Lab-04

2016136108 이찬규

컴퓨터공학부
cksrb4225@koreatech.ac.kr

1 Introduction

1.1 Assignment Introduction

About my database : My database is about convenience stores. A total of 5 tables were create through the design process (Branch, Buying, Contain, Customer, Stuff), and there are 2 scalar values functions, 17 table-valued functions, 15 stored procedures, and 6 triggers. I will explain in detail through the tasks below.

This task is to develop my own database designed so far as an application.
First, I do that establish a connection between your data base and Java Application using JDBC driver
Second, I create an appropriate GUI using Java language. my GUI should facilitate user to perform basic operations on database. So, I implemented the following four steps.

1. Display information about my database(tables, functions, stored procedures etc)
2. Navigate (see) records (data) from each table in my database.
3. Run functions to insert, update and delete records for each table in my database.
4. Print reports (data from different tables)

2 Task

2.1 Display information about my database(tables, functions, stored procedures etc)

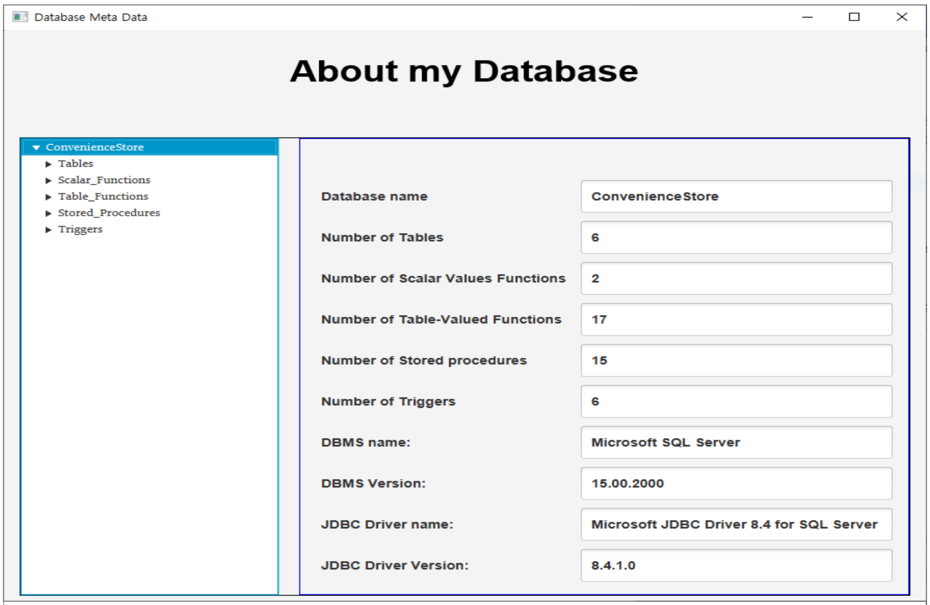
Information related to my database, such as Database name, Information of table, function, stored procedures and triggers, is printed.

Resolution

The fillgrid and nodestoTree functions were implemented by referring to the professor’s lecture and lecture materials, and correct values were displayed.

Depending on the situation, the getParameter() function and the getColumnList() function were used appropriately.

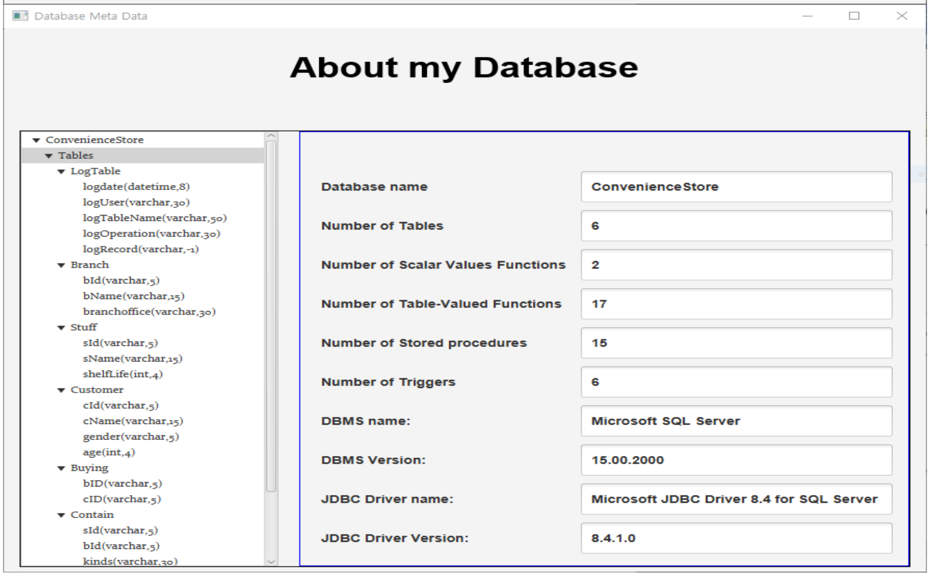
Results screen



About my Database

- ConvenienceStore
  - Tables
  - Scalar_Functions
  - Table_Functions
  - Stored_Procedures
  - Triggers

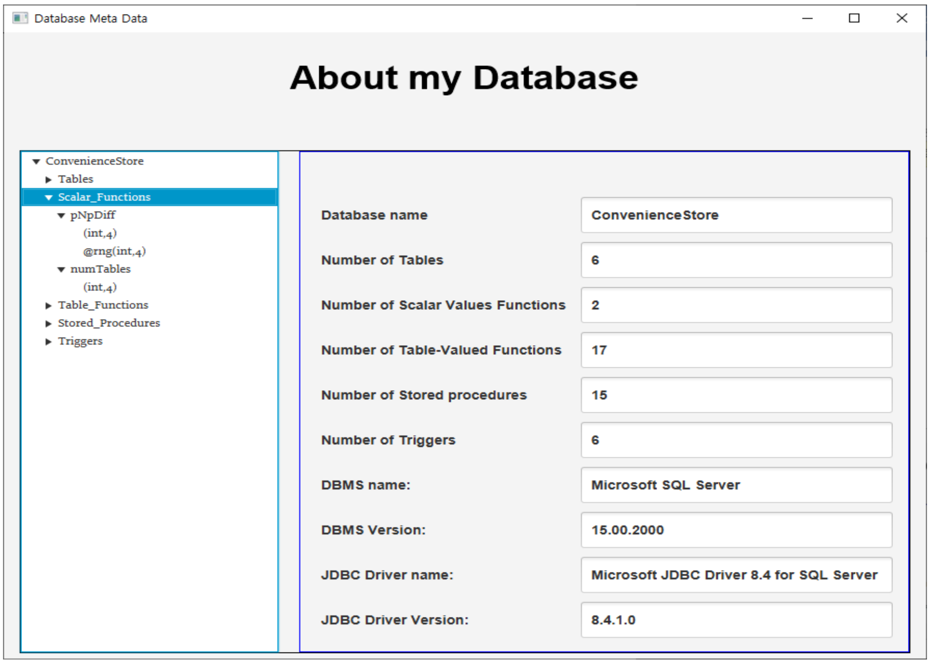
Database name	ConvenienceStore
Number of Tables	6
Number of Scalar Values Functions	2
Number of Table-Valued Functions	17
Number of Stored procedures	15
Number of Triggers	6
DBMS name:	Microsoft SQL Server
DBMS Version:	15.00.2000
JDBC Driver name:	Microsoft JDBC Driver 8.4 for SQL Server
JDBC Driver Version:	8.4.1.0



About my Database

- ConvenienceStore
  - Tables
    - LogTable
      - logdate(datetime,8)
      - logUser(varchar,30)
      - logTableName(varchar,50)
      - logOperation(varchar,30)
      - logRecord(varchar,-1)
    - Branch
      - bId(varchar,5)
      - bName(varchar,15)
      - branchoffice(varchar,30)
    - Stuff
      - sId(varchar,5)
      - sName(varchar,15)
      - shelfLife(int,4)
    - Customer
      - cId(varchar,5)
      - cName(varchar,15)
      - gender(varchar,5)
      - age(int,4)
    - Buying
      - bID(varchar,5)
      - cID(varchar,5)
    - Contain
      - sId(varchar,5)
      - bId(varchar,5)
      - kinds(varchar,30)

