JDBC

강사 : 강병준

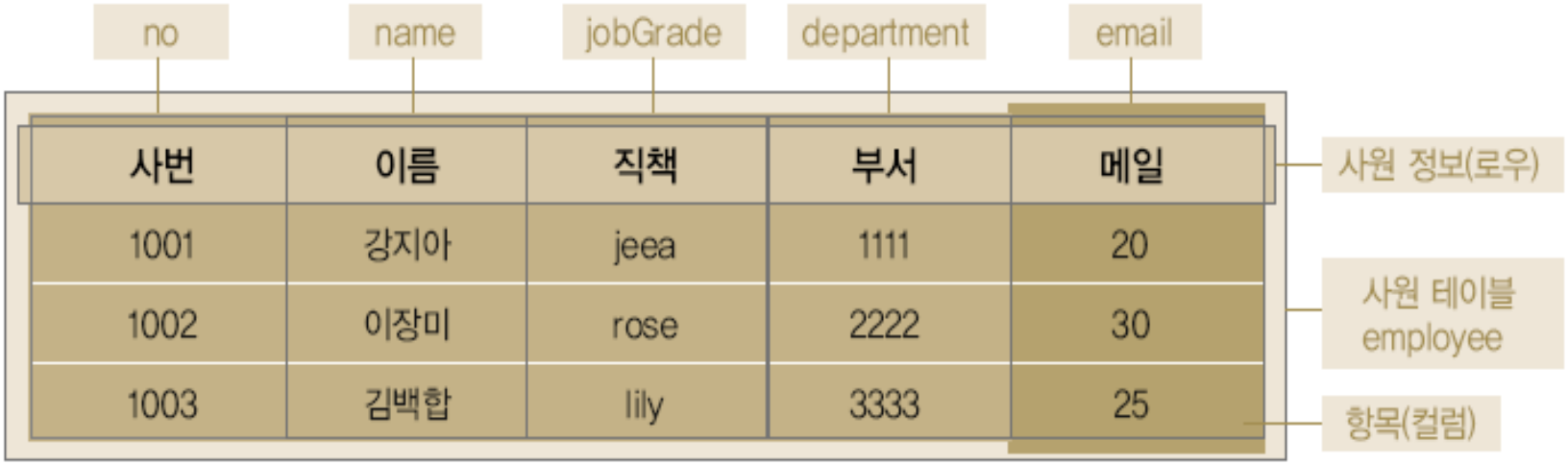
데이터베이스 개요

연관된 데이터들의 묶음을 데이터베이스라고 한다.
데이터베이스에서는 정보를 저장하기 위해서 테이블을 사용한다.

테이블은 표처럼 볼 수 있도록 로우(ROW)와 칼럼(COLUMN)으로 구성한다.


```
module ch17 {
        requires java.sql;
}
```

데이터베이스 개요




```
mysql> create table employee (
    no int not null auto_increment primary key,
    name varchar(20) not null,
    jobGrade varchar(10),
    department int(2),
    email varchar(20) );
```

데이터베이스 개요

■ insert

데이터를 입력하기 위한 명령어는 INSERT이다.

```
INSERT INTO tablename (cname1, …cnameN) VALUES (value1, …valueN)
```

테이블명 / 입력할 컬럼명들 / 입력할 값들

```
insert into employee(name, jobGrade, department, email)
values('김네이버', '차장', 10, 'naver@magic.com');

insert into employee(name, jobGrade, department, email)
values('이다음', '부장', 20, 'daum@magic.com');

insert into employee(name, jobGrade, department, email)
values('최엠파스', '대리', 20, 'empas@magic.com');
```

JDBC 프로그래밍

JDBC(Java Database Connectivity)

자바 패키지의 일부로 자바 프로그램이 데이터베이스와 연결되어

데이터를 주고받을 수 있게 해 주는 프로그래밍 인터페이스이다.

자바 데이터베이스 프로그래밍 API라고 할 수 있다.

