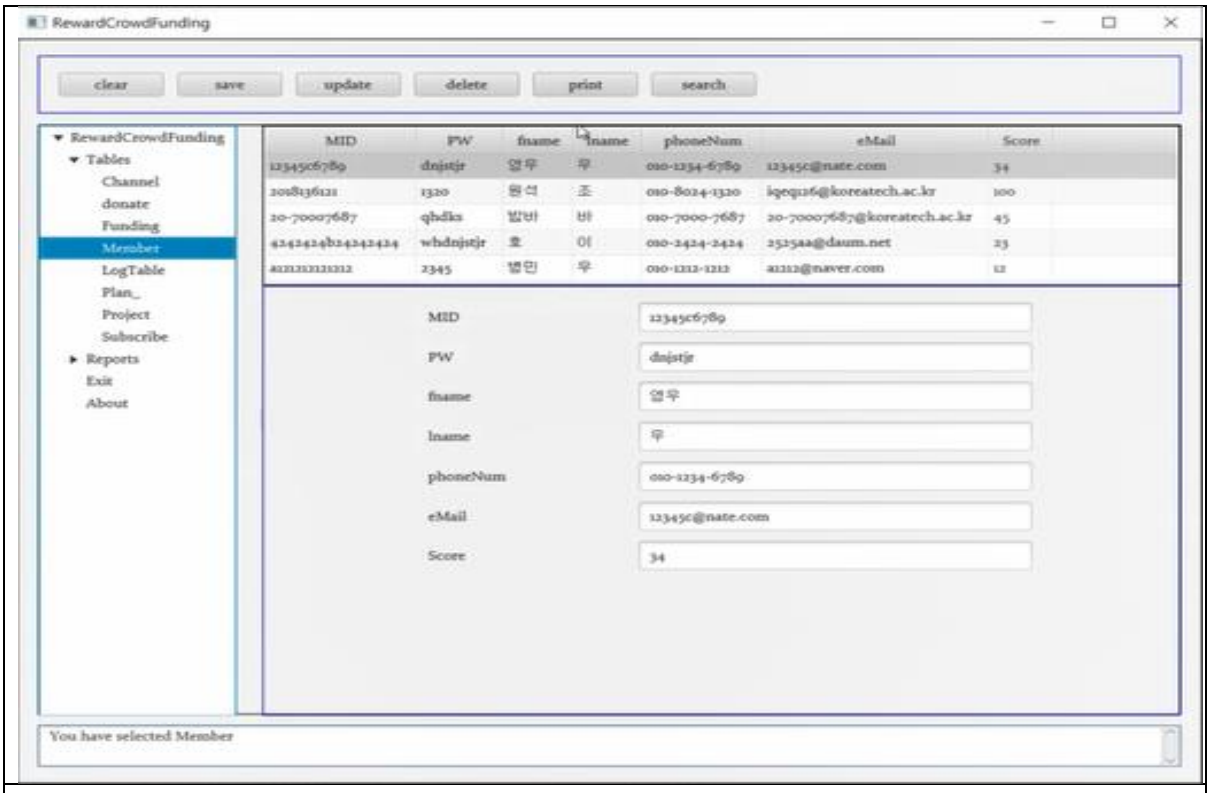
4.3 Funding Table



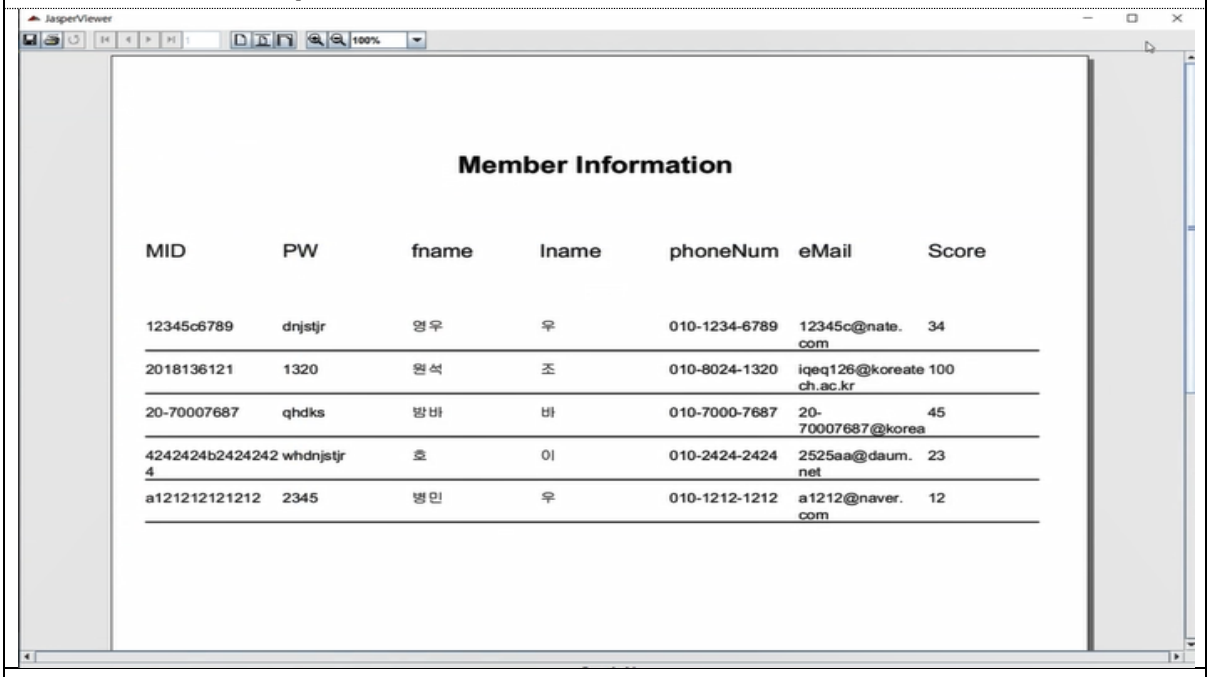