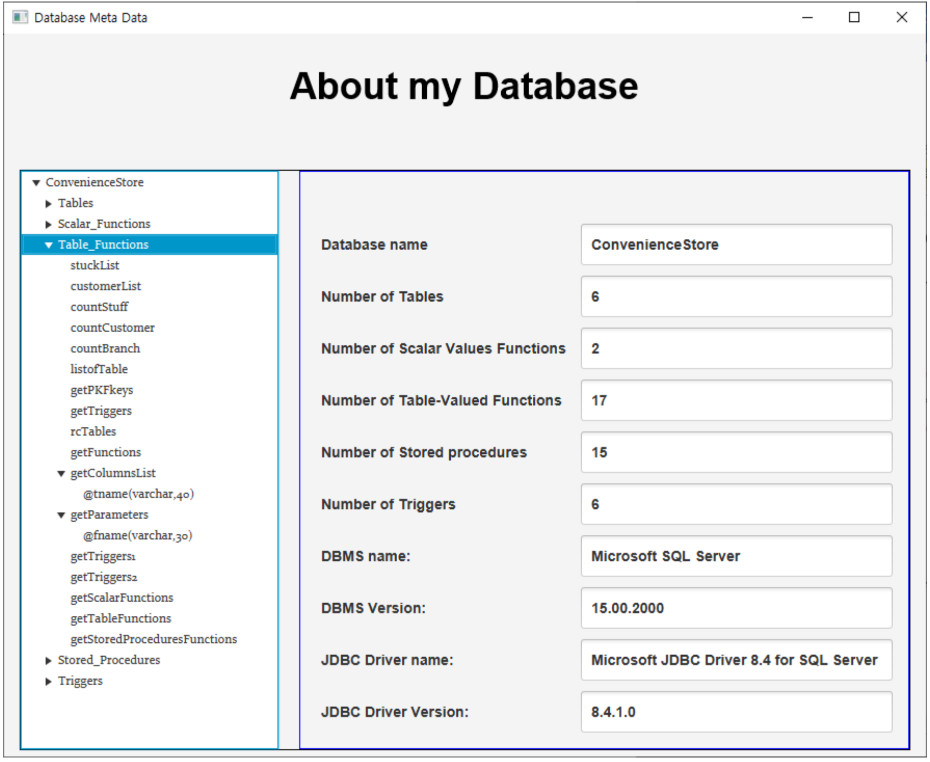
Database name	ConvenienceStore
Number of Tables	6
Number of Scalar Values Functions	2
Number of Table-Valued Functions	17
Number of Stored procedures	15
Number of Triggers	6
DBMS name:	Microsoft SQL Server
DBMS Version:	15.00.2000
JDBC Driver name:	Microsoft JDBC Driver 8.4 for SQL Server
JDBC Driver Version:	8.4.1.0


Database Meta Data

About my Database

- ConvenienceStore
  - Tables
  - Scalar_Functions
    - pNpDiff
      - (int,4)
      - @rng(int,4)
    - numTables
      - (int,4)
  - Table_Functions
  - Stored_Procedures
  - Triggers

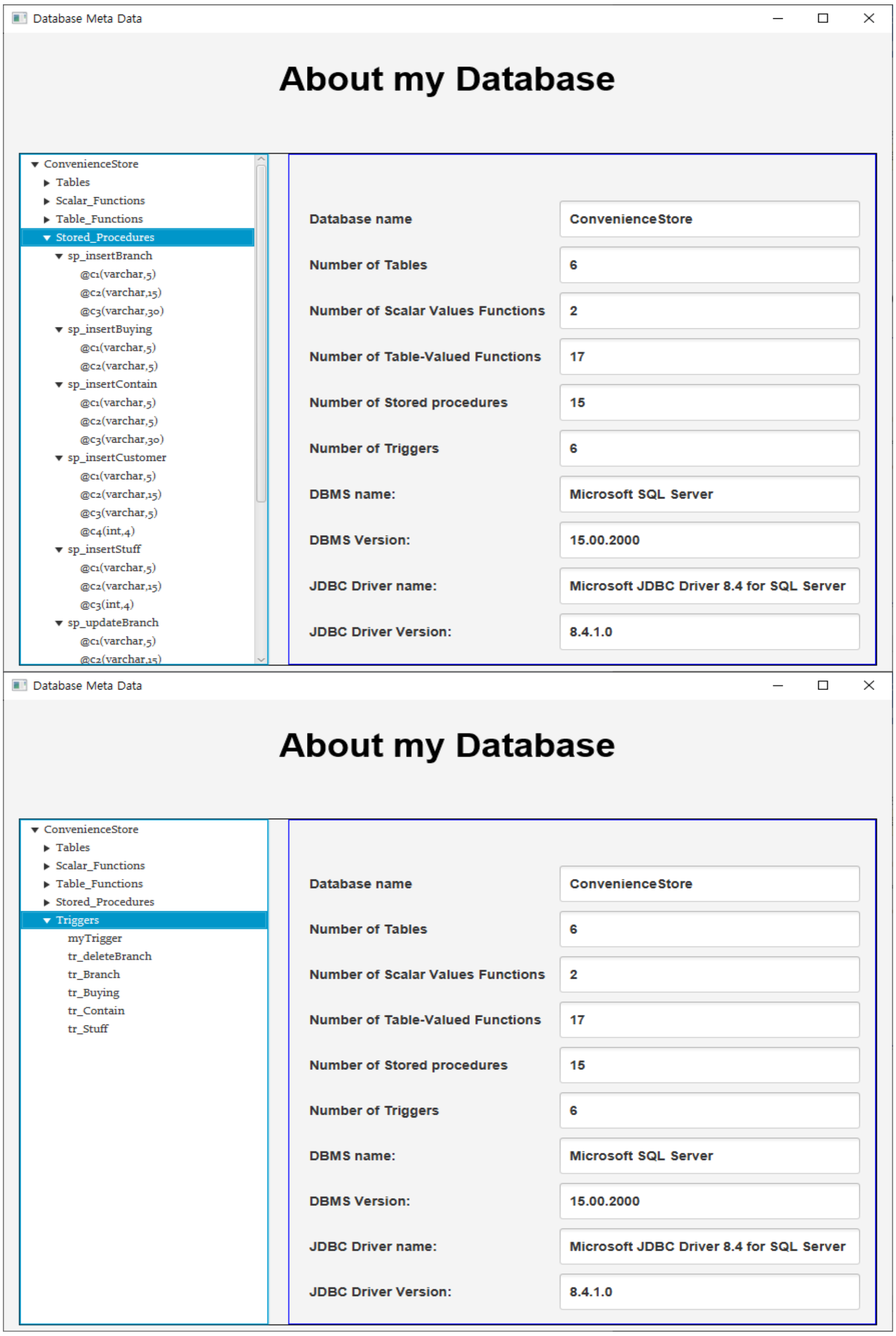
Database name	ConvenienceStore
Number of Tables	6
Number of Scalar Values Functions	2
Number of Table-Valued Functions	17
Number of Stored procedures	15
Number of Triggers	6
DBMS name:	Microsoft SQL Server
DBMS Version:	15.00.2000
JDBC Driver name:	Microsoft JDBC Driver 8.4 for SQL Server
JDBC Driver Version:	8.4.1.0


Database Meta Data

About my Database

- ConvenienceStore
  - Tables
  - Scalar_Functions
  - Table_Functions
    - stuckList
    - customerList
    - countStuff
    - countCustomer
    - countBranch
    - listofTable
    - getPKFkeys
    - getTriggers
    - rcTables
    - getFunctions
    - getColumnsList
      - @tname(varchar,40)
    - getParameters
      - @fname(varchar,30)
    - getTriggers1
    - getTriggers2
    - getScalarFunctions
    - getTableFunctions
    - getStoredProceduresFunctions
  - Stored_Procedures
  - Triggers

Database name	ConvenienceStore
Number of Tables	6
Number of Scalar Values Functions	2
Number of Table-Valued Functions	17
Number of Stored procedures	15
Number of Triggers	6
DBMS name:	Microsoft SQL Server
DBMS Version:	15.00.2000
JDBC Driver name:	Microsoft JDBC Driver 8.4 for SQL Server
JDBC Driver Version:	8.4.1.0



