

알 기 쉽 게 해 설 한

J A V A

9th edition

18장 JDBC와 응용 프로젝트

Section 1 데이터 저장 : 파일 시스템과 데이터베이스 시스템

Section 2 MySQL 데이터 베이스 설치

Section 3 MySQL 서버의 실행과 SQL 사용

Section 4 JDBC

Section 5 JDBC 응용 프로젝트



• 학습 목표

9th edition

- 파일 시스템과 데이터베이스 시스템의 개념과 차이에 관해 학습한다.
- MySQL 데이터베이스 시스템의 설치에 관해 학습한다.
- 데이터베이스 표준 질의어인 SQL에 관해 학습한다.
- 자바 응용 프로그램과 데이터베이스를 연동시키는 JDBC에 관해 학습한다.
- JDBC를 이용한 자바 응용 프로젝트에 관해 학습한다..



1.1 파일 시스템

● 파일 시스템

- 작고 간단한 응용에 가능(13장)
- 순차적인 데이터를 공란이나 쉼표 등으로 구분하여 저장
- 대부분의 응용 프로그램에서 효율적인 문제로 파일 시스템 사용 안함

1.1 파일 시스템

● 파일 시스템 문제점

- 데이터의 종속성 문제뿐만 아니라 데이터의 무결성(data integrity)(데이터의 정확성, 일관성, 유효성을 유지하는 것)과 같은 문제점이 발생

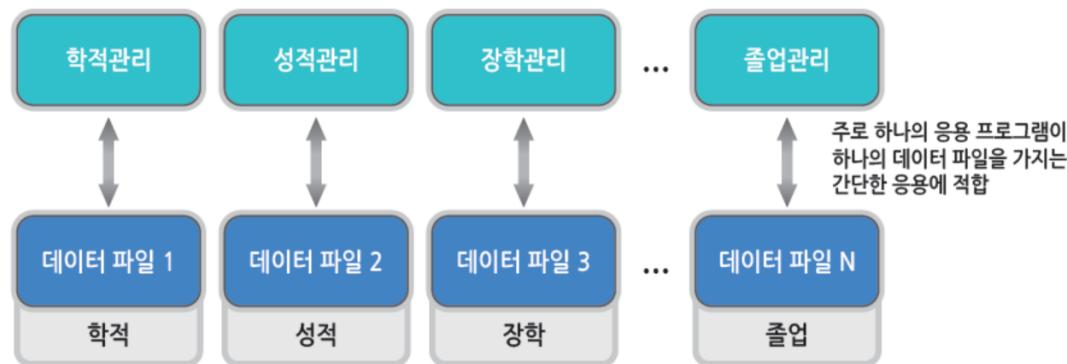


그림 18-1 파일 시스템

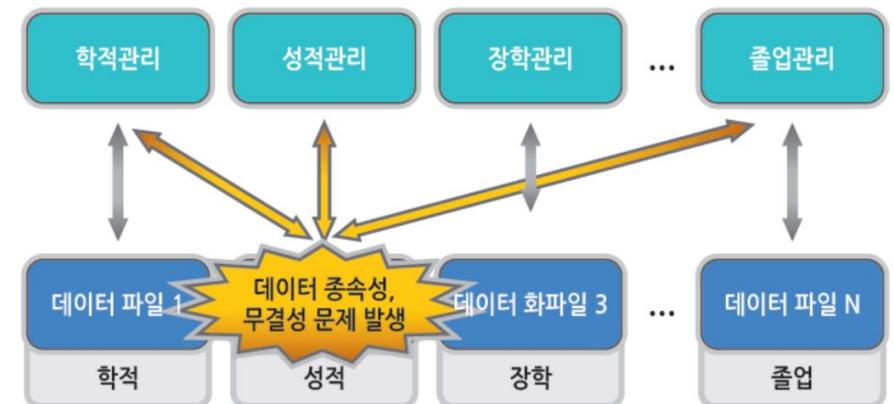
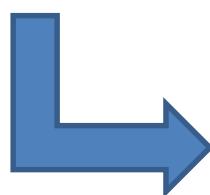


그림 18-2 파일 시스템에서 데이터 파일의 공유

성적 데이터를 사로 다른
응용 프로그램에서 공유하는 경우
많은 문제점이 발생

1.2 데이터베이스 시스템

● 데이터베이스 시스템

- 파일 시스템의 문제점을 해결
- 데이터와 데이터를 사용하는 응용 프로그램과를 분리해주는 DBMS DataBase Management System

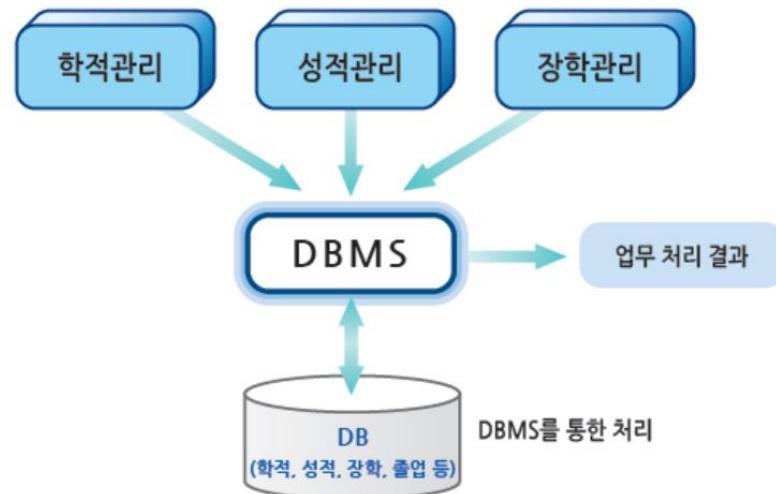


그림 18-3 데이터베이스 시스템

1.2 데이터베이스 시스템

● 데이터베이스 시스템의 사용

- 개발자는 데이터베이스 외부 스키마를 기준으로 프로그래밍
- 실제 외부 스키마와 데이터의 맵핑은 DBMS가 담당
- DBMS를 관리하는 관리자 DBA : Data Base Administrator

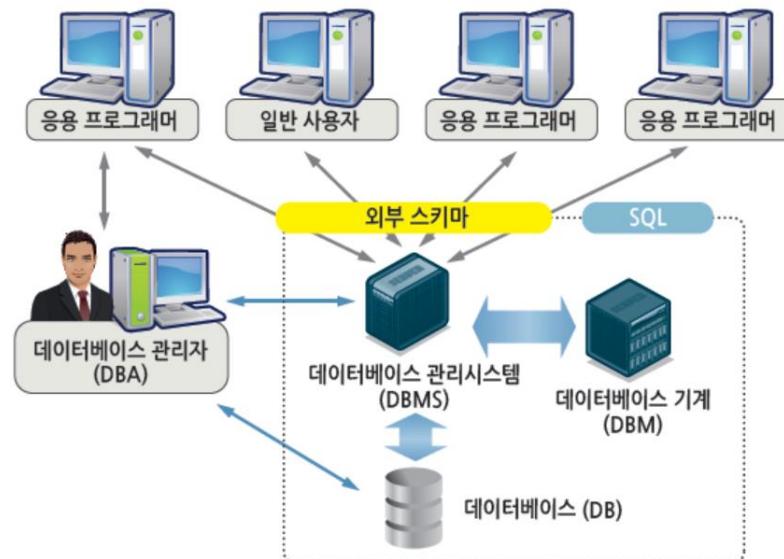


그림 18-4 데이터베이스 시스템의 구조와 사용자

1.2 데이터베이스 시스템

● 파일 시스템 vs 데이터베이스 시스템

표 18-1 파일 시스템과 데이터베이스 시스템의 장단점

	파일 시스템	데이터베이스 시스템
장점	<ul style="list-style-type: none">처리속도가 DB에 비해 빠름구현이 간단비용이 저렴	<ul style="list-style-type: none">자료의 독립성 보장데이터 무결성 보장개발 생산성 높음(표준 SQL, 개발기간 절감, 운영비용 절감)
단점	<ul style="list-style-type: none">자료의 중복 발생데이터 무결성 보장의 어려움개발 생산성 낮음	<ul style="list-style-type: none">시스템의 부하 및 복잡성 증대별도의 관리 이력 필요추가 도입비용 발생



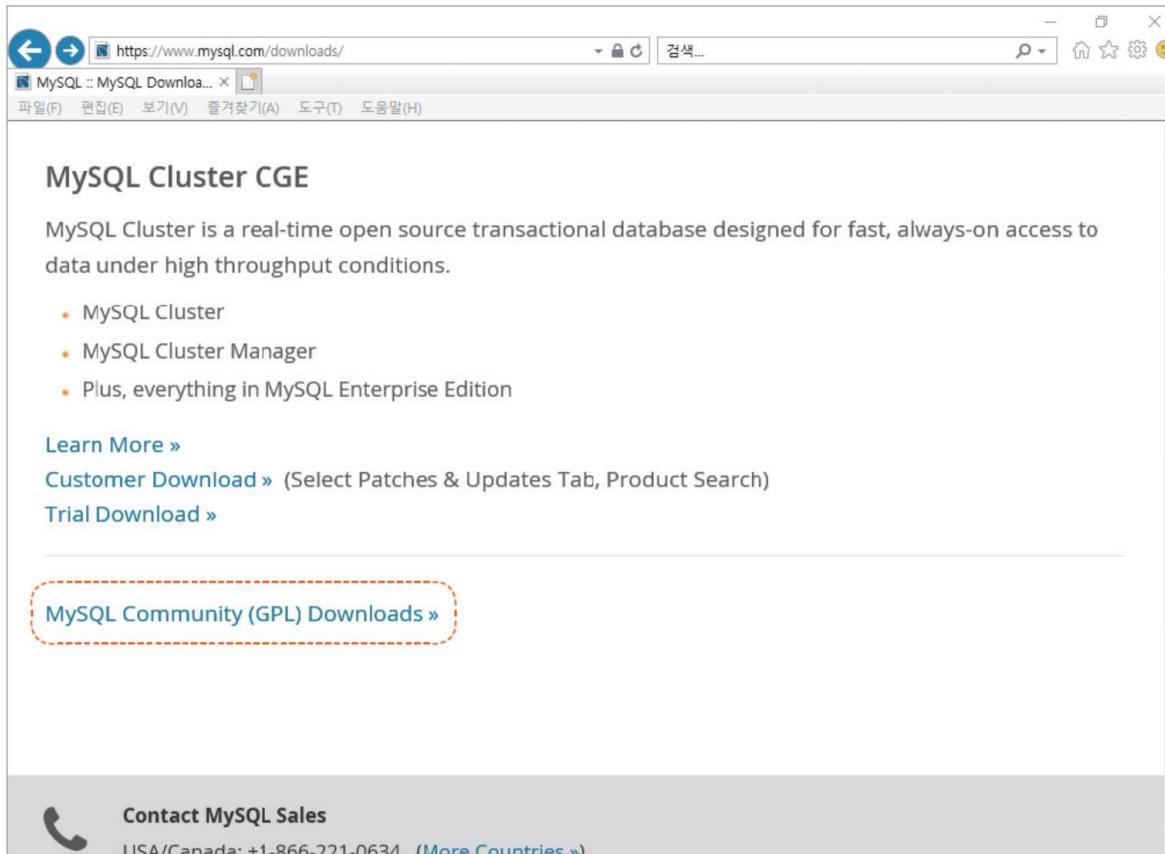
● MySQL 데이터베이스

- 공개 소스^{open source}로서 상업용이 아닌 경우 무료로 사용
- 1995년 5월 스웨덴의 MySQL AB사에 의해 개발
- 표준 데이터베이스 질의 언어인 SQL^{Structured Query Language}을 사용
- 관계형 데이터베이스 관리 시스템^{RDBMS : Relational Data Base Management System}