Funding Table JasperViewer



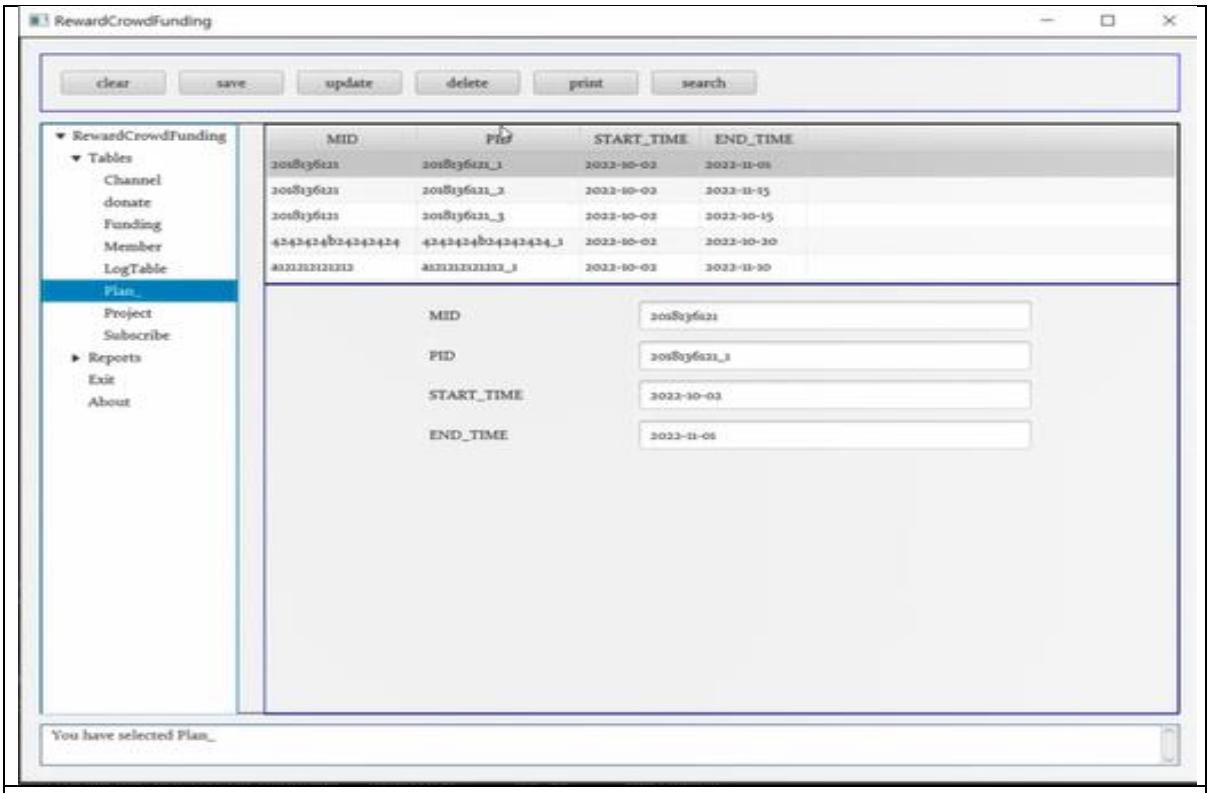
4.4 Member Table



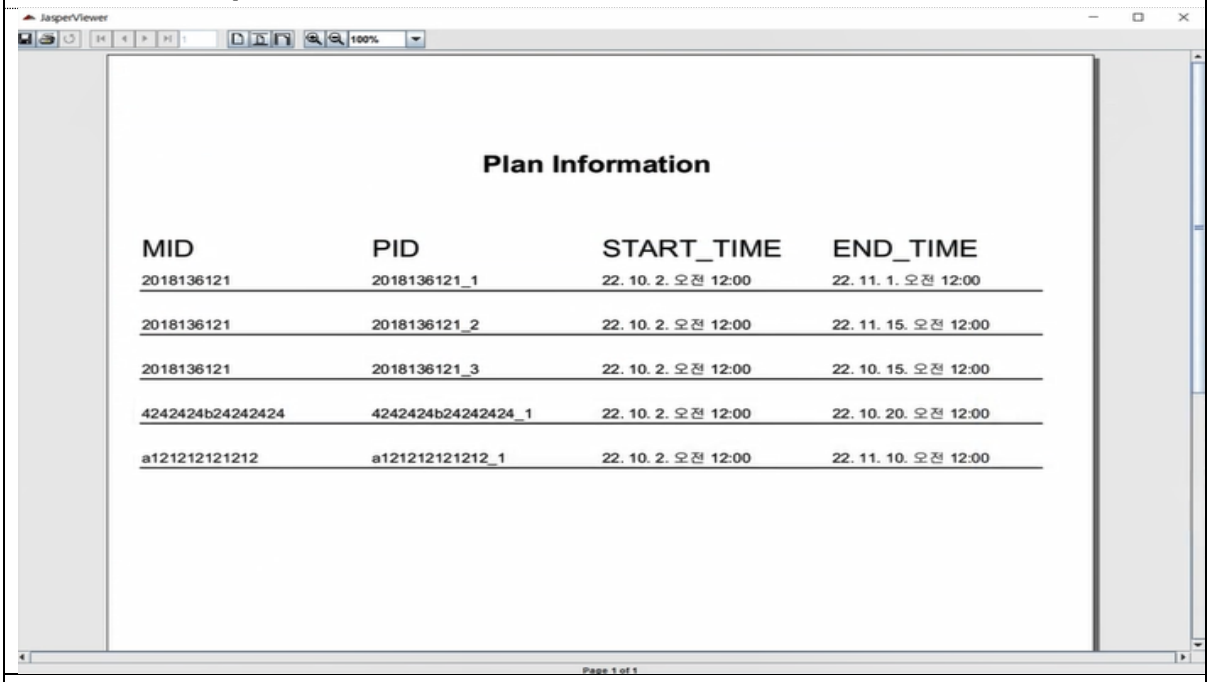