Database Meta Data

About my Database

- ConvenienceStore
  - Tables
  - Scalar_Functions
  - Table_Functions
  - Stored_Procedures
    - sp_insertBranch
      - @c1(varchar,5)
      - @c2(varchar,15)
      - @c3(varchar,30)
    - sp_insertBuying
      - @c1(varchar,5)
      - @c2(varchar,5)
    - sp_insertContain
      - @c1(varchar,5)
      - @c2(varchar,5)
      - @c3(varchar,30)
    - sp_insertCustomer
      - @c1(varchar,5)
      - @c2(varchar,15)
      - @c3(varchar,5)
      - @c4(int,4)
    - sp_insertStuff
      - @c1(varchar,5)
      - @c2(varchar,15)
      - @c3(int,4)
    - sp_updateBranch
      - @c1(varchar,5)
      - @c2(varchar,15)

Database name	ConvenienceStore
Number of Tables	6
Number of Scalar Values Functions	2
Number of Table-Valued Functions	17
Number of Stored procedures	15
Number of Triggers	6
DBMS name:	Microsoft SQL Server
DBMS Version:	15.00.2000
JDBC Driver name:	Microsoft JDBC Driver 8.4 for SQL Server
JDBC Driver Version:	8.4.1.0

Database Meta Data

About my Database

- ConvenienceStore
  - Tables
  - Scalar_Functions
  - Table_Functions
  - Stored_Procedures
  - Triggers
    - myTrigger
    - tr_deleteBranch
    - tr_Branch
    - tr_Buying
    - tr_Contain
    - tr_Stuff

Database name	ConvenienceStore
Number of Tables	6
Number of Scalar Values Functions	2
Number of Table-Valued Functions	17
Number of Stored procedures	15
Number of Triggers	6
DBMS name:	Microsoft SQL Server
DBMS Version:	15.00.2000
JDBC Driver name:	Microsoft JDBC Driver 8.4 for SQL Server
JDBC Driver Version:	8.4.1.0

Fig. 1. Result of Task-2.1

Results and analysis

The corresponding value was derived using the query statement, and the information of the table, function, stored procedure, and trigger was output so that the user could easily know the database .

I felt that

When a user wants to use my database, I think it will be easier to understand the database by looking at this first.

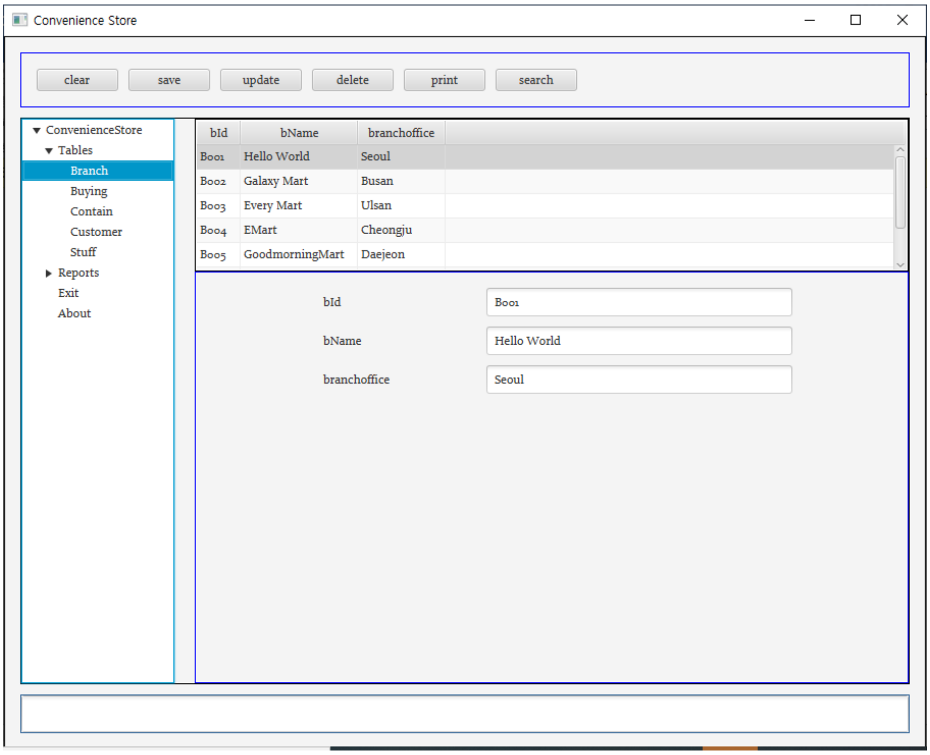
2.2 Navigate (see) records (data) from each table in my database.

This is a task that displays the records on the UI when selecting a table.

Resolution

I applied what the professor told me in the lecture, and modified it to fit the table of my own database to perform the assignment.

Results screen

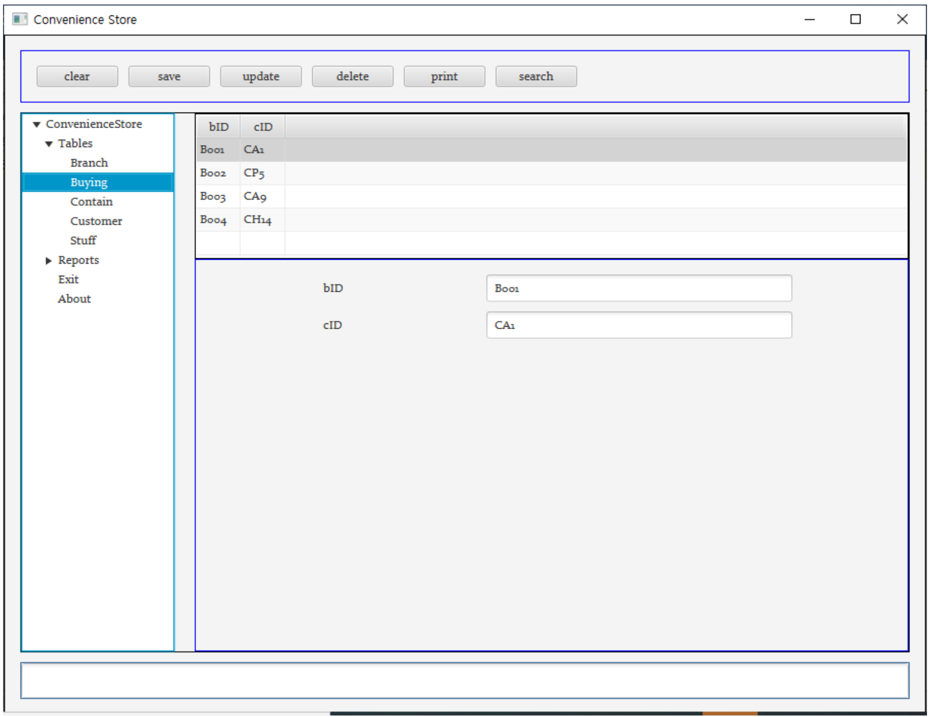


bId	bName	branchoffice
B001	Hello World	Seoul
B002	Galaxy Mart	Busan
B003	Every Mart	Ulsan
B004	EMart	Cheongju
B005	GoodmorningMart	Daejeon