2.1 MySQL 다운로드

- www.mysql.com/downloads에서 다운로드

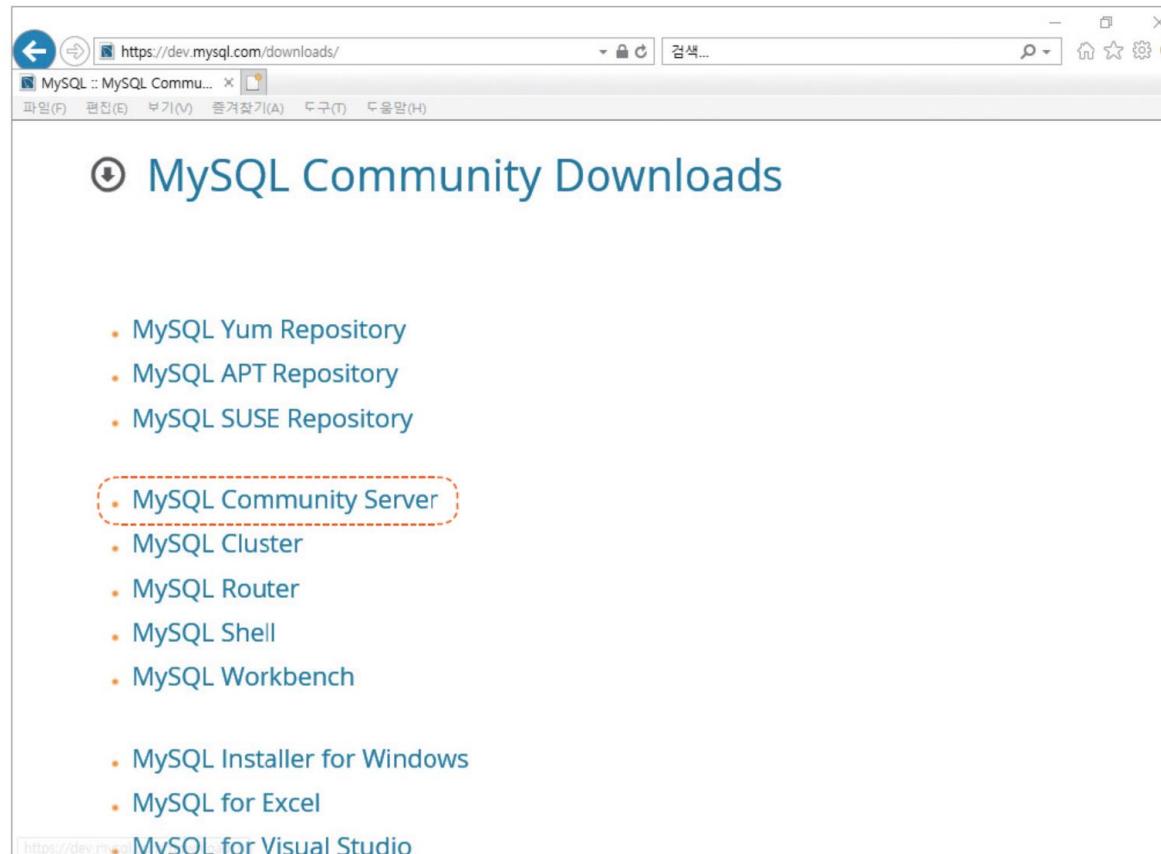
«Step 1 홈페이지 아래부분에 “MySQL Community (GPL) Downloads » 선택



The screenshot shows a web browser window displaying the MySQL Cluster CGE page at <https://www.mysql.com/downloads/>. The page content includes a brief description of MySQL Cluster, a bulleted list of features, and links for 'Learn More', 'Customer Download', and 'Trial Download'. At the bottom, there is a contact section for MySQL Sales. A red dashed box highlights the 'MySQL Community (GPL) Downloads »' link, which is located in the footer area of the page.

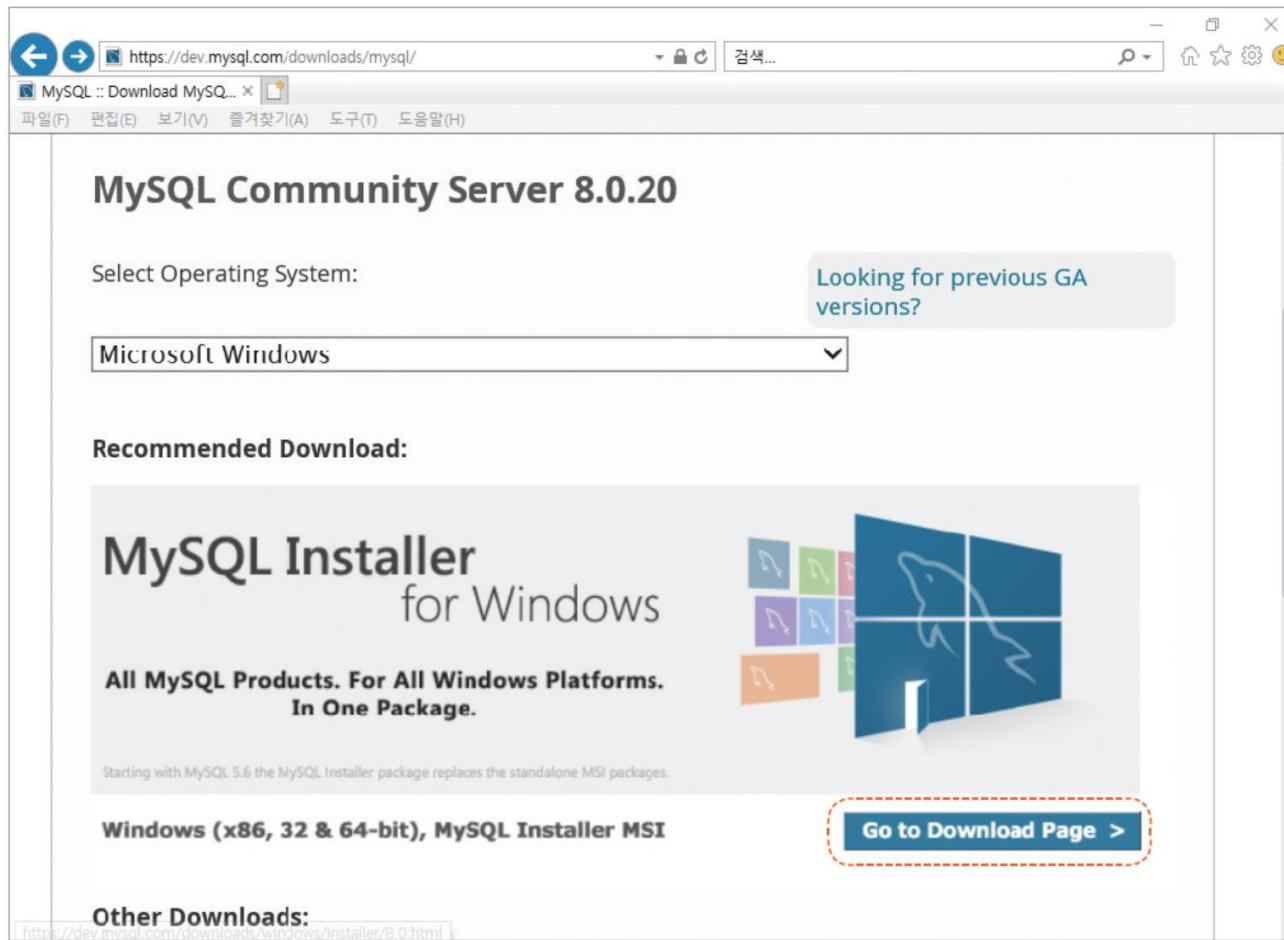
2.1 MySQL 다운로드

«Step 2 다음 화면에서 “MySQL Community Server” 선택



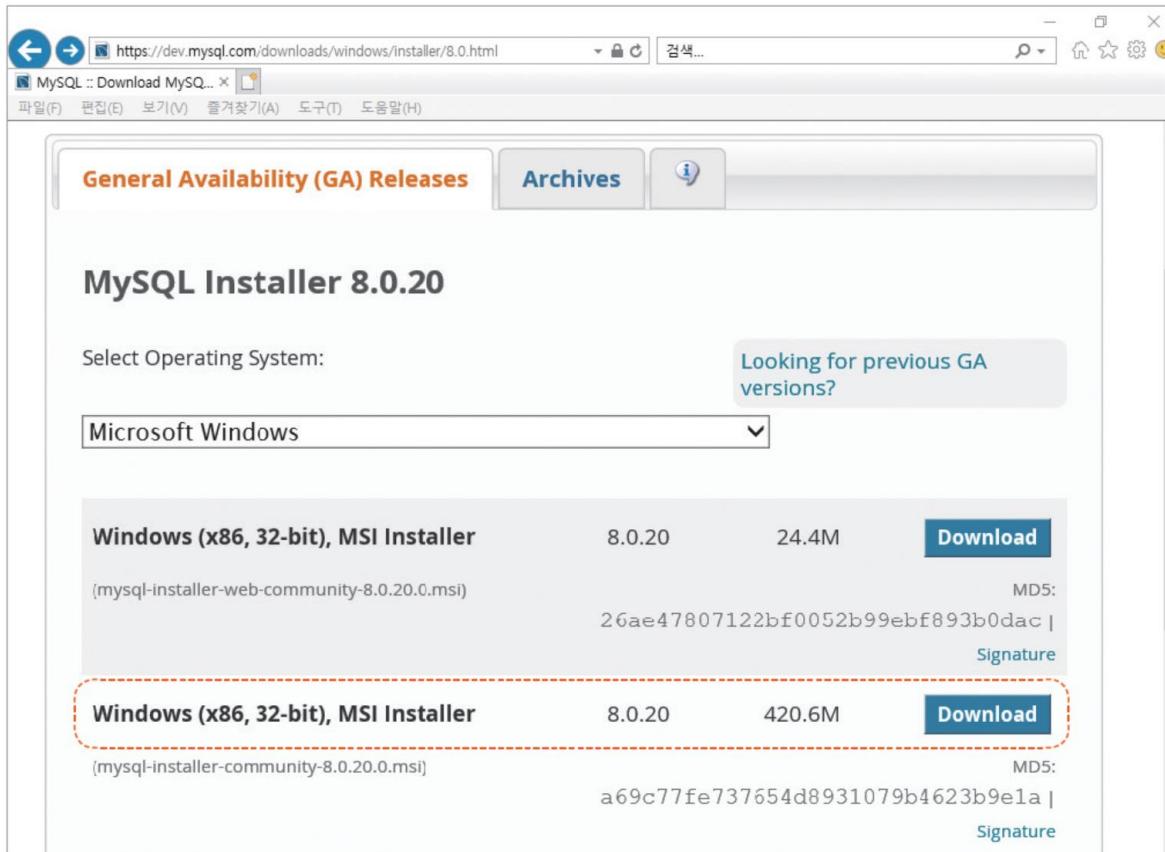
2.1 MySQL 다운로드

«Step 3 다음 화면에서 “Go to Download Page >” 선택



2.1 MySQL 다운로드

«Step 4 “mysql-installer-community-8.0.20.0.msi”을 선택



The screenshot shows a web browser displaying the MySQL download page for Windows. The URL in the address bar is <https://dev.mysql.com/downloads/windows/installer/8.0.html>. The page title is "MySQL :: Download MySQL". A navigation bar at the top includes "General Availability (GA) Releases" (which is highlighted in orange), "Archives", and a help icon.

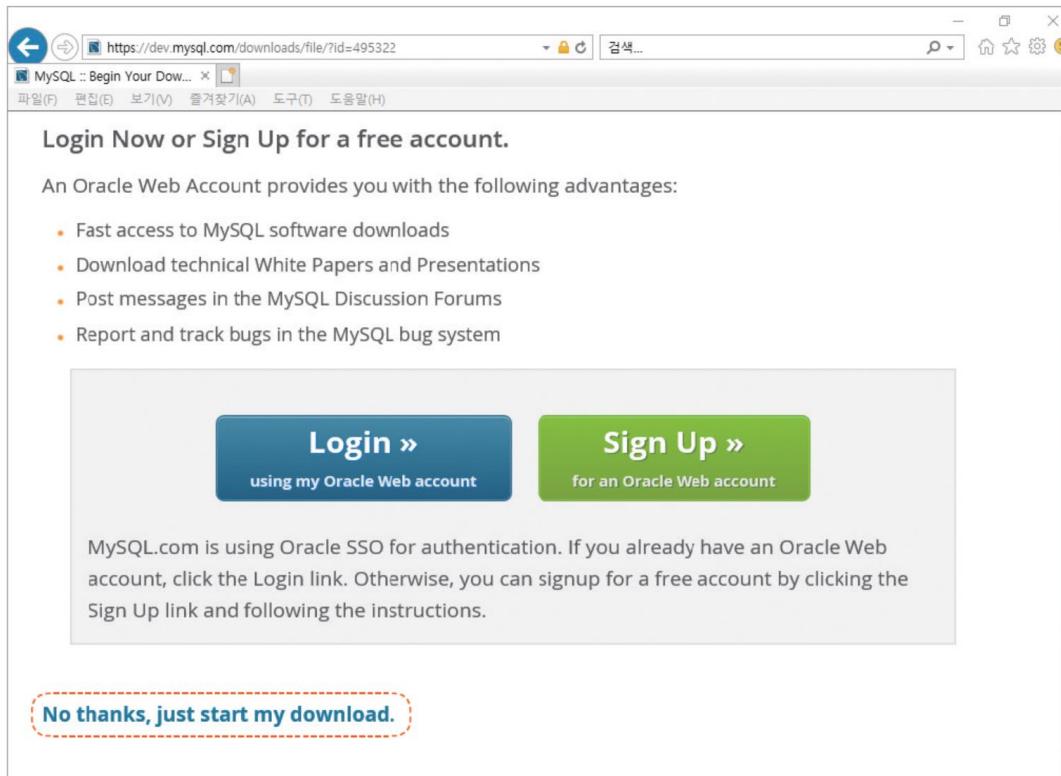
The main content area is titled "MySQL Installer 8.0.20". It asks "Select Operating System:" and has a dropdown menu set to "Microsoft Windows". To the right, there is a link "Looking for previous GA versions?".