Member Table JasperViewer



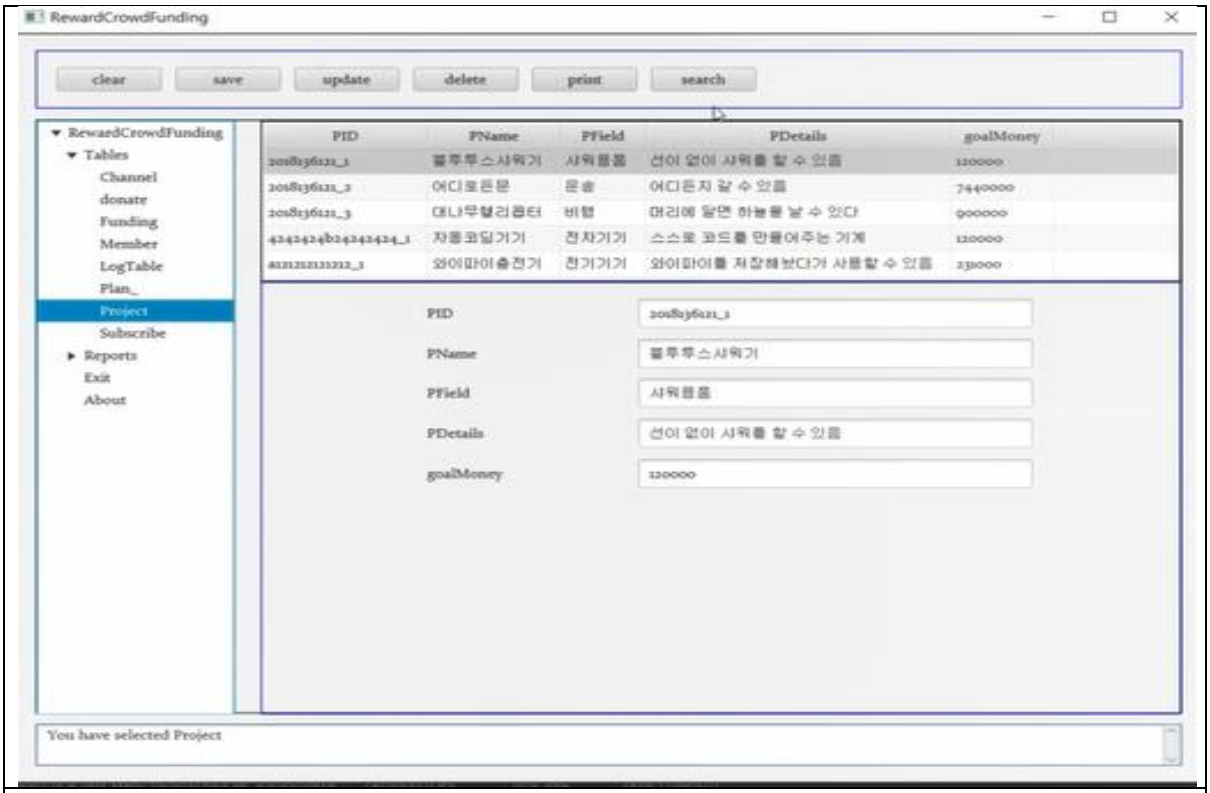
4.5 Plan Table



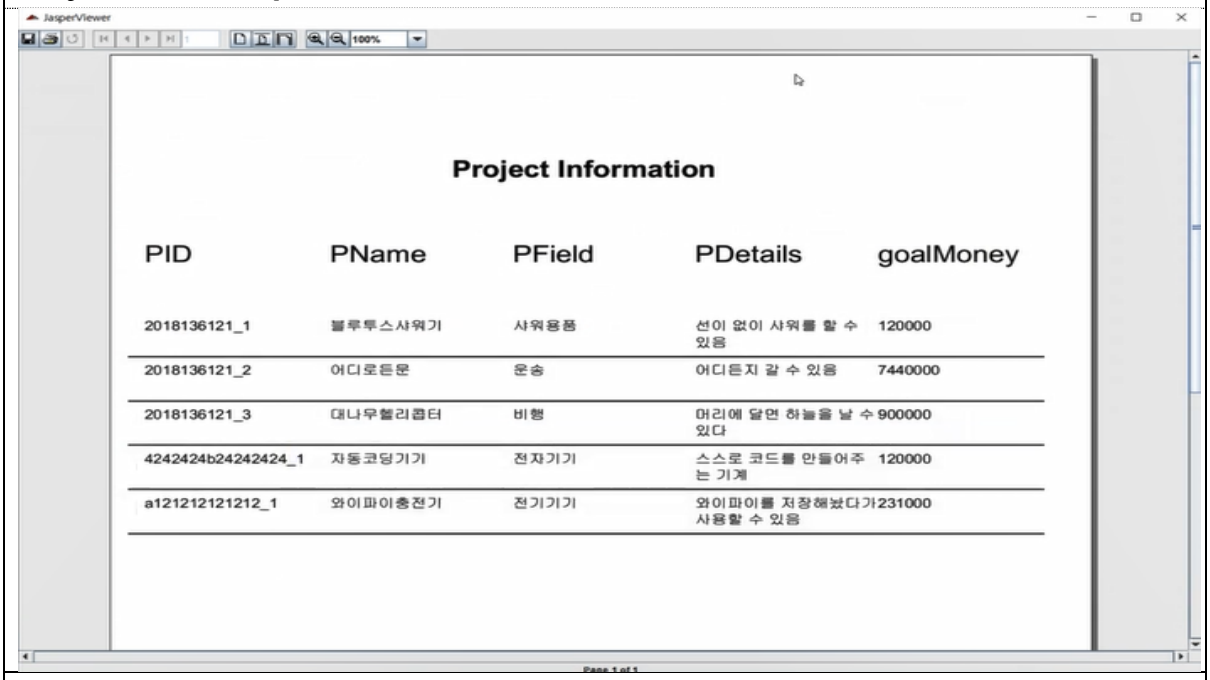