bId: B001

bName: Hello World

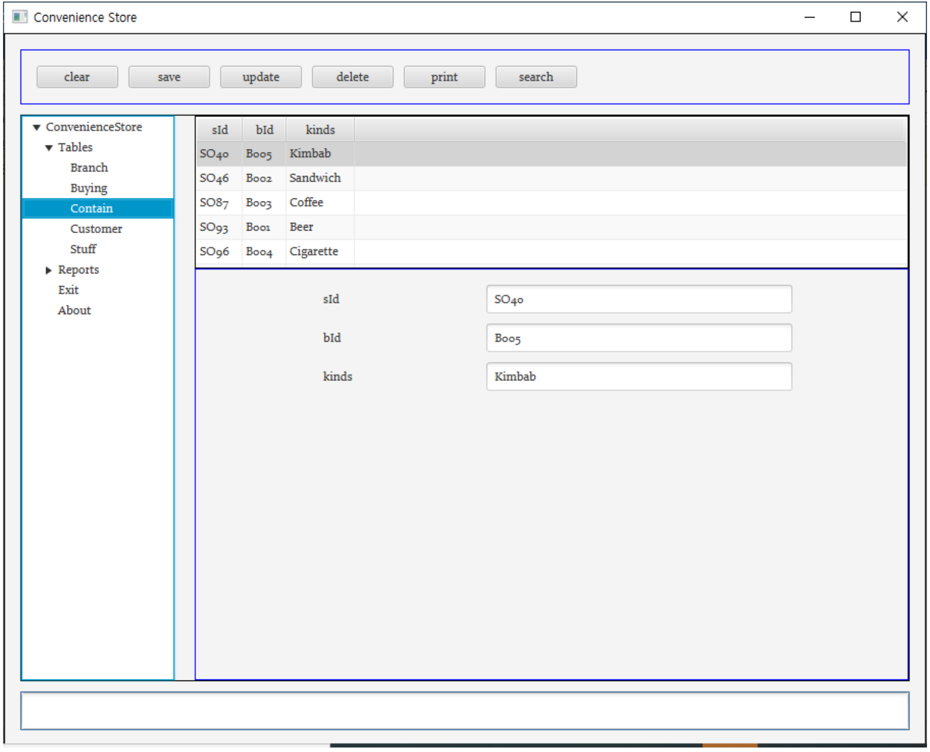
branchoffice: Seoul



bID	cID
B001	CA1
B002	CP5
B003	CA9
B004	CH14

bID B001

cID CA1

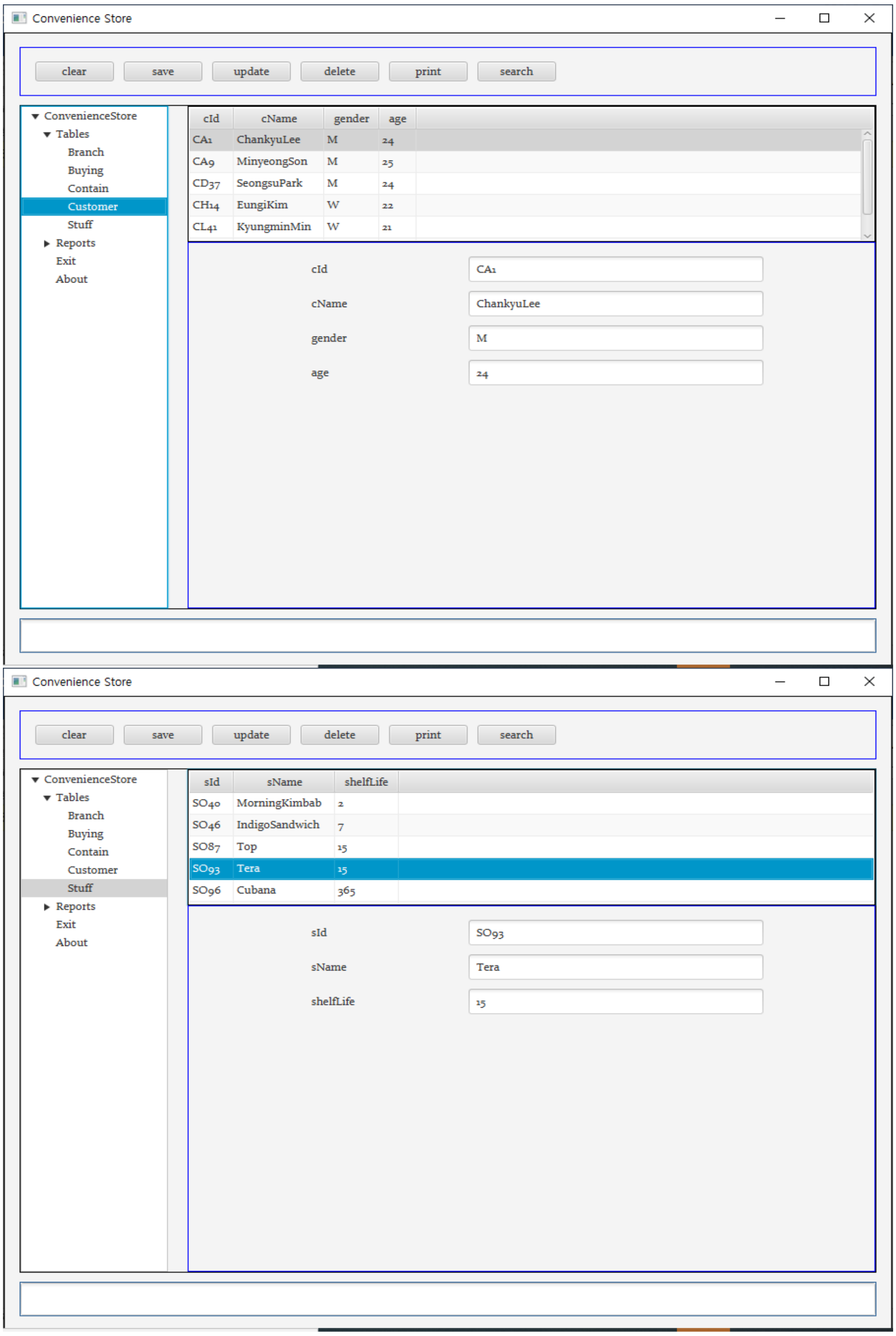


sId	bId	kinds
SO40	B005	Kimbab
SO46	B002	Sandwich
SO87	B003	Coffee
SO93	B001	Beer
SO96	B004	Cigarette

sId SO40

bId B005

kinds Kimbab



cId	cName	gender	age
CA1	ChankyuLee	M	24
CA9	MinyeongSon	M	25
CD37	SeongsuPark	M	24
CH14	EungiKim	W	22
CL41	KyungminMin	W	21

cId: CA1

cName: ChankyuLee

gender: M

age: 24

sId	sName	shelfLife
SO40	MorningKimbab	2
SO46	IndigoSandwich	7
SO87	Top	15
SO93	Tera	15
SO96	Cubana	365

sId: SO93

sName: Tera

shelfLife: 15

Fig. 2. Result of Task-2.2

Results and analysis

When the table is selected, the corresponding records are displayed normally, and when Insert, delete or update are performed, they are immediately applied and output normally.

I felt that

It was convenient and easy to see at a glance because I could easily check the records of the table I created through the GUI.

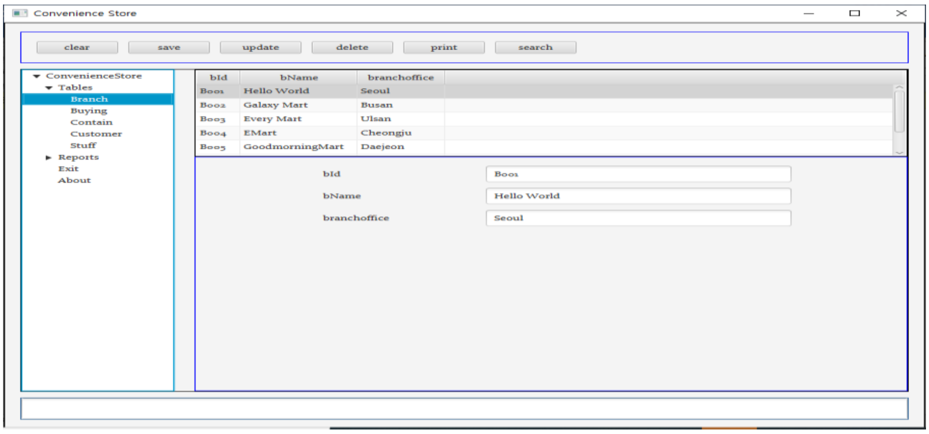
2.3 Run functions to insert, update and delete records for each table in my database.

This is a task of easily inserting, updating and deleting the database using GUI.

Resolution

I learned the basic method through the professor's lecture and applied it to my database and applied it with different properties for each table.