Two download options are listed:

- Windows (x86, 32-bit), MSI Installer**: Version 8.0.20, file size 24.4M. The download button is labeled "Download". Below it, the MD5 hash is listed as 26ae47807122bf0052b99ebf893b0dac | Signature.
- Windows (x86, 32-bit), MSI Installer**: Version 8.0.20, file size 420.6M. This option is highlighted with a red dashed box around its row. The download button is labeled "Download". Below it, the MD5 hash is listed as a69c77fe737654d8931079b4623b9e1a | Signature.

2.1 MySQL 다운로드

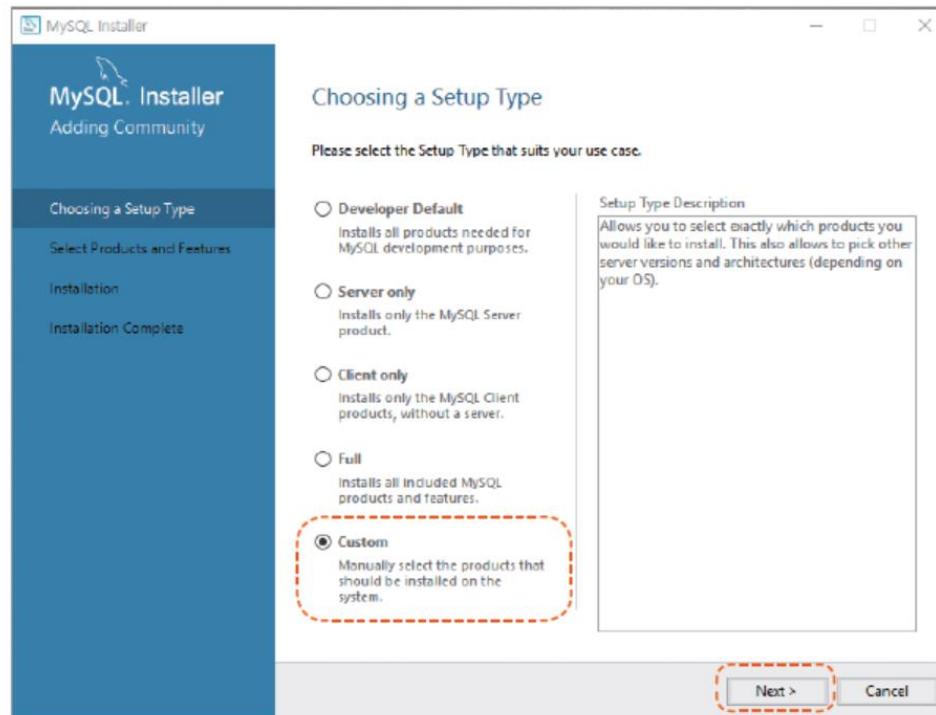
«Step 5 다음 화면에서 로그인 화면 아래의 “No thanks, just start my download.”를 선택하여 파일을 다운로드



«Step 6 컴퓨터의 특정 폴더에 파일을 저장

(1) 다운 받은 파일을 실행

«Step 1 설치 화면에서 설치 타입을 “Custom”으로 설치 시작



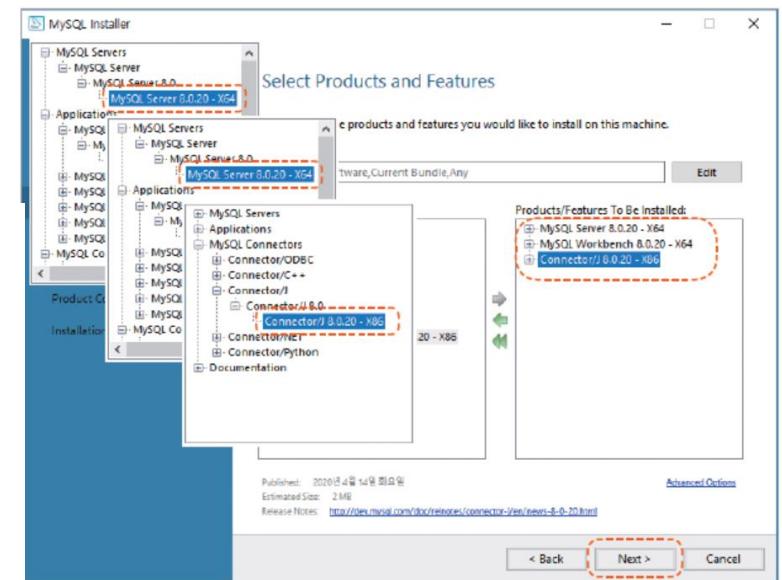
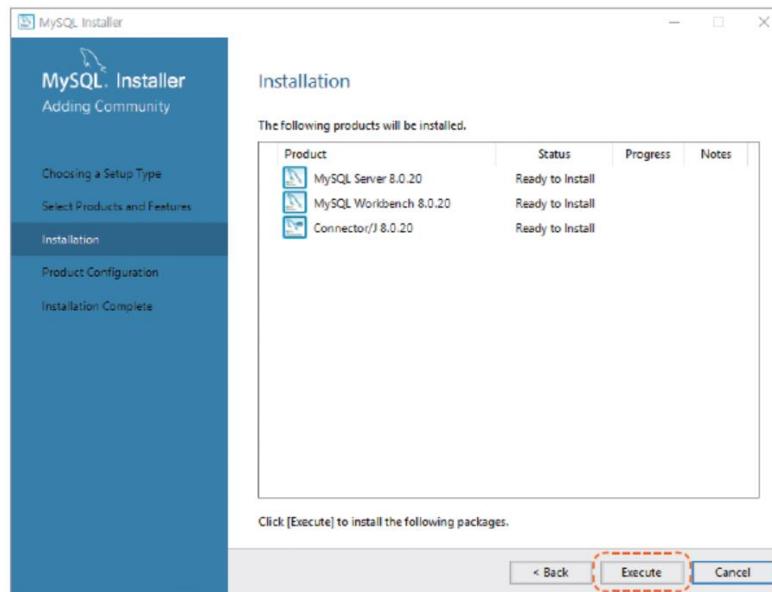
2 MySQL 데이터베이스 설치

2.2 MySQL 설치 및 설정

9th edition

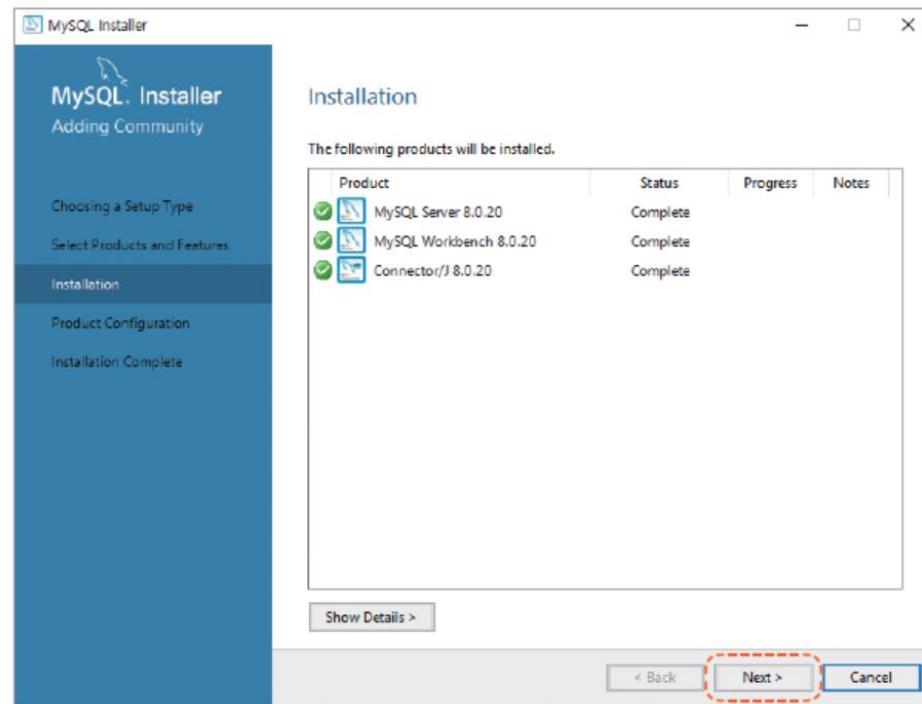
«Step 2 다음 화면에서 3개의 항목을 찾아 설정(버전이 다를 수 있음)
 “Available Products”창에서 3개의 항목을 선택

«Step 3 다음 화면에서 “Execute” 버튼으로 설치 시작



2.2 MySQL 설치 및 설정

«Step 4 설치 완료 화면. “Next”선택

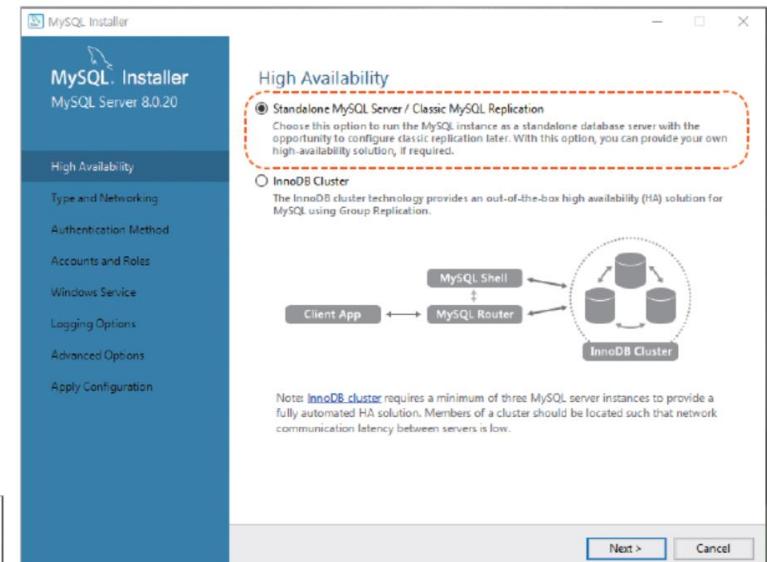
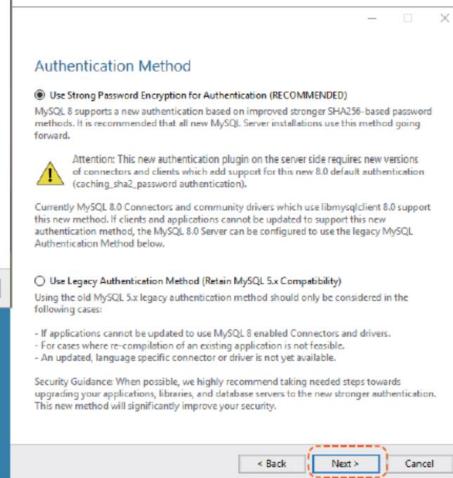
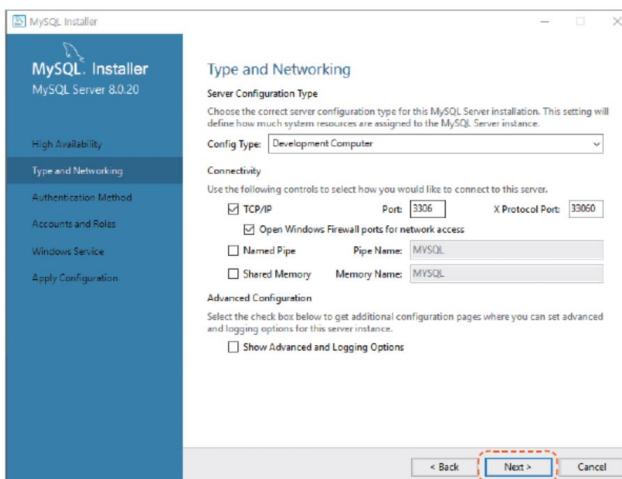