Plan Table JasperViewer



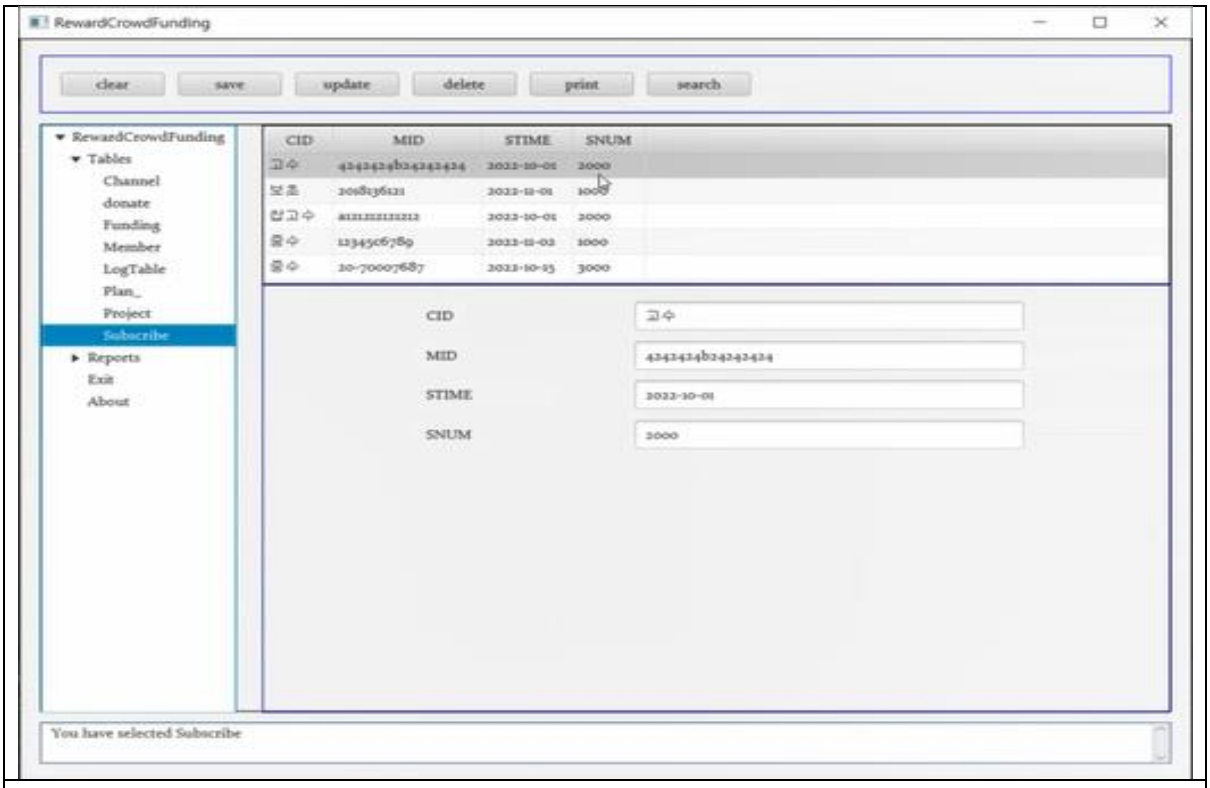
4.6 Project Table