Results screen



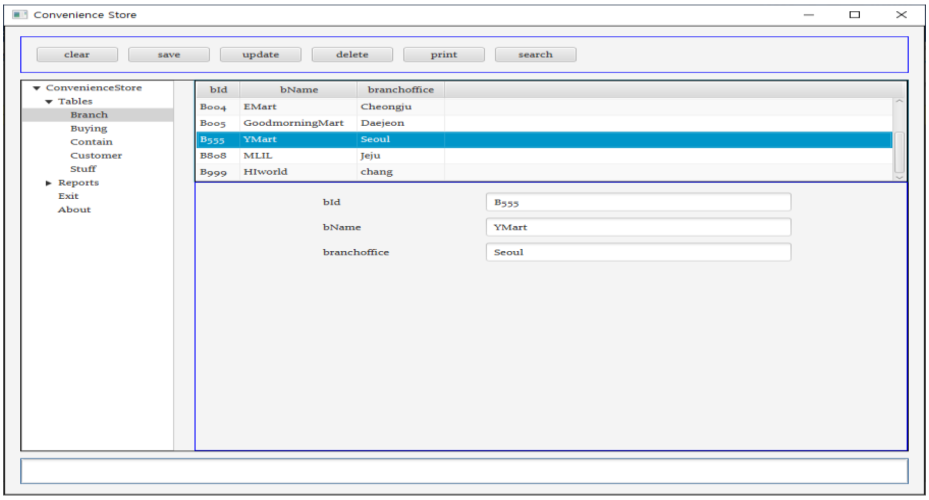
bId	bName	branchoffice
B001	Hello World	Seoul
B002	Galaxy Mart	Busan
B003	Every Mart	Ulsan
B004	EMart	Cheongju
B005	GoodmorningMart	Daejeon

bId: B001

bName: Hello World

branchoffice: Seoul

Fig. 3. (BranchTable)Before Inserting (bId=B555, bName=YMart, branchoffice=Seoul)



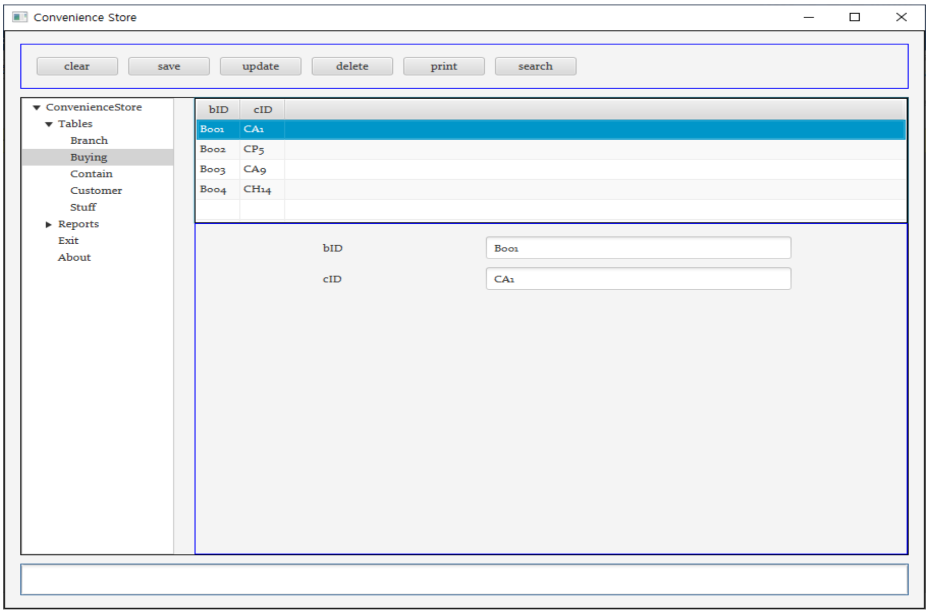
bId	bName	branchoffice
B004	EMart	Cheongju
B005	GoodmorningMart	Daejeon
B555	YMart	Seoul
B808	MLIL	Jeju
B999	HIworld	chang

bId: B555

bName: YMart

branchoffice: Seoul

Fig. 4. (BranchTable)After Inserting (bId=B555, bName=YMart, branchoffice=Seoul)

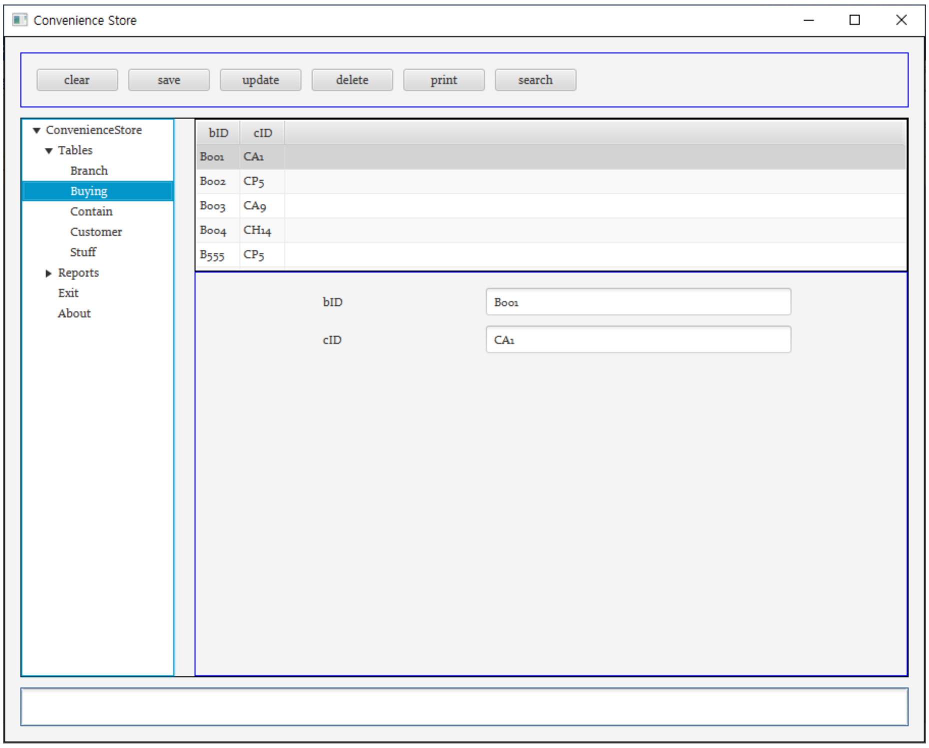


bID	cID
B001	CA1
B002	CP5
B003	CA9
B004	CH14

bID B001

cID CA1

Fig. 5. (BuyingTable)Before Inserting (bId=B555, Cid=CP5)

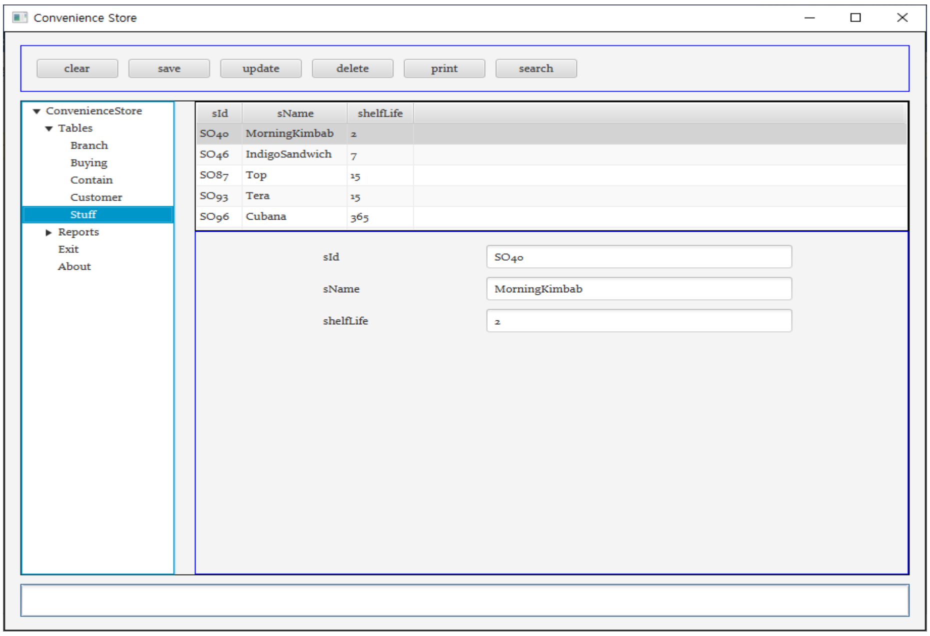


bID	cID
B001	CA1
B002	CP5
B003	CA9
B004	CH14
B555	CP5

bID B001

cID CA1

Fig. 6. (BuyingTable)After Inserting (bId=B555, Cid=CP5)