2.2 MySQL 설치 및 설정

«Step 5 MySQL 서버를 설정하는 화면

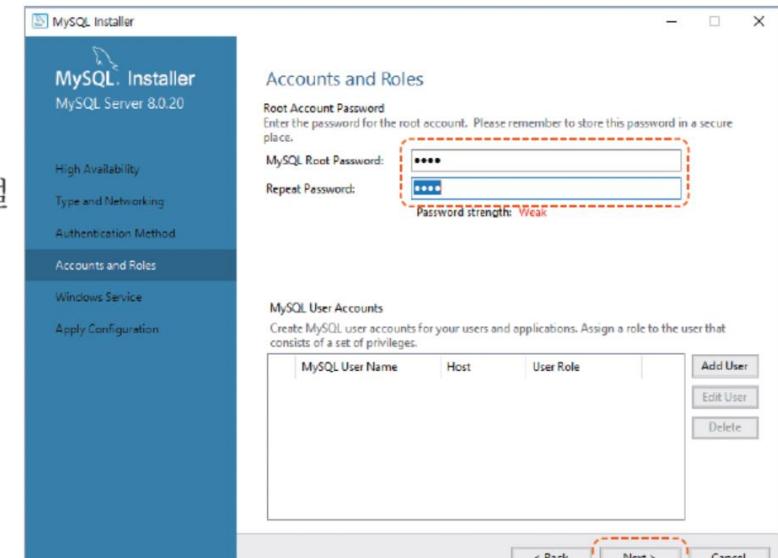
“Standalone MySQL Server/Classic MySQL Replication” 선택

설정화면 그대로 “Next” 버튼 선택

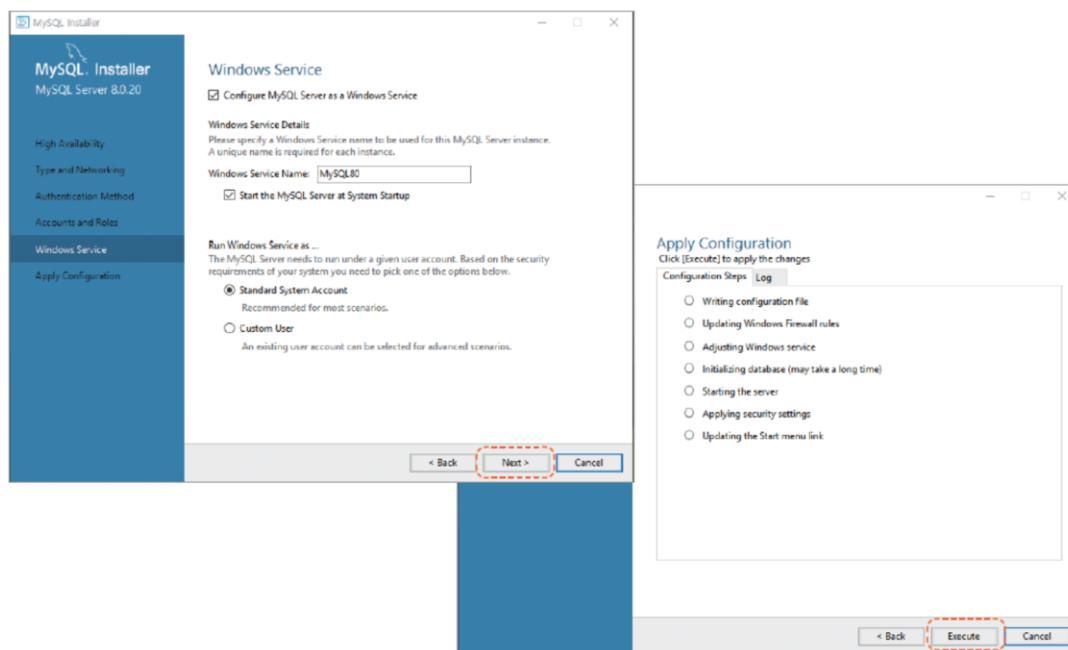


2.2 MySQL 설치 및 설정

◀ Step 6 MySQL 서버의 루트 패스워드를 설정하는 화면
암호를 설정하고 “Next” 버튼 선택

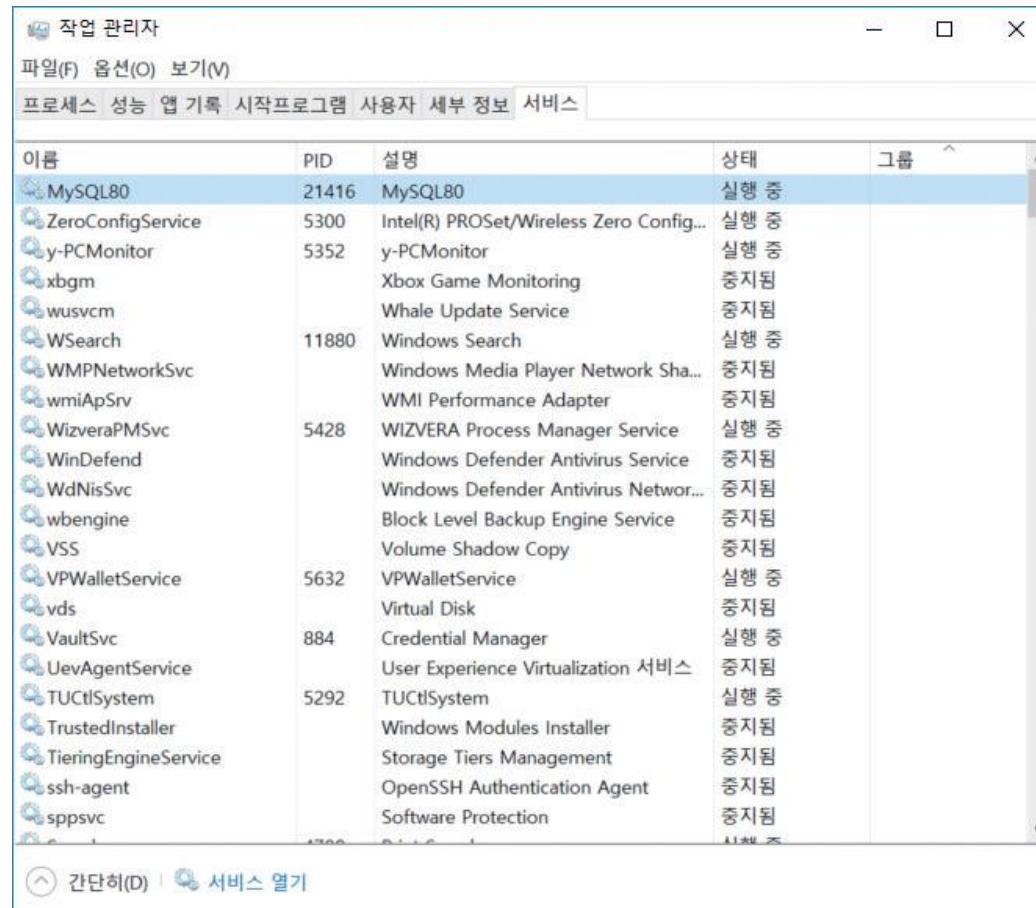


◀ Step 7 MySQL 서버를 윈도우 서비스에 등록하고 “Next” 버튼을 누르고, 다음 화면에서 설정을 등록하는 “Execute” 버튼을 선택



3.1 MySQL 서버의 실행

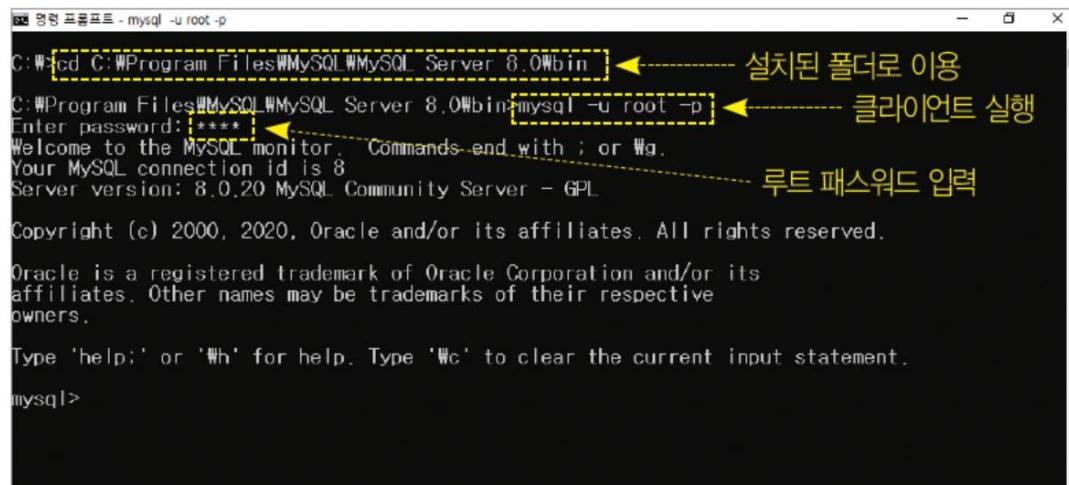
● 시스템 부팅 시 MySQL 서버 자동 실행(작업 관리자에서 확인)



3.2 MySQL 클라이언트 실행

● MySQL 명령어 라인 클라이언트 사용(MYSQL Workbench 클라이언트를 사용해도 됨)