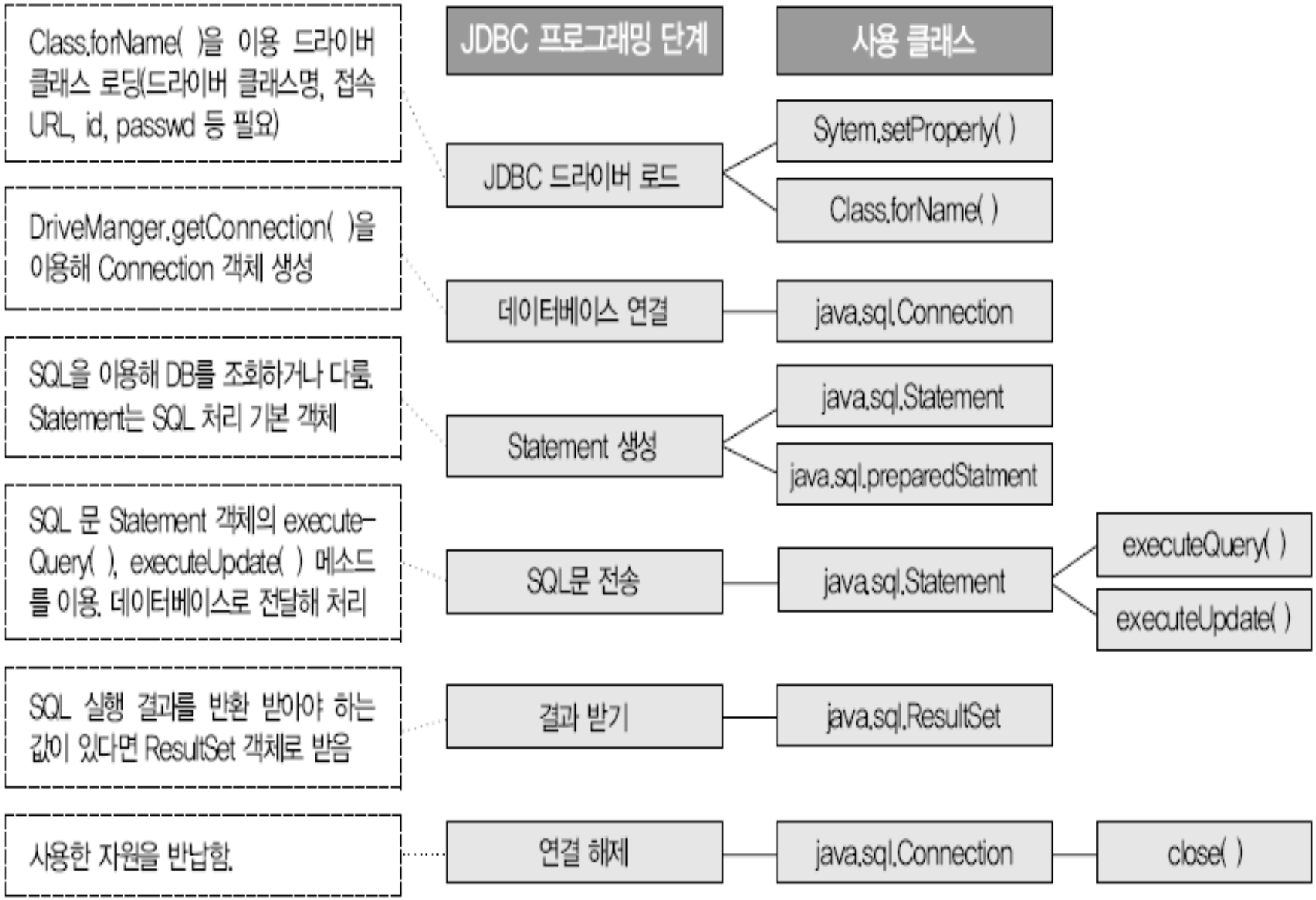
JDBC 드라이버

My SQL 또는 오라클 드라이버를 자바가 설치되어 있는 JAVA_HOME의 위치

에서 %JAVA_HOME%/jdk/jre/lib/ext란 폴더에 복사한다.

OJDBC14.JAR, mysql-connector-java-5.1.15-bin.jar

JDBC 프로그래밍


Class.forName( )을 이용 드라이버 클래스 로딩(드라이버 클래스명, 접속 URL, id, passwd 등 필요)
DriveManger.getConnection( )을 이용해 Connection 객체 생성
SQL을 이용해 DB를 조회하거나 다룸. Statement는 SQL 처리 기본 객체
SQL 문 Statement 객체의 execute-Query( ), executeUpdate( ) 메소드를 이용. 데이터베이스로 전달해 처리
SQL 실행 결과를 반환 받아야 하는 값이 있다면 ResultSet 객체로 받음
사용한 자원을 반납함.
JDBC 프로그래밍 단계
사용 클래스
JDBC 드라이버 로드
Sytem.setProperly( )
Class.forName( )
데이터베이스 연결
java.sql.Connection
Statement 생성
java.sql.Statement
java.sql.preparedStatment
SQL문 전송
java.sql.Statement
executeQuery( )
executeUpdate( )
결과 받기
java.sql.ResultSet
연결 해제
java.sql.Connection
close( )

JDBC 프로그래밍

java.lang.Class 클래스의 정적 메소드 forName()는 패키지 명을 기술한 후 클래스 이름을 문자열 형태로 지정해 주면 이를 자바 가상 기계에 안으로 읽어 들이도록 한다.

JDBC 드라이버는 다음과 같이 로드한다.