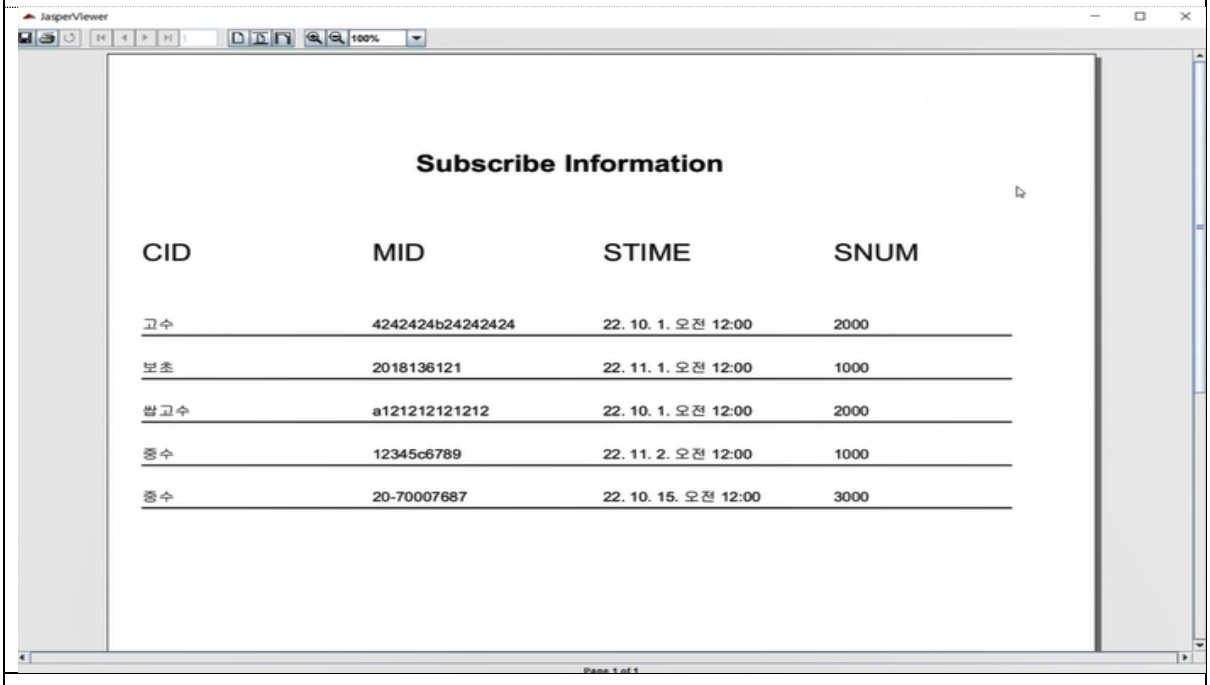
Project Table JasperViewer



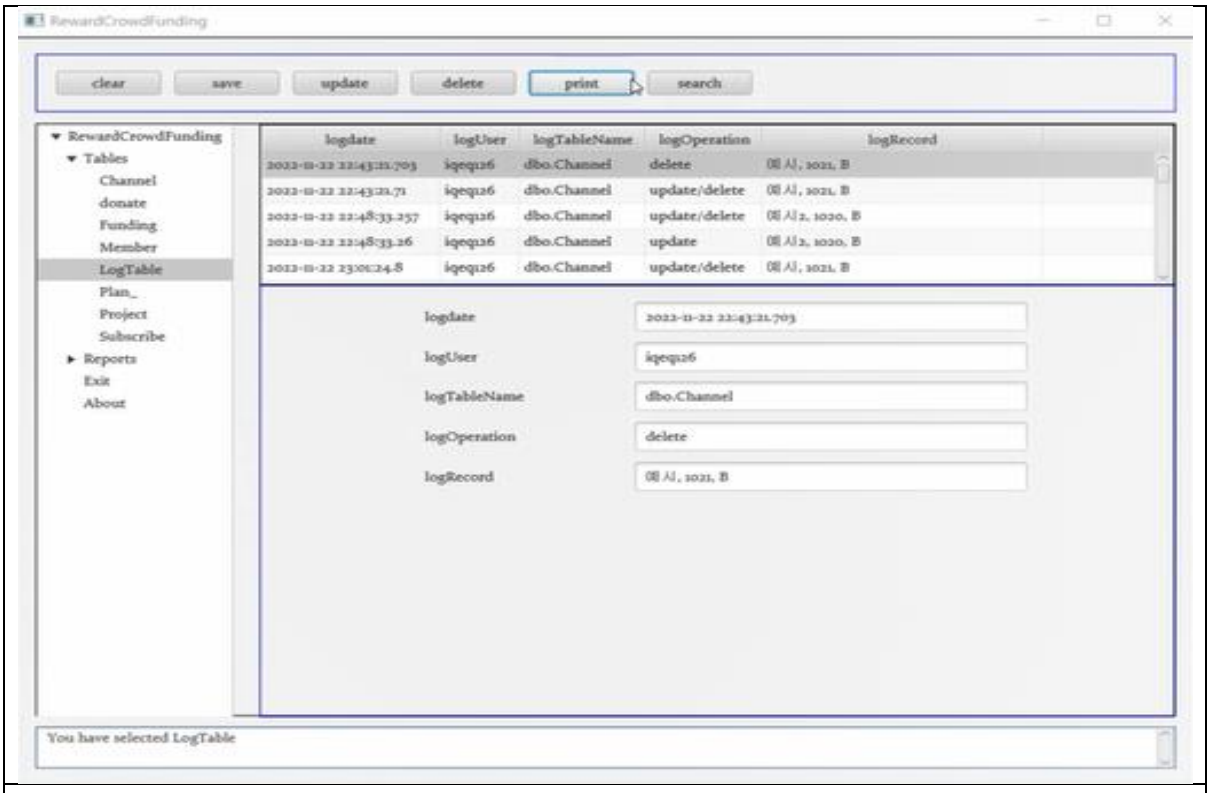
4.7 Subscribe Table