- ① 명령어 창을 열어 MySQL의 폴더 아래의 bin 폴더로 이동
- ② mysql 명령어를 이용하여 클라이언트를 실행
- ③ 암호를 입력하여 클라이언트 프롬프트 생성



```
명령 프롬프트 - mysql -u root -p
C:\> cd C:\Program Files\MySQL\MySQL Server 8.0\bin          ← 설치된 폴더로 이용
C:\> Program Files\MySQL\MySQL Server 8.0\bin\mysql -u root -p          ← 클라이언트 실행
Enter password:*****          ← 루트 패스워드 입력
Welcome to the MySQL monitor. Commands end with ; or \q.
Your MySQL connection id is 8
Server version: 8.0.20 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- SQL Structured Query Language은 1970년대에 개발하여 현재까지 관계형 데이터베이스 관리 시스템에서 표준으로 사용하는 언어
- 관계형 데이터베이스 관리 시스템(RDBMS)은 테이블 구조의 데이터를 기본으로 사용
- SQL은 관계형 데이터베이스 관리 시스템에서 테이블을 생성, 수정, 삭제하고 테이블에 저장된 데이터를 검색, 추가, 수정, 삭제하는 등의 기능을 수행하는 표준 언어

● 관계형 데이터베이스 구성요소

- 컬럼 또는 필드 : 데이터베이스를 구성하는 가장 작은 단위의 데이터
- 행 또는 레코드 : 몇 개의 필드가 모여 하나의 행 또는 레코드로 구성
- 테이블 : 몇 개의 레코드가 모여 하나의 테이블로 구성
- 데이터베이스 : 몇 개의 테이블이 모여 하나의 데이터베이스로 구성.

database : Haksa

table : prof

Name	Dept	Subjects
CSKIM	COMP	3
ORKIM	ELEC	2
JULEE	MACH	4

table : student

ID	Name	Dept	Score
2022951110	jljung	COMP	99
2021981114	pspark	ELEC	88
2021981450	ojlee	MACH	98

컬럼(column)
또는
필드(field)

행(row)
또는
레코드(record)

그림 18-6 관계형 데이터베이스에서 레코드, 필드, 테이블, 데이터베이스 구성의 예



3 MySQL 실행과 SQL 사용

3.3 SQL 개요 – (2) SQL : DDL^{Data Definition Language}과 DML^{Data Manipulation Language}

9th edition

- SQL은 데이터베이스와 테이블을 생성하고, 수정하고, 삭제하는 기능을 제공하는 DDL과
- 테이블의 필드 값을 검색하고, 생성하고, 수정하고, 삭제하는 등의 기능을 제공하는 DML로 구성



3 MySQL 실행과 SQL 사용

3.3 SQL 개요 – (2) SQL : DDL^{Data Definition Language}과 DML^{Data Manipulation Language}

9th edition

표 18-2 DDL과 DML

SQL 구분	구문 형태	의미	구문의 예
DDL	create database	데이터베이스를 생성	create database Hksa;
	drop database	데이터베이스를 삭제	drop database Hksa;
	create table	데이터베이스에 테이블을 생성	create table student(Name varchar(10) not null, ID varchar(10) not null primary key);
	drop table	테이블을 삭제	drop table student;
	alter table	테이블 구조를 변경	alter table student add column dept varchar(10);



3 MySQL 실행과 SQL 사용

3.3 SQL 개요 – (2) SQL : DDL Data Definition Language과 DML Data Manipulation Language

9th edition

SQL 구분	구문 형태	의미	구문의 예
DML	select	테이블에서 행을 검색	select * from student;
	insert	테이블에 행을 추가	insert into student values("CSKIM","2021951110");
	update	테이블 내용을 수정	update student set ID="2021951111" where Name="CSKIM";
	delete	테이블 행을 삭제	delete from student where Name="김대한";

● 주요 필드 자료형

- 필드 : 테이블을 구성하는 가장 작은 단위의 데이터

표 18-3 관계형 데이터베이스 필드 자료형

유형	데이터 유형	의미
문자형	char(n)	고정 길이 데이터 타입(최대 255byte)– 지정된 길이보다 짧은 데이터 입력될 시 나머지 공간 공백으로 채워짐
	varchar(n)	가변 길이 데이터 타입(최대 65535byte)– 지정된 길이보다 짧은 데이터 입력될 시 나머지 공간은 채우지 않음
숫자형	int(n)	정수형 데이터 타입(4byte)
	bigint(n)	정수형 데이터 타입(8byte)
	float(size, decimal)	부동 소수형 데이터 타입(4byte)
	double(size,decimal)	부동 소수형 데이터 타입(8byte)
날짜형	date	날짜(년도, 월, 일) 형태의 기간 표현 데이터 타입(3byte)
	time	시간(시, 분, 초) 형태의 기간 표현 데이터 타입(3byte)
	datetime	날짜와 시간 형태의 기간 표현 데이터 타입(8byte)
	year	년도 표현 데이터 타입(1byte)



● 주요 SQL 명령어

① create database *databaseName* ;

특정 업무를 수행하기 위한 데이터베이스를 생성하는 명령어입니다.

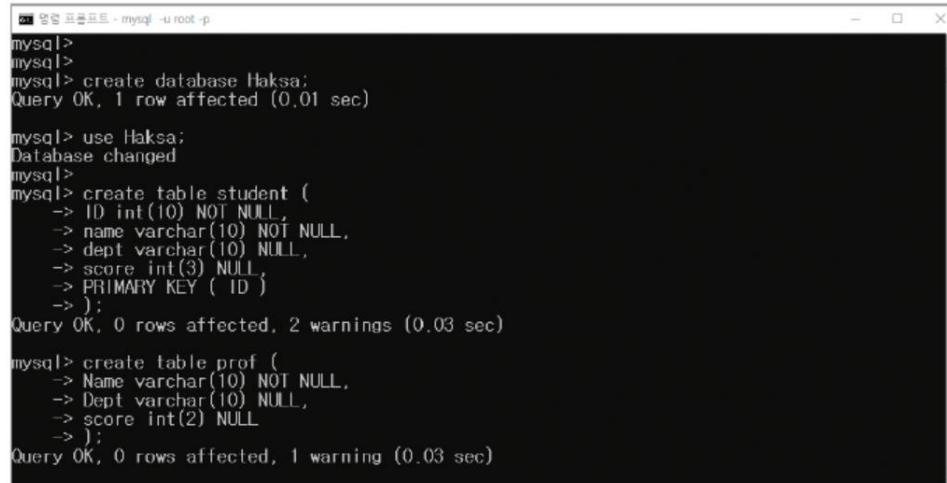
② use *databaseName* ;

사용할 데이터베이스 이름을 지정하는 명령어입니다. 이후로 사용되는 모든 SQL 명령어는 use로 지정된 데이터베이스에 적용됩니다.

● 주요 SQL 명령어

③ create table *tablename* (*fieldname fieldtype*,);

*tablename*으로 지정된 테이블을 데이터베이스에 생성하는 명령어입니다. 테이블을 구성하는 각 필드를 괄호 안에 지정합니다. 각 필드는 필드이름, 필드타입, NULL 지정여부를 지정하며, 테이블에서 키^{key} 필드를 지정하기 위해 PRIMARY KEY(...)를 이용합니다.



```
mysql>
mysql>
mysql> create database Haksa;
Query OK, 1 row affected (0.01 sec)

mysql> use Haksa;
Database changed
mysql>
mysql> create table student (
    -> ID int(10) NOT NULL,
    -> name varchar(10) NOT NULL,
    -> dept varchar(10) NULL,
    -> score int(3) NULL
    -> PRIMARY KEY ( ID )
    -> );
Query OK, 0 rows affected, 2 warnings (0.03 sec)

mysql> create table prof (
    -> Name varchar(10) NOT NULL,
    -> Dept varchar(10) NULL,
    -> score int(2) NULL
    -> );
Query OK, 0 rows affected, 1 warning (0.03 sec)
```

그림 18-7 SQL 사용 예



● 주요 SQL 명령어

④ show databases;

현재 데이터베이스 시스템에 생성된 모든 데이터베이스를 나타냅니다.

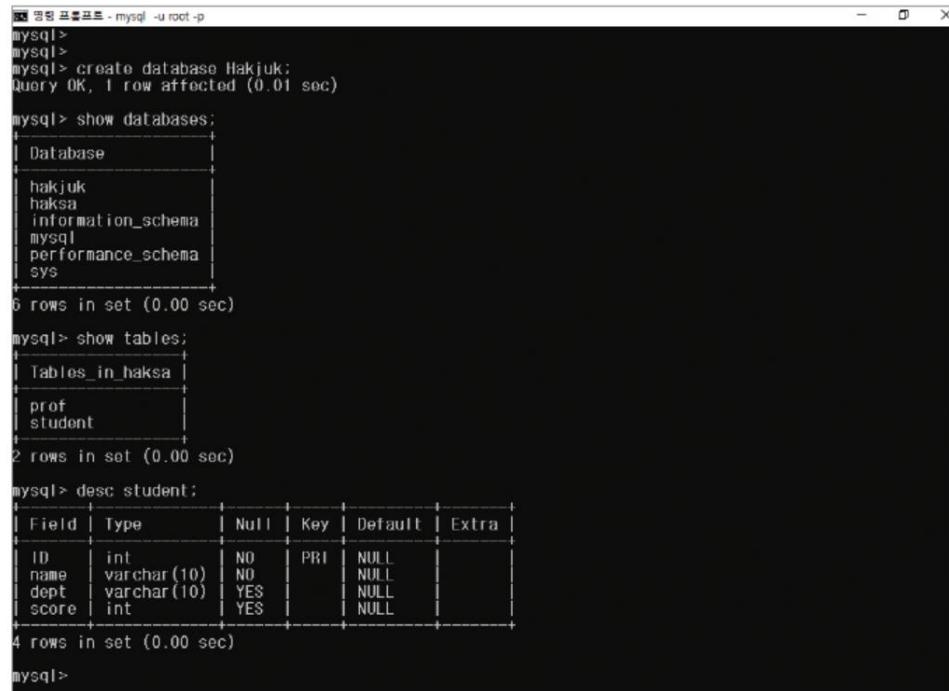
⑤ show tables;

현재 데이터베이스에서 사용되는 모든 테이블의 목록을 나타냅니다.

● 주요 SQL 명령어

⑥ desc tablename;

지정된 테이블의 구조를 나타냅니다.



```
mysql>
mysql>
mysql> create database Hakjuk;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| hakjuk   |
| haksa    |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
6 rows in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_haksa |
+-----+
| prof      |
| student   |
+-----+
2 rows in set (0.00 sec)

mysql> desc student;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| ID    | int   | NO  | PRI | NULL    |       |
| name  | varchar(10) | NO  |     | NULL    |       |
| dept  | varchar(10) | YES |     | NULL    |       |
| score | int   | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

그림 18-8 SQL 사용 예

● 주요 SQL 명령어

⑦ drop database *databasename* ;

지정된 이름의 데이터베이스를 삭제합니다. 이 명령어는 경고문 없이 바로 삭제하므로 주의해서 사용해야 합니다.

⑧ drop table *tablename* ;

사용중인 데이터베이스로부터 지정된 테이블을 삭제합니다. 이 명령어는 경고문 없이 바로 삭제하므로 주의해서 사용해야 합니다.

● 주요 SQL 명령어

⑨ alter 문장

테이블의 구조를 변경할 수 있는 명령어입니다. 다음과 같이 3가지 형태로 사용이 가능합니다.

- alter table *tablename* add column *statements* ;

*statements*로 지정된 컬럼을 테이블에 추가

- alter table *tablename* drop *columnname* ;

*columnname*로 지정된 컬럼을 테이블에서 삭제

- alter table *tablename* rename *newtablename* ;

*tablename*으로 지정된 테이블의 이름을 *newtablename*로 변경

3.4 SQL 사용

● 주요 SQL 명령어

```
명령 프롬프트 - mysql -u root -p
mysql>
mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID    | int    | NO   | PRI  | NULL    |       |
| name  | varchar(10) | NO  |       | NULL    |       |
| dept  | varchar(10) | YES |       | NULL    |       |
| score | int    | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> alter table student add column grade int(1);
Query OK, 0 rows affected, 1 warning (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID    | int    | NO   | PRI  | NULL    |       |
| name  | varchar(10) | NO  |       | NULL    |       |
| dept  | varchar(10) | YES |       | NULL    |       |
| score | int    | YES  |       | NULL    |       |
| grade | int    | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> alter table student rename std;
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_haksa |
+-----+
| prof           |
| std            |
+-----+
2 rows in set (0.00 sec)
```

● 주요 SQL 명령어

⑩ insert 문장

테이블에 하나의 레코드를 삽입하는 명령어입니다. 필드를 지정하지 않으면 테이블에 정의된 순서대로 레코드가 입력됩니다.

- `insert into tablename values (...);`

테이블에 지정된 순서로 값이 저장

- `insert into t tablename (fieldname,...,fieldname) values (...);`

필드명을 지정하여 지정된 순서대로 값을 저장

● 주요 SQL 명령어

⑪ select 문장

테이블에서 조건을 지정하여 만족하는 레코드를 읽어오는 명령어입니다.