예	Class.forName('oracle.jdbc.driver.OracleDriver');

mySQL : Class.forName('com.mysql.jdbc.Driver');

Connection 객체

데이터베이스를 연결해 작업을 수행할 수 있도록 만들어 주는 중요한 객체

DriverManager 클래스의 static 메소드인 getConnection()을 호출해야 한다.

예	Connection con = DriverManager.getConnection(url , uid, pwd); ⓐ ⓑ ⓒ

ⓐ 관계형 데이터베이스 엔진에서 위치,

ⓑ 사용자 계정

ⓒ 사용자 패스워드

JDBC 프로그래밍

```
import java.sql.*;
class  JDBC_Connect{
     public static void main(String[] args) {
          String driver =  “oracle.jdbc.driver.OracleDriver";
          String url =  “jdbc:oracle:thin:@127.0.0.1:1521:xe"; // orcl
          Connection con = null;
          try{     Class.forName(driver);
             con = DriverManager.getConnection(url, “scott “,” tiger" );
              System.out.println("데이터베이스 연결 성공!");
            }  catch(Exception e){
                  System.out.println("데이터베이스 연결 실패!");
               } finally{
                           try{
                                if(con != null) con.close();
                           }  catch(Exception e){
                               System.out.println( e.getMessage());
                           }
                   }
          }
}
```

MySQL 8.0

```
import java.sql.*;
public class MyConnect {
	public static void main(String[] args) {
		String driver="com.mysql.cj.jdbc.Driver";
		String url =
"jdbc:mysql://127.0.0.1:3306/test?useSSL=false&serverTimezone=UTC&allow
PublicKeyRetrieval=true";
		try {
			Class.forName(driver);
			Connection conn =
	DriverManager.getConnection(url,"root","mysql");
			if (conn != null)
				System.out.println("연결성공 대~박");
			conn.close();
		}catch (Exception e) {
			System.out.println("에궁 ㅠㅠ");
			System.out.println(e.getMessage());
		}
	}
}
```

Mysql 5이하

```
String url = "jdbc:mysql://127.0.0.1:3306/test?useSSL=true";
String driver =  "com.mysql.jdbc.Driver";
```

SELECT 문과 Statement와 ResultSe

Statement 객체

이 전 단 계 에 서 생 성 한 Connection 객 체 (con) 로 접 근 해 서 createStatement() 메소드를 호출해서 생성한다.

예	Statement stmt = con.createStatement( );

모든 작업이 끝나면 Statement 객체 역시 close() 메소드를 호출해서 데이터베이스와 연결을 해제해야 한다.

예	stmt.close( );

Statement 객체로 쿼리문 수행

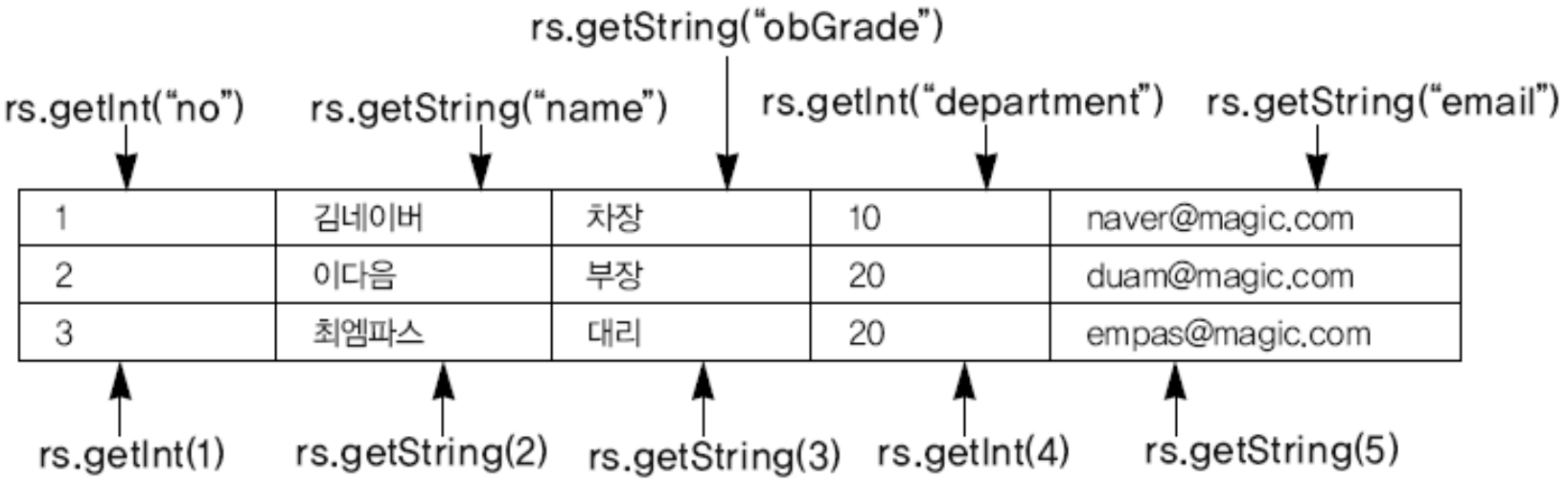
select 문과 같이 결과가 있는 쿼리문인 경우에는 executeQuery () 메소드 사용

insert, update, delete 문과 같이 내부적으로는 어떤 변화가 있지만 결과가 없는 경우에는 executeUpdate() 메소드 사용