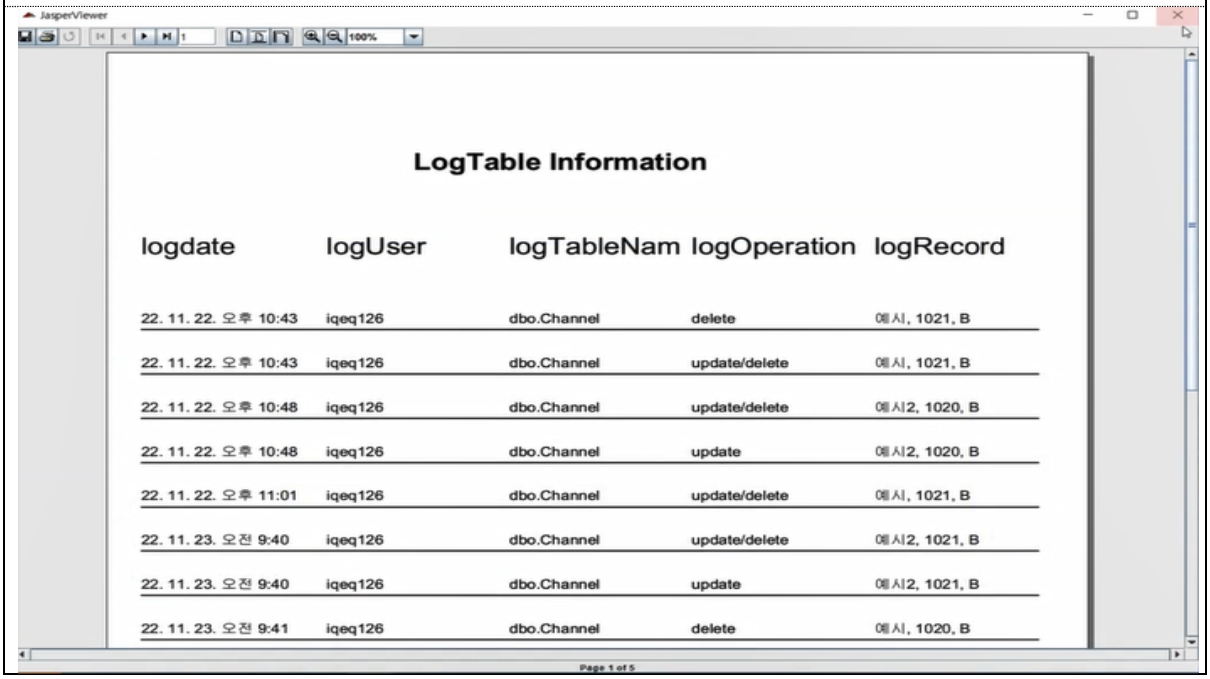
Subscribe Table JasperViewer



4.8 LogData Table



LogData Table JasperViewer



Tasks-3: Prepare a video presentation for the project (5-7 minuets)

[Press <ctrl> & <click>]

..₩Video₩Task3.mp4