- `select * from tablename;`

지정된 테이블의 모든 레코드를 반환

- `selectfieldname[, fieldname,...] from tablename;`

지정된 테이블로부터 레코드를 반환. 단 지정된 필드만 반환

- `select * from tablename where fieldname = fieldvalue ;`

지정된 테이블로부터 fieldvalue 값으로 지정된 레코드만 반환



3 MySQL 실행과 SQL 사용

3.4 SQL 사용

9th edition

● 주요 SQL 명령어

```
명령 프롬프트 - mysql -u root -p
mysql>
mysql> insert into std values(2021951110,'jspark','computer',99,2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into std values(2022981131,'pklee','english',100,1);
Query OK, 1 row affected (0.01 sec)

mysql> select * from std;
+----+-----+-----+-----+-----+
| ID | name | dept | score | grade |
+----+-----+-----+-----+-----+
| 2021951110 | jspark | computer | 99 | 2 |
| 2022981131 | pklee | english | 100 | 1 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
mysql> select ID, name from std where score=100;
+----+-----+
| ID | name |
+----+-----+
| 2022981131 | pklee |
+----+-----+
1 row in set (0.00 sec)

mysql> select * from std where score >= 80;
+----+-----+-----+-----+-----+
| ID | name | dept | score | grade |
+----+-----+-----+-----+-----+
| 2021951110 | jspark | computer | 99 | 2 |
| 2022981131 | pklee | english | 100 | 1 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

● 주요 SQL 명령어

⑫ update 문장

테이블의 내용을 수정하는 명령어입니다.

- `update tablename set columnname = value [, columnname = value,.....] [where 조건]`

주어진 테이블에서 조건에 적합한 필드(컬럼)의 내용을 value로 설정

⑬ delete 문장

테이블에서 전체 또는 조건에 맞는 레코드를 삭제하는 명령어입니다.

- `delete from tablename;`

주어진 테이블의 모든 레코드를 삭제

- `delete from tablename where columnname = value ;`

주어진 테이블에서 조건에 적합한 레코드만 삭제

3.4 SQL 사용

● 주요 SQL 명령어

```
MySQL - MySQL 편집기 - mysql -u root -p
mysql>
mysql> select * from std;
+----+-----+-----+-----+-----+
| ID | name | dept | score | grade |
+----+-----+-----+-----+-----+
| 2021951110 | jspark | computer | 99 | 2 |
| 2022981131 | pklee | english | 100 | 1 |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> update std set score=100 where grade=2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from std;
+----+-----+-----+-----+-----+
| ID | name | dept | score | grade |
+----+-----+-----+-----+-----+
| 2021951110 | jspark | computer | 100 | 2 |
| 2022981131 | pklee | english | 100 | 1 |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> delete from std where grade=1;
Query OK, 1 row affected (0.00 sec)

mysql> select * from std;
+----+-----+-----+-----+-----+
| ID | name | dept | score | grade |
+----+-----+-----+-----+-----+
| 2021951110 | jspark | computer | 100 | 2 |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

자바 언어와 DBMS를 연동하기 위한 API와 드라이버

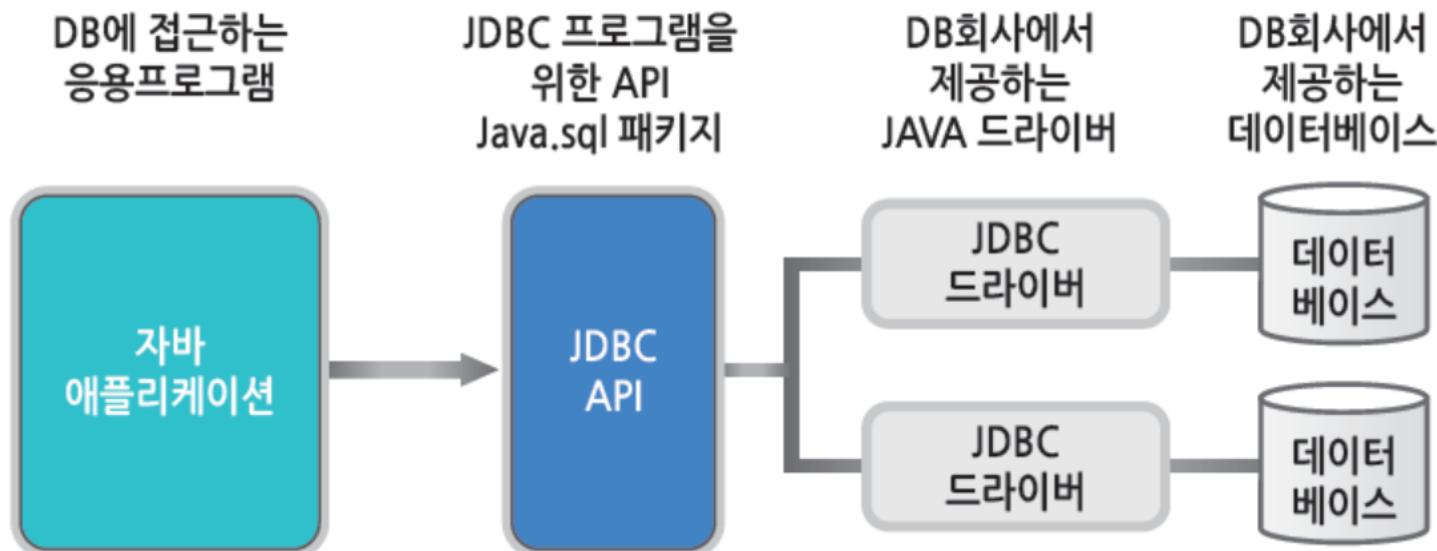
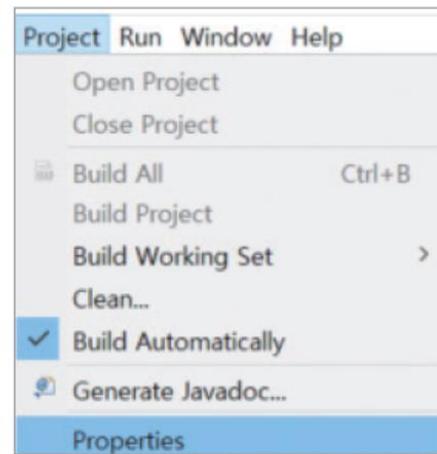


그림 18-12 데이터베이스를 사용하는 자바 프로그램



4.1 이클립스에서 JDBC 드라이버의 경로를 지정하는 방법

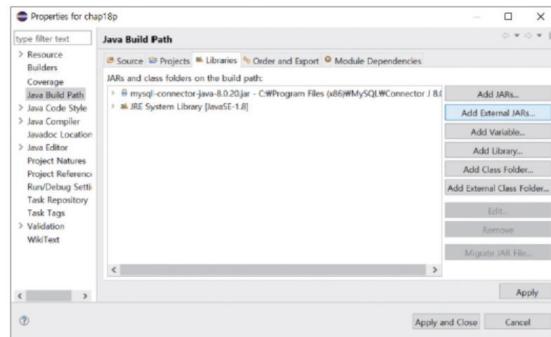
«Step 1 이클립스에서 “Project”=>“Properties”를 선택



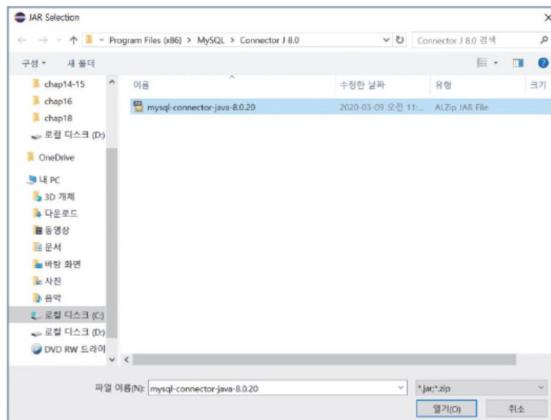
4.1 이클립스에서 JDBC 드라이버의 경로를 지정하는 방법

9th edition

«Step 2 팝업 창에서 “Libraries”=>“Add External JARs” 버튼을 선택



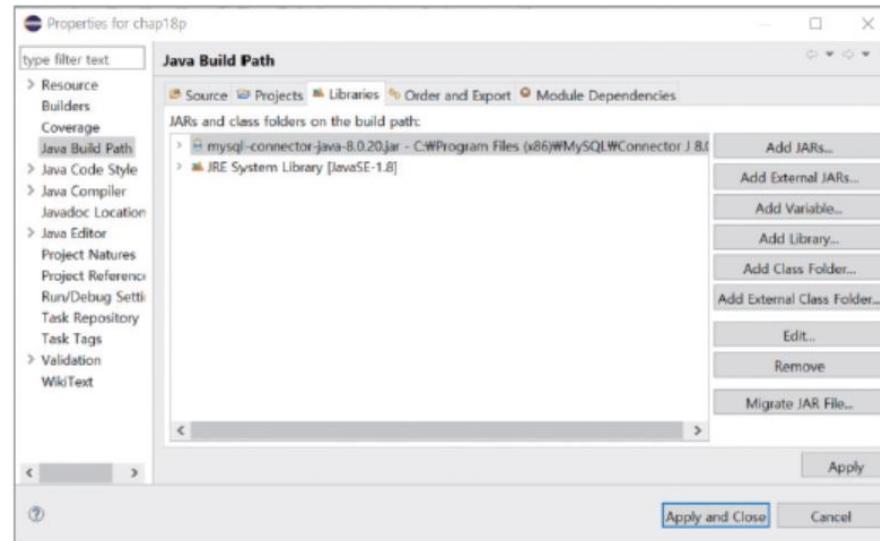
«Step 3 팝업창에서 “C:\Program Files (x86)\MySQL\Connector J 8.0” 폴더에 있는 “mysql-connector-java-8.0.20” 파일을 선택(버전에 따라 이름이 다를 수 있음)



4.1 이클립스에서 JDBC 드라이버의 경로를 지정하는 방법

9th edition

«Step 4 경로가 지정된 화면





4.2 JDBC 프로그래밍

JDBC 프로그래밍은 다음과 같은 전형적인 절차로 진행됩니다.

- ① URL로 지정된 JDBC 드라이버를 로드
- ② 이름과 패스워드를 이용하여 데이터베이스 연결
- ③ SQL 명령어를 사용하여 결과를 얻는다
- ④ 결과를 적절한 방법으로 출력
- ⑤ 프로그램에서 사용한 드라이버 등과 같은 자원을 닫는다



4.2 JDBC 프로그래밍

예제 18.1

JDBCTest1.java

```
01: import java.sql.Connection;
02: import java.sql.DriverManager;
03: import java.sql.SQLException;
04: //java.sql 패키지의 클래스들을 import
05:
06: public class JDBCTest1 {
07:     public static void main(String args[]) {
08:         Connection conn;
09:         String url = "jdbc:mysql://localhost:3306/Haksa?serverTimezone=UTC";
10:         //자신의 컴퓨터에 저장한 Haksa 데이터베이스와 연결하기 위한 url
11:
12:         String id="root"; //데이터베이스 root id
13:         String pw="1234"; //패스워드
14:
```



4.2 JDBC 프로그래밍

```
15:     try {  
16:         Class.forName("com.mysql.cj.jdbc.Driver");  
17:         // JDBC 드라이버를 로드함  
18:  
19:         conn = DriverManager.getConnection(url,id,pw);  
20:         // url과 id, 패스워드로 데이터베이스와 연결  
21:  
22:         System.out.println("DB연결완료");  
23:         // 오류없이 연결이 되는 경우 출력  
24:  
25:     } catch(ClassNotFoundException e) {  
26:         System.out.println("JDBC 드라이버 로드 오류");  
27:         // JDBC 드라이버 로드 오류  
28:  
29:     } catch(SQLException e) {  
30:         System.out.println("DB 연결 오류");  
31:         // DB 연결 오류  
32:     }  
33: }  
34: }
```

실행 결과

DB연결완료



4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

 Statement 클래스

Connection 클래스의 `createStatement()` 메소드는 Statement 클래스의 객체를 반환합니다. Statement 클래스에서 제공되는 메소드를 이용하여 SQL 명령어를 실행합니다.

표 18-4 Statement 클래스의 주요 메소드

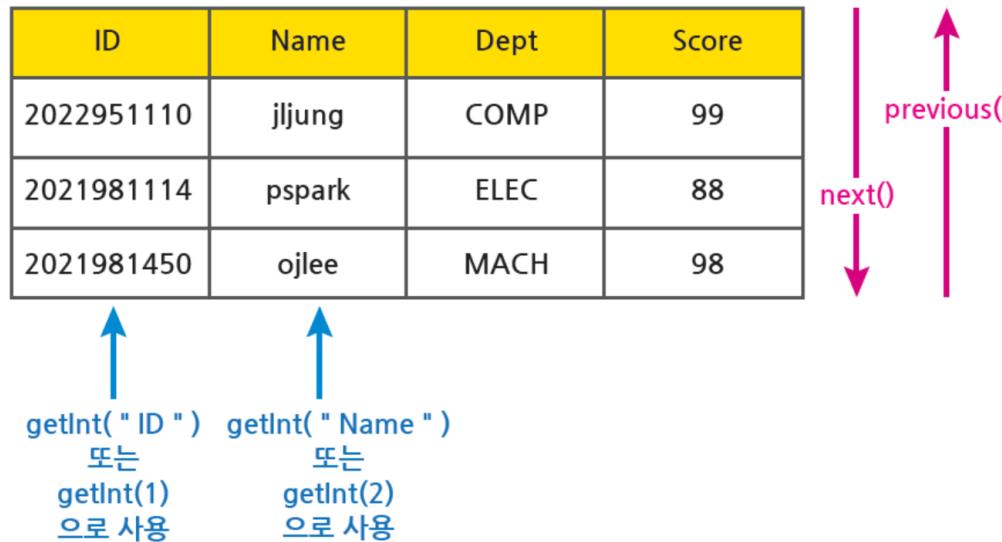
메소드	설명
<code>ResultSet executeQuery(String sql)</code>	지정된 sql 명령을 수행하고 결과를 ResultSet 객체로 반환
<code>int executeUpdate(String sql)</code>	지정된 sql(insert, delete, update) 명령을 수행하고 수정되거나 삭제된 레코드의 수를 반환
<code>void close()</code>	Statement 객체와 JDBC 자원을 반환

4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

ResultSet 클래스

`executeQuery(String sql)` 명령의 실행결과가 ResultSet 객체로 반환됩니다. ResultSet 클래스에서 제공되는 다양한 메소드를 이용하여 반환된 결과를 출력합니다.