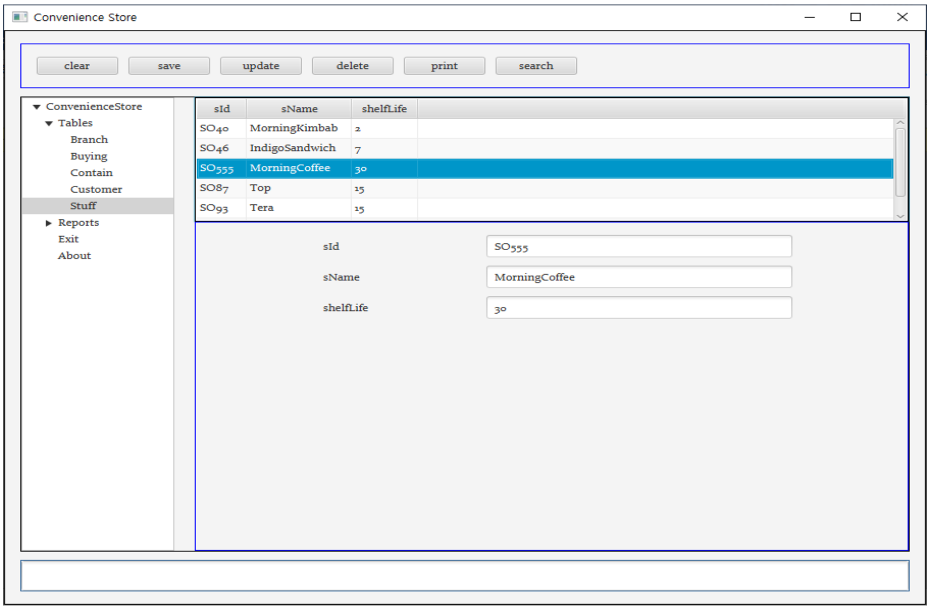
sId	sName	shelfLife
SO40	MorningKimbab	2
SO46	IndigoSandwich	7
SO87	Top	15
SO93	Tera	15
SO96	Cubana	365

sId SO40

sName MorningKimbab

shelfLife 2

Fig. 7. (StuffTable)Before Inserting (sId=SO555, sName=MorningCoffee, shelfLife=30)



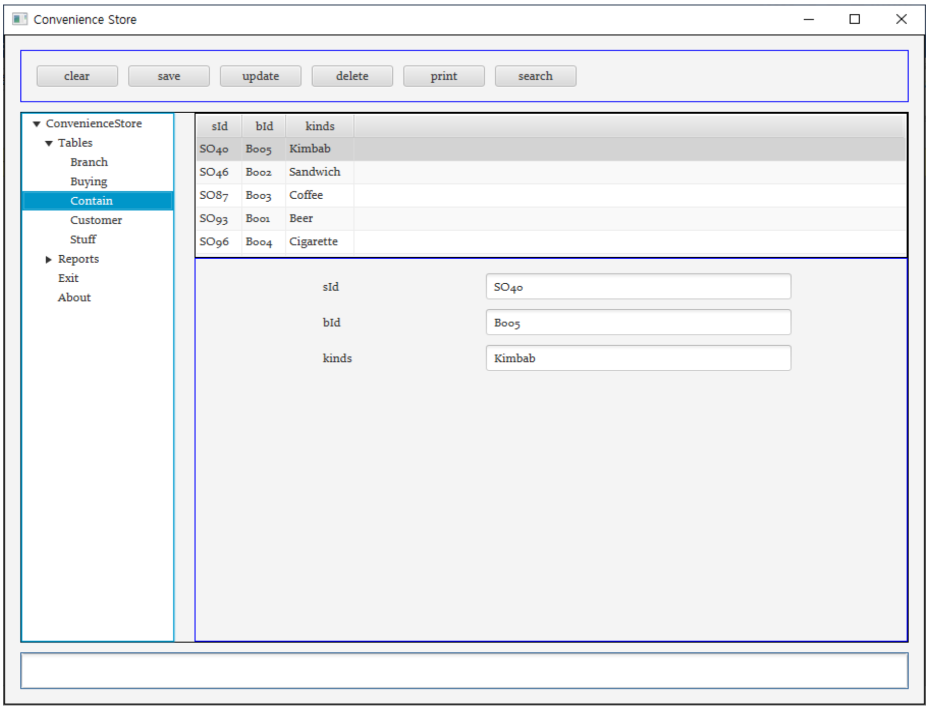
sId	sName	shelfLife
SO40	MorningKimbab	2
SO46	IndigoSandwich	7
SO555	MorningCoffee	30
SO87	Top	15
SO93	Tera	15

sId SO555

sName MorningCoffee

shelfLife 30

Fig. 8. (Stuff Table) After Inserting (sId=SO555, sName=MorningCoffee, shelfLife=30)



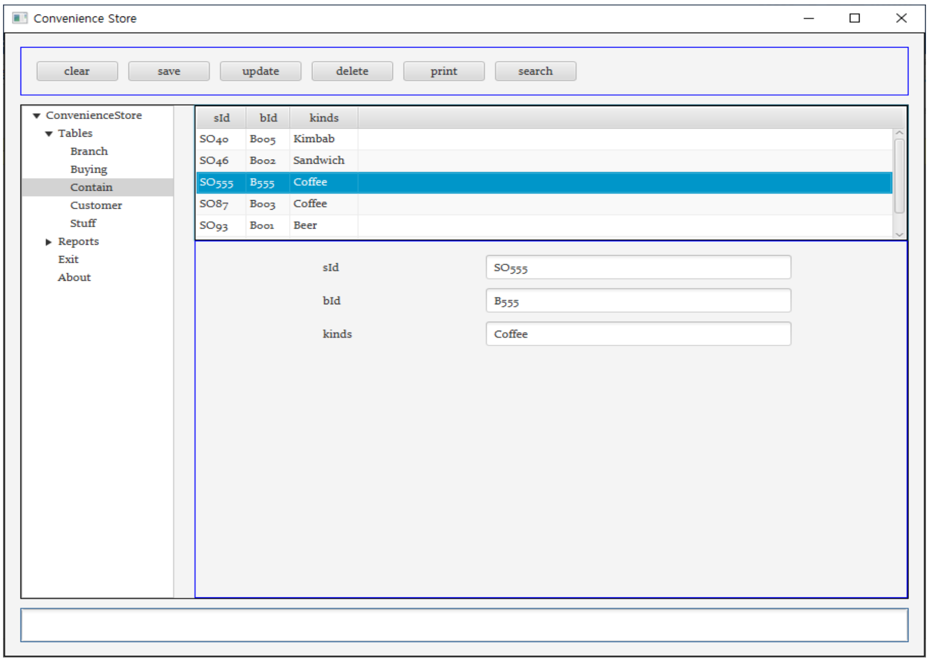
sId	bId	kinds
SO40	B005	Kimbab
SO46	B002	Sandwich
SO87	B003	Coffee
SO93	B001	Beer
SO96	B004	Cigarette

sId SO40

bId B005

kinds Kimbab

Fig. 9. (ContainTable) Before Inserting (sId=SO555, bId=B555, kinds=Coffee)



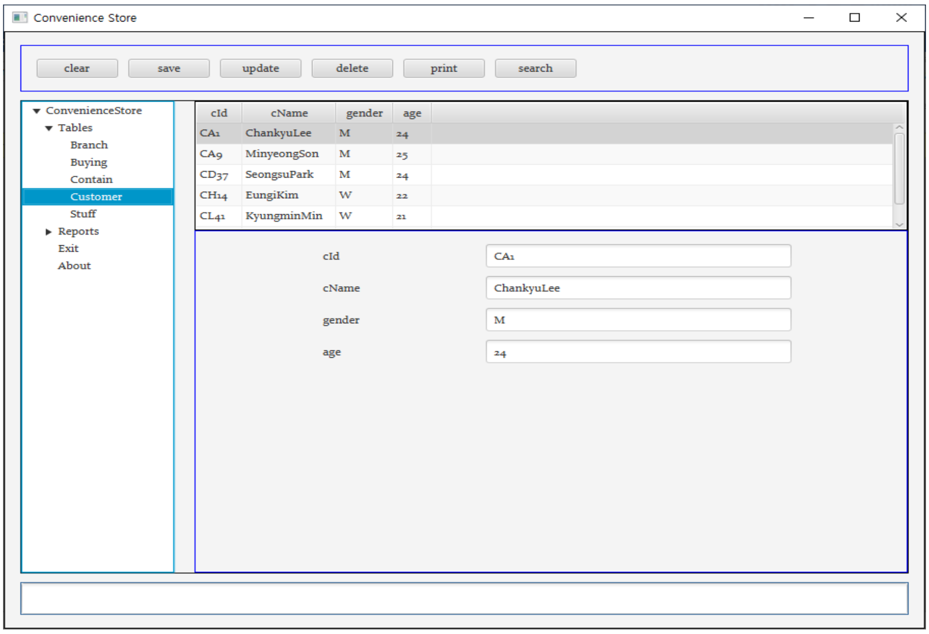
sId	bId	kinds
SO40	B005	Kimbab
SO46	B002	Sandwich
SO555	B555	Coffee
SO87	B003	Coffee
SO93	B001	Beer

sId SO555

bId B555

kinds Coffee

Fig. 10. (ContainTable) After Inserting (sId=SO555, bId=B555, kinds=Coffee)



cId	cName	gender	age
CA1	ChankyuLee	M	24
CA9	MinyeongSon	M	25
CD37	SeongsuPark	M	24
CH14	EungiKim	W	22
CL41	KyungminMin	W	21