ResultSet 객체





4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

표 18-5 ResultSet 클래스의 메소드

메소드	설명
boolean first()	커서를 첫 번째 행으로 이동
boolean last()	커서를 마지막 행으로 이동
boolean next()	커서를 다음 행으로 이동
boolean previous()	커서를 이전 행으로 이동
boolean absolute(int row)	row로 지정된 행으로 커서 이동
boolean isFirst()	첫 번째 행이면 true 반환
boolean isLast()	마지막 행이면 true 반환
void close()	Result 객체의 DB와 드라이버 반환
Type getType(String columnname)	columnname으로 지정된 컬럼의 값을 반환 반환되는 타입은 컬럼의 타입
Type getType(int columnindex)	columnindex로 지정된 컬럼의 값을 반환 columnindex는 1부터 시작



4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

예제 18.2

JDBCTest2.java

```
01: import java.sql.*;  
02:  
03: public class JDBCTest2 {  
04:     public static void main(String args[]) {  
05:         Connection conn;  
06:         String url = "jdbc:mysql://localhost:3306/Haksa?serverTimezone=UTC";  
07:  
08:         String id="root";  
09:         String pw="1234";  
10:  
11:         Statement stmt;  
12:         ResultSet result;  
13:  
14:         try {  
15:             Class.forName("com.mysql.cj.jdbc.Driver");  
16:             conn = DriverManager.getConnection(url,id,pw);  
17:             System.out.println("DB연결완료");  
18:  
19:             stmt = conn.createStatement();  
20:             // createStatement() 메소드를 사용하여 Statement 객체를 반환  
21:
```



4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

```
22:     result = stmt.executeQuery("select * from std");
23:     // 객체의 executeQuery() 메소드를 사용하여 SQL을 실행하고 결과를
        ResultSet 객체로 반환
24:
25:     while(result.next()) { // 결과의 끝까지 반복
26:         int hakbun = result.getInt(1); // 인덱스를 이용
27:         String name = result.getString("name"); // 컬럼 이름을 이용
28:         String dept = result.getString("dept");
29:         int score = result.getInt(4);
30:         int grade = result.getInt("grade");
31:         System.out.println(hakbun+ " "+name+ " "+dept+ " "+score+ " "+grade);
32:         //결과를 출력
33:     }
34:     result.close(); // 사용된 자원을 반환
35:     stmt.close();
36:     conn.close();
37:
38: } catch(ClassNotFoundException e) {
39:     System.out.println("JDBC 드라이버 로드 오류");
40: } catch(SQLException e) {
41:
42:     System.out.println("DB 연결 오류");
43: }
```

실행 결과

DB연결완료

2021951110 jspark computer 100 2



4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

예제 18.3

JDBCTest3.java

```
01: import java.sql.Connection;
02: import java.sql.DriverManager;
03: import java.sql.ResultSet;
04: import java.sql.SQLException;
05: import java.sql.Statement;
06:
07: public class JDBCTest3 {
08:     public static void main(String args[]) {
09:         Connection conn;
10:         String url = "jdbc:mysql://localhost:3306/Haksa?serverTimezone=UTC";
11:
12:         String id="root";
13:         String pw="1234";
14:
15:         Statement stmt;
16:         ResultSet result;
17:
```



4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

```
18:     try {
19:         Class.forName("com.mysql.cj.jdbc.Driver");
20:         conn = DriverManager.getConnection(url,id,pw);
21:         System.out.println("DB연결완료");
22:
23:         stmt = conn.createStatement();
24:         // createStatement() 메소드를 사용하여 Statement 객체를 반환
25:
26:         stmt.executeUpdate("insert into std values(2022951111,'pklee','Manageme
27: nt',98,3)");
28:         stmt.executeUpdate("insert into std values(2021981133,'sdkim','Music',100
29: ,1)");
30:         stmt.executeUpdate("insert into std values(2022951141,'chlim','FineA
31: rt',99,2)");
32:         // 객체의 executeUpdate() 메소드를 사용하여 SQL을 실행하여 테이블에
33:         레코드를 추가
34:
35:         result = stmt.executeQuery("select * from std");
36:         // 객체의 executeQuery() 메소드를 사용하여 SQL을 실행하고 결과를
37:         ResultSet 객체로 반환
38:         while(result.next()) { // 결과의 끝까지 반복
39:             int hakbun = result.getInt(1); // 인덱스를 이용
40:             String name = result.getString("name"); // 컬럼 이름을 이용
41:             String dept = result.getString("dept");
42:             int score = result.getInt(4);
43:             int grade = result.getInt("grade");
44:             System.out.println(hakbun+" "+name+" "+dept+" "+score+" "+grade);
45:             //결과를 출력
46:         }
47:         result.close(); // 사용된 자원을 반환
48:         stmt.close();
49:         conn.close();
50:
```



4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

```
46:  
47:     } catch(ClassNotFoundException e) {  
48:         System.out.println("JDBC 드라이버 로드 오류");  
49:     } catch(SQLException e) {  
50:         System.out.println("DB 연결 오류");  
51:     }  
52: }  
53: }
```

실행 결과

```
DB연결완료  
2021951110 jspark computer 100 2  
2021981133 sdkim Music 100 1  
2022951111 pklee Management 98 3  
2022951141 chlim FineArt 99 2
```



4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

예제 18.4

JDBCTest3.java

```
01: import java.sql.Connection;
02: import java.sql.DriverManager;
03: import java.sql.ResultSet;
04: import java.sql.SQLException;
05: import java.sql.Statement;
06:
07: public class JDBCTest4 {
08:     public static void main(String args[]) {
09:         Connection conn;
10:         String url = "jdbc:mysql://localhost:3306/Haksa?serverTimezone=UTC";
11:
12:         String id="root";
13:         String pw="1234";
14:
15:         Statement stmt;
16:         ResultSet result;
17:
18:         try {
19:             Class.forName("com.mysql.cj.jdbc.Driver");
20:             conn = DriverManager.getConnection(url,id,pw);
21:             System.out.println("DB연결완료");
22:
23:             stmt = conn.createStatement();
24:             // createStatement() 메소드를 사용하여 Statement 객체를 반환
```



4.2 JDBC 프로그래밍 (1) java.sql 패키지 클래스

```
25:  
26:     stmt.executeUpdate("delete from std where grade = 2");  
27:     // 객체의 executeUpdate() 메소드를 사용하여 SQL을 실행하여 테이블에  
28:     // 레코드를 삭제  
29:  
30:     result = stmt.executeQuery("select * from std");  
31:     // 객체의 executeQuery() 메소드를 사용하여 SQL을 실행하고 결과를  
32:     // ResultSet 객체로 반환  
33:  
34:     while(result.next()) { // 결과의 끝까지 반복  
35:         int hakbun = result.getInt(1); // 인덱스를 이용  
36:         String name = result.getString("name"); // 컬럼 이름을 이용  
37:         String dept = result.getString("dept");  
38:         int score = result.getInt(4);  
39:         int grade = result.getInt("grade");  
40:         System.out.println(hakbun+ " "+name+ " "+dept+ " "+score+ " "+grade);  
41:         // 결과를 출력  
42:     }  
43:  
44:     result.close(); // 사용된 자원을 반환  
45:     stmt.close();  
46:     conn.close();  
47:  
48: } catch(ClassNotFoundException e) {  
49:     System.out.println("JDBC 드라이버 로드 오류");  
50: } catch(SQLException e) {  
51:     System.out.println("DB 연결 오류");  
52: }
```

실행 결과

DB연결완료
2021981133 sdkim Music 100 1
2022951111 pklee Management 98 3



● 편의점 POS Point Of Sales 정보시스템 구현

▣ 프로젝트 개요

- 편의점 POS 시스템

- 관리자

제품 등록, 조회, 수정, 삭제기능

제품 재고 관리 기능

- 사용자(알바생)

제품 선택, 수량 입력, 제품 추가, 제품 판매 목록 확인, 총 금액 확인, 결재, 취소, 받은 금액 입력, 잔돈 계산



● 편의점 POS Point Of Sales 정보시스템 구현

POS 시스템

POS 재고관리

제품 불러오기

1	커피	2	2400
2	삼각김밥	1	700
3	새우깡	5	5000

상품 새우깡

수량 5

추가

총가격 8100

결제 취소

POS 화면

POS 시스템

POS 재고관리

재고현황

상품 새로 고침

등록

수정

삭제

21	라면	10	1000
22	커피	200	1200
23	콜라	100	1000
24	삼각김밥	8	700
25	새우깡	100	1000

재고관리화면



● 편의점 POS Point Of Sales 정보시스템 구현

▣ 프로젝트 주요 기능

주요기능	설명	비고
상품 불러오기	[제품 불러오기] 버튼을 눌러 구입 가능한 상품 목록을 상품 콤보 박스에 불러온다.	재고 현황에 등록된 상품 정보를 가져옴
상품	구입 가능한 상품이 목록으로 보인다.	콤보 박스에서 구매할 상품 선택
수량	구매 수량을 입력한다.	
총가격	구매하는 상품의 총 가격이 표시된다.	
추가	구매 상품과 수량을 입력하고 [추가]버튼을 누르면 구매할 상품과 수량, 가격이 표시된다.	입력한 수량이 재고 수량보다 많으면 경고 메시지 표시
결재	[결재] 버튼을 누르면 지불할 금액을 입력 받는다. 금액 입력 후 [확인] 버튼을 누르면 거스름돈 금액이 뜬다.	
취소	[취소] 버튼을 누르면 선택한 상품이 모두 취소되어 화면이 초기화 된다.	

▣ 프로젝트 주요 기능 – 재고 관리

주요기능	설명	비고
상품 새로 고침	데이터베이스에 저장된 상품 불러오기	재고 상황 보기
등록	데이터베이스에 상품 등록	등록 후 [상품 새로 고침] 버튼을 눌러 데이터베이스에 저장된 목록을 불러온다.
수정	데이터베이스에 등록된 상품 수정	수량과 가격만 수정 가능
삭제	데이터베이스에 저장된 상품 삭제	삭제 후 [상품 새로 고침]

● 편의점 POS Point Of Sales 정보시스템 구현

▣ 테이블 생성

컬럼	데이터 타입	용도	비고
id	int	연번	primary key
item_name	varchar(45)	상품명	unique key
item_stock	varchar(45)	상품 개수	
item_price	varchar(45)	상품 가격	

```

MySQL 5.6 Command Line Client

mysql> use pos;
Database changed
mysql> CREATE TABLE item (
    -> id int NOT NULL AUTO_INCREMENT,
    -> item_name varchar(45) NOT NULL,
    -> item_stock varchar(45) NOT NULL,
    -> item_price varchar(45) NOT NULL,
    -> PRIMARY KEY (id),
    -> UNIQUE KEY item_name_UNIQUE (item_name)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> desc item;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+
| id    | int(11) | NO   | PRI  | NULL    | auto_increment |
| item_name | varchar(45) | NO   | UNI  | NULL    |                |
| item_stock | varchar(45) | NO   |      | NULL    |                |
| item_price | varchar(45) | NO   |      | NULL    |                |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> alter table item convert to charset utf8;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>

```



- 편의점 POS Point Of Sales 정보시스템 구현

-상세 코드 교재 참조

● 편의점 POS Point Of Sales 정보시스템 프로그램 실행 화면

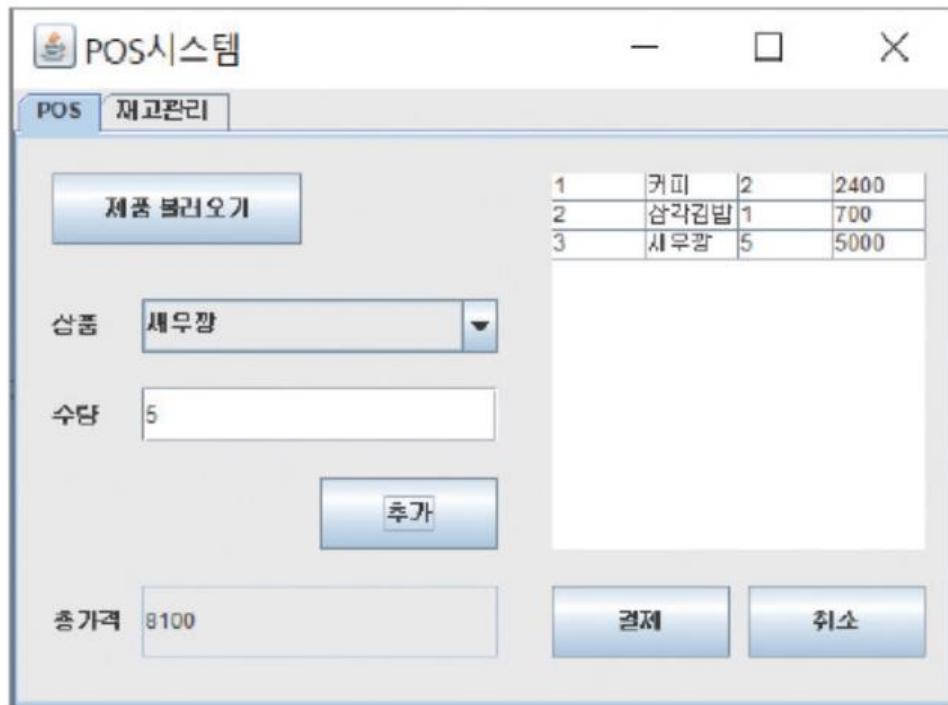


그림 18-16 POS 화면 – 구입 상품 추가

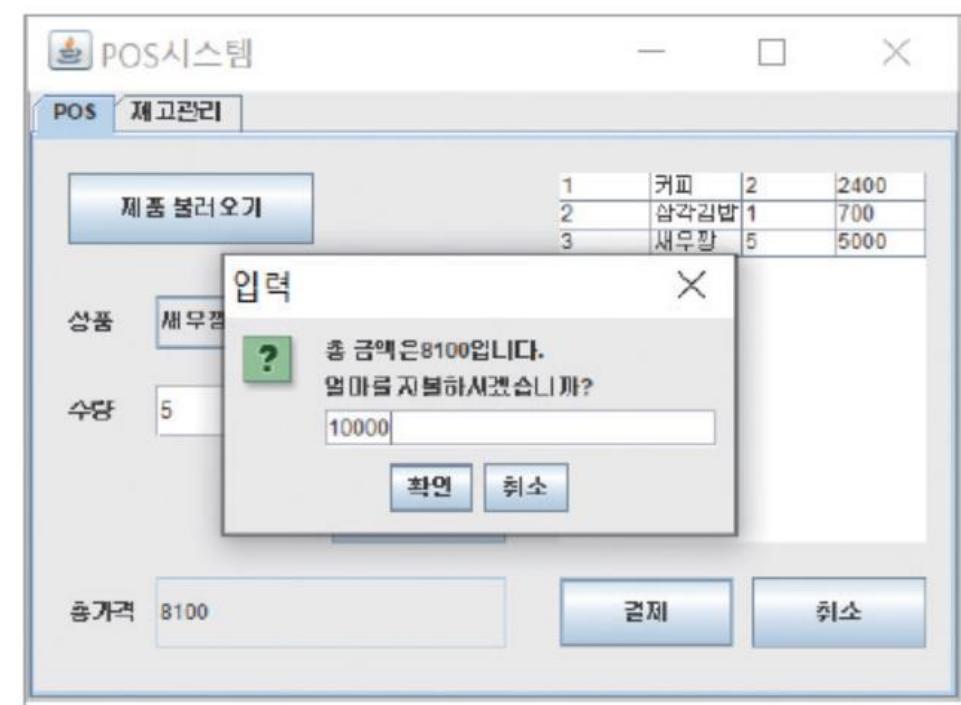


그림 18-17 POS 결재 화면

● 편의점 POS Point Of Sales 정보시스템 프로그램 실행 화면

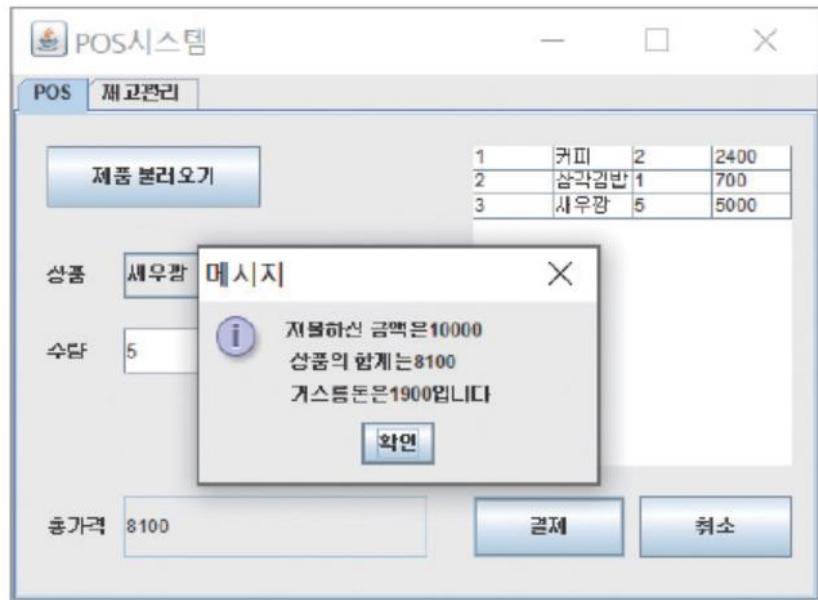


그림 18-18 POS 결재 완료 화면

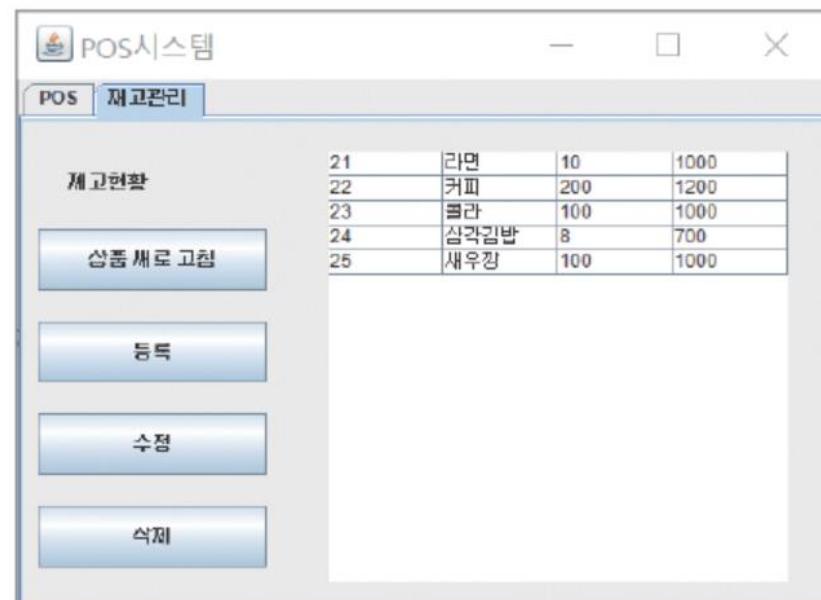


그림 18-19 재고관리 상품 새로 고침 화면

● 편의점 POS Point Of Sales 정보시스템 프로그램 실행 화면

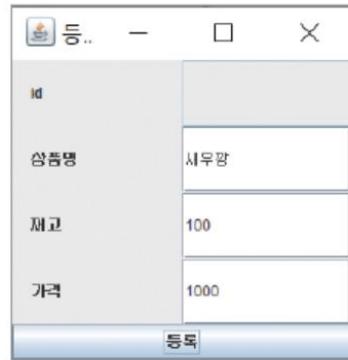
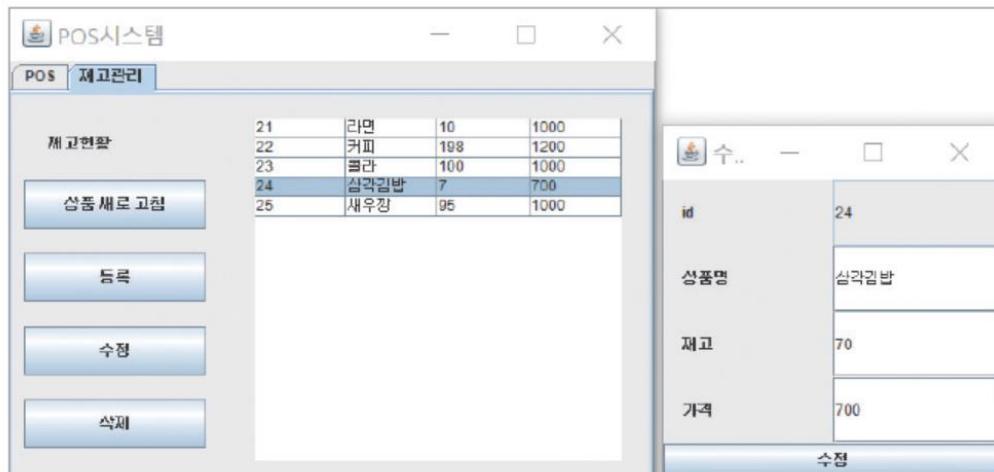


그림 18-20 재고관리 등록 화면



제고현황	21 라면 10 1000
	22 커피 100 1200
	23 물 100 1000
	24 김각길밥 7 700
	25 새우장 95 1000

id	24
상품명	김각길밥
재고	70
가격	700

그림 18-21 재고관리 수정 화면

● 편의점 POS Point Of Sales 정보시스템 프로그램 실행 화면

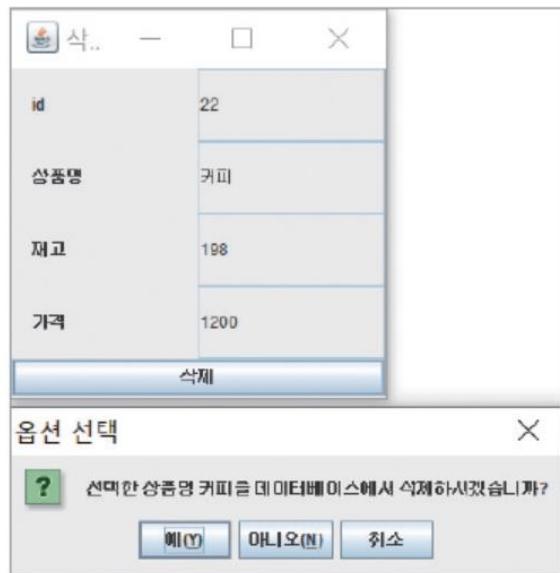


그림 18-22 재고관리 삭제 화면

- 학생 정보 수정 후 화면



연번	학과	학번	학년	이름	연락처
2	컴퓨터공학과	201895012	3	홍길동	010-1234-5678
3	컴퓨터공학과	201895017	3	이황	010-4567-8978
4	인공지능공학과	202045033	1	미순신	010-3212-6545
5	인공지능공학과	202045077	1	신사임당	010-9832-4585



● 편의점 POS Point Of Sales 정보시스템 프로그램 실행 화면

- 학생 정보 삭제 화면 – 정보 조회

Student Information

학생 정보를 가져왔습니다!!

연 번	4
학 과	인공지능공학과
학 번	202045033
학 년	1
이 름	이순신
연락처	010-3212-6545

등록

조회

수정

삭제

연 번	학 과	학 번	학 년	이 름	연락처
2	컴퓨터공학과	201895012	3	홍길동	010-1234-5678
3	컴퓨터공학과	201895017	3	미황	010-4567-8978
4	인공지능공학과	202045033	1	이순신	010-3212-6545
5	인공지능공학과	202045077	1	신사임당	010-9832-4585

● 편의점 POS Point Of Sales 정보시스템 프로그램 실행 화면

- 학생 정보 삭제 화면 – 삭제