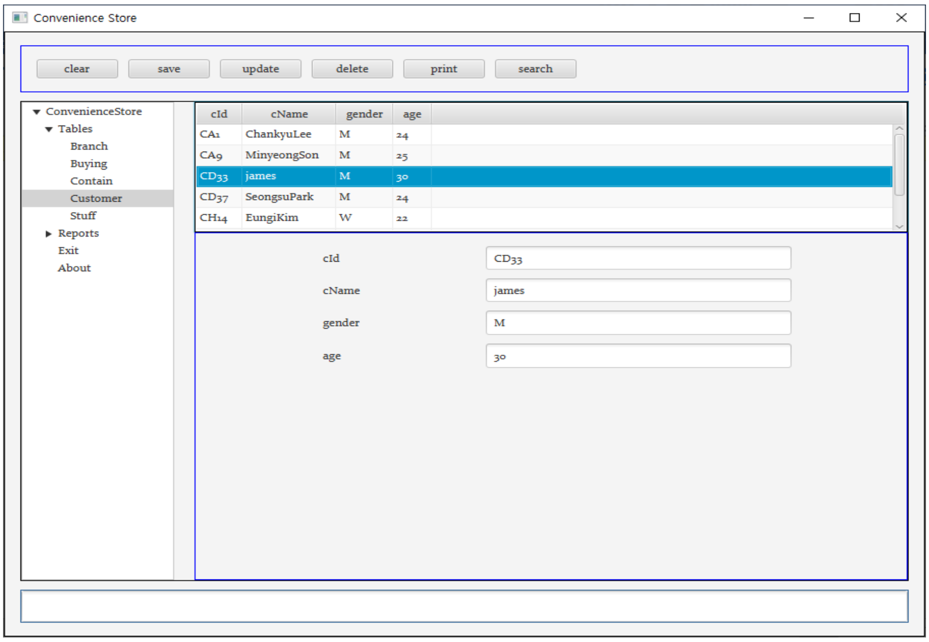
cId: CA1

cName: ChankyuLee

gender: M

age: 24

Fig. 11. (CustomerTable) Before Inserting (cId=CD33, cName=james, gender=m, age=30)



cId	cName	gender	age
CA1	ChankyuLee	M	24
CA9	MinyeongSon	M	25
CD33	james	M	30
CD37	SeongsuPark	M	24
CH14	EungiKim	W	22

cId: CD33

cName: james

gender: M

age: 30

Fig. 12. (CustomerTable) After Inserting (cId=CD33, cName=james, gender=m, age=30)

Results and analysis

I checked whether it words normally by performing the functions of insert, update, and delete. The result screen displays it. It has been confirmed that the functions performed through the GUI are applied normally to the database.

I felt that

Creating such an application was very convenient because even people who do not know much about the database could easily use it and it was more visualized.

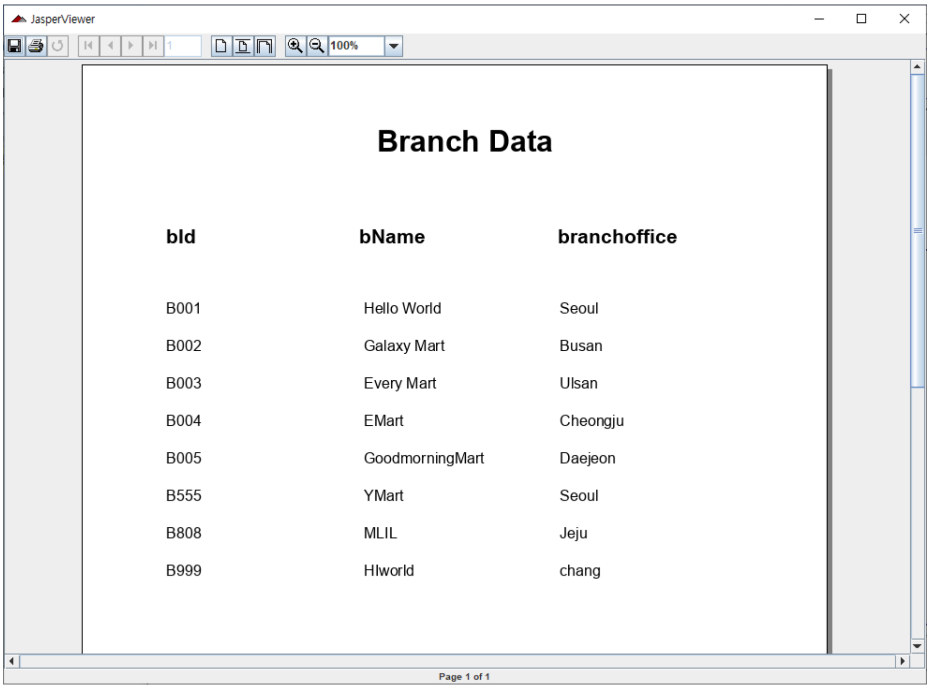
2.4 Print reports (data from different tables)

It is a problem of printing the records of the corresponding table using JasperSoftStudio.

Resolution

Reports in jrxml format were created and designed using JasperSoftStudio. After that, when the "print" button is clicked in connection with the database, the record of the corresponding table is displayed..

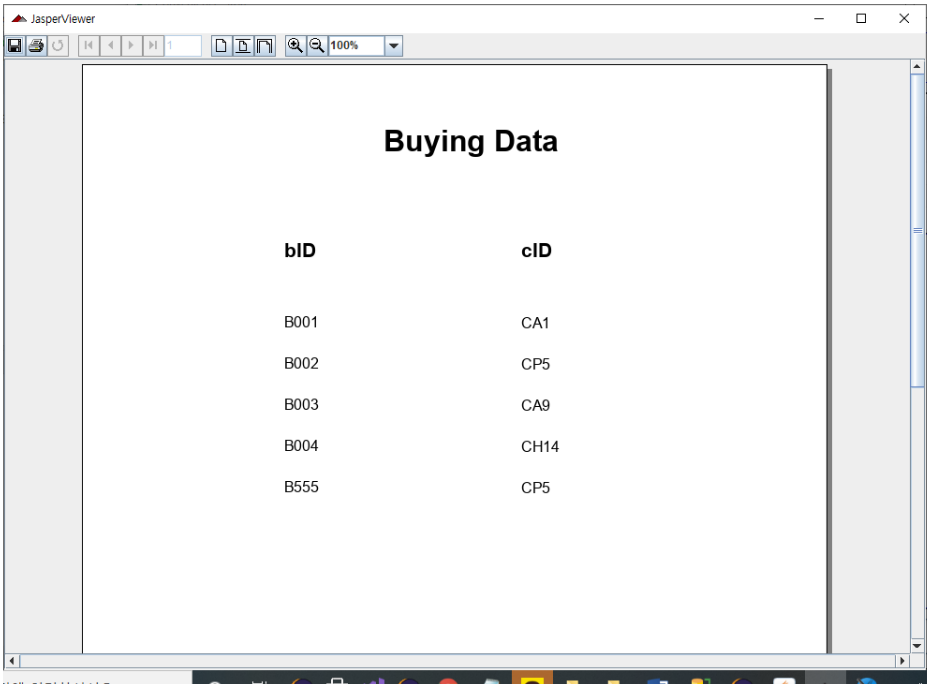
Results screen



Branch Data

bId	bName	branchoffice
B001	Hello World	Seoul
B002	Galaxy Mart	Busan
B003	Every Mart	Ulsan
B004	EMart	Cheongju
B005	GoodmorningMart	Daejeon
B555	YMart	Seoul
B808	MLIL	Jeju
B999	HIworld	chang

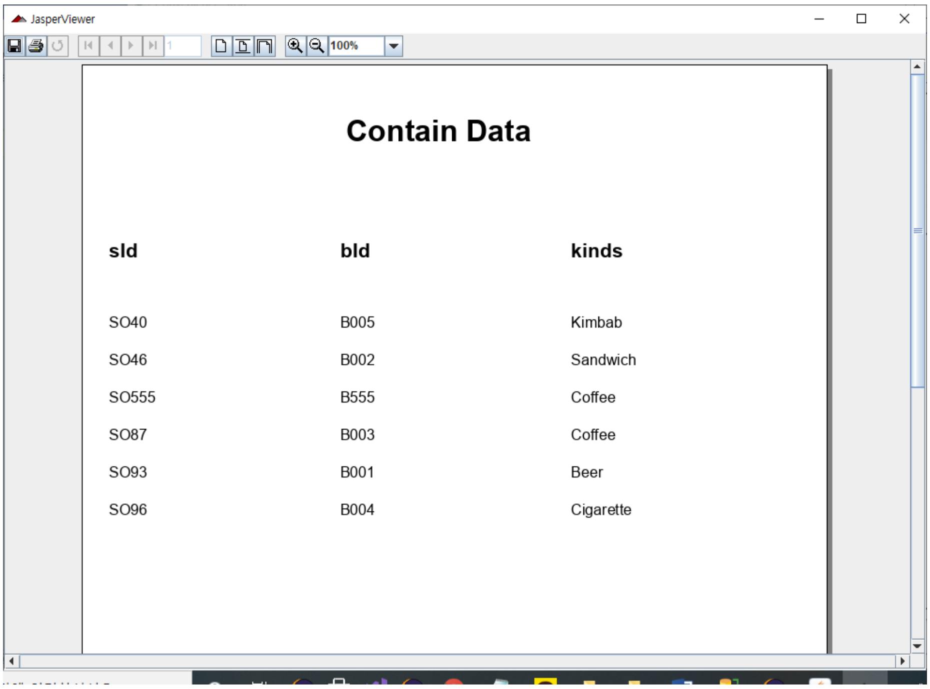
Fig. 13. (BranchTable) print reports



Buying Data

bID	cID
B001	CA1
B002	CP5
B003	CA9
B004	CH14
B555	CP5

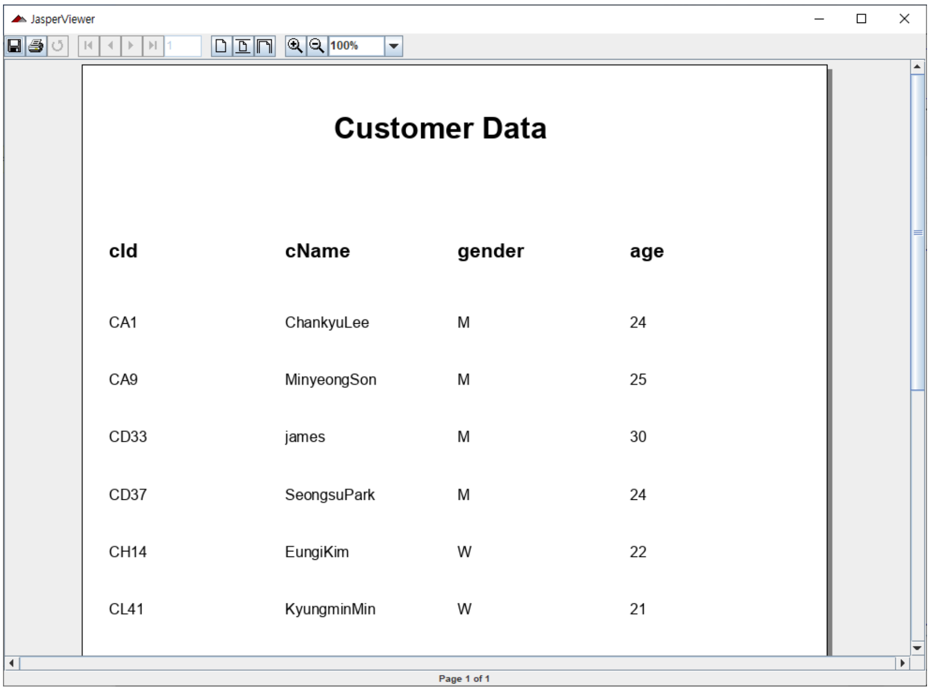
Fig. 14. (BuyingTable) print reports



Contain Data

sId	bId	kinds
SO40	B005	Kimbab
SO46	B002	Sandwich
SO555	B555	Coffee
SO87	B003	Coffee
SO93	B001	Beer
SO96	B004	Cigarette

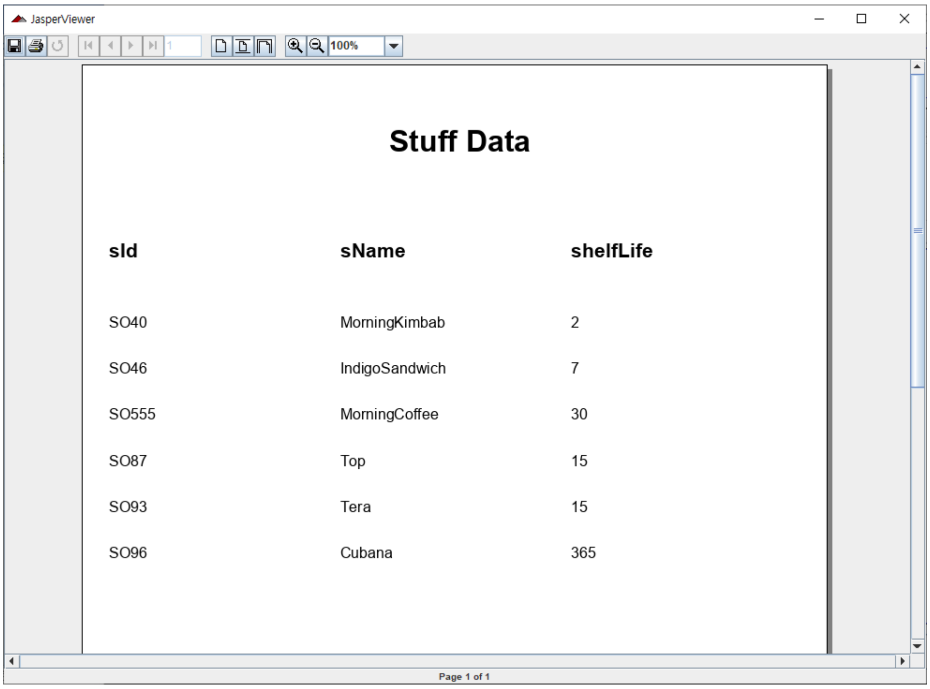
Fig. 15. (ContainTable) print reports



Customer Data

cId	cName	gender	age
CA1	ChankyuLee	M	24
CA9	MinyeongSon	M	25
CD33	james	M	30
CD37	SeongsuPark	M	24
CH14	EungiKim	W	22
CL41	KyungminMin	W	21

Fig. 16. (CustomerTable) print reports



Stuff Data

sId	sName	shelfLife
SO40	MorningKimbab	2
SO46	IndigoSandwich	7
SO555	MorningCoffee	30
SO87	Top	15
SO93	Tera	15
SO96	Cubana	365

Fig. 17. (StuffTable) print reports

Results and analysis

The records of the table corresponding to the reports I designed in JasperSotfStudio were displayed normally.

I felt that

This is the first time I knew a platform like JasperSoftStudio. I was curious about how the UI that represents the output on the web is usually made, but this was solved by carrying out this task. It seems to be very convenient when I want to physically use a database that contains a lot of information.

3 Conclusion

I felt that

In fact, being a database is full of vast amounts and difficult to understand information. However, I thought it was really convenient to be able to check this neatly through the GUI, add, modify, and update data, and know various information. I also thought this would be a lot to be used on other projects in the future.

Difficult point

There was a difficulty in implementing the fillgrid function and the addNodestoTree function in the MetaData.java file. It took a lot of time to think to properly express the contents of parameters and columns for each table, function, trigger, and stored procedure and in the end I followed the professor’s lecture to understand the overall flow and wrote the code to fit the logic.

Suggestions

None.

References