예	String str = “select * from employee” ; ResultSet rs = stmt.executeQuery(str);

SELECT 문과 Statement와 ResultSet

ResultSet 객체는 다음과 같은 형태로 executeQuery() 메소드는 기술한 select 문의 결과 값을 저장하고 있다.



ResultSet은 다음과 같은 다양한 메소드를 제공한다.

메소드	설명
next( )	현재 행에서 한 행 앞으로 이동
previouse( )	현재 행에서 한행 뒤로 이동
first( )	현재 행에서 첫 번째 행의 위치로 이동
last( )	현재 행에서 마지막 행의 위치로 이동

SELECT 문과 Statement와 ResultSet

결과 값으로 얻어진 여러 개의 로우를 모두 출력하기 위해서는 ResultSet 객체로 레코드 단위로 이동하는 next() 메소드를 사용해야 하는데 일반적으로 다음과 같이 while 문과 함께 사용한다.

예

```
while( rs.next( ) ){
  System.out.print(rs.getInt(1) + " \t" ); //no
  System.out.print(rs.getString(2) + " \t" ); //name
  System.out.print(rs.getString(3) + " \t" ); //jobGrade
  System.out.print(rs.getInt(4) + " \t" ); //department
  System.out.print(rs.getString(5) + " \n" ); //email
}
```

SELECT 문과 Statement와 ResultSet

```
import java.sql.*;          import java.util.*;
public class StateTest {
   public static void main(String[] ar) {
try {
     Class.forName("oracle.jdbc.driver.OracleDriver");
     System.out.println("드라이버 검색 성공!");
  } catch(ClassNotFoundException e) {
     System.err.println("error = " + e);
    }
Connection conn = null;
          Statement stmt = null;
ResultSet rs = null;
String url ="jdbc:oracle:thin:@127.0.0.1:1521:xe"; //localhost,
String id = "scott";
 String pass = "tiger";
String query = "select empno,ename,job,hiredate from emp";
try {
    conn = DriverManager.getConnection(url, id, pass);
   stmt = conn.createStatement();
   rs = stmt.executeQuery(query);
```

SELECT 문과 Statement와 ResultSet

```
while(rs.next()) {
      int empno = rs.getInt(1); // rs.getInt("empno");
     String ename = rs.getString(2); // rs.getString("ename");
     String job = rs.getString(3); // rs.getString("job");
     java.sql.Date date = rs.getDate(4); // rs.getDate("hiredate");
     System.out.print(empno + "\t");
      System.out.print(ename + "\t");
     System.out.print(job + "\t");
     System.out.println(date.toString());
           }
     rs.close();
     stmt.close();
     conn.close();
         }catch(SQLException e) {
     System.err.println("error sql = " + e);
         }
       }
     }
```

SELECT 문과 Statement와 ResultSet

```
import java.sql.*;
public class Exam_03 {
    public static void main(String[] ar) {
        try {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                System.out.println("드라이버 검색 성공!");
          }catch(ClassNotFoundException e) {
                System.err.println("error = " + e);
          }
        Connection conn = null;    Statement stmt = null;
        ResultSet rs = null;
        String url =     "jdbc:oracle:thin:@127.0.0.1:1521:xe";
        String id = "scott";
        String pass = "tiger";
        String query = "select * from dept";
        try {
                conn = DriverManager.getConnection(url, id, pass);
                stmt = conn.createStatement();
                rs = stmt.executeQuery(query);
```

SELECT 문과 Statement와 ResultSet

```
        while(rs.next()) {
            int deptno = rs.getInt(1); // rs.getInt("deptno");
            String dname = rs.getString(2); // rs.getString("dname");
            String loc = rs.getString(3); // rs.getString("loc");
            System.out.print(deptno + "\t");
            System.out.print(dname + "\t");
            System.out.println(loc);
        }
        rs.close();
        stmt.close();
        conn.close();
        }catch(SQLException e) {
                System.err.println("error sql = " + e);
        }
    }
}
```

SELECT 문과 Statement와 ResultSet

```
import java.sql.*;
public class Ex02 {
   public static void main(String[] ar) {
      Connection conn=null; Statement stmt=null;  ResultSet rs = null;
      String url ="jdbc:oracle:thin:@127.0.0.1:1521:xe";
      String query ="select empno,ename,job,hiredate from emp";
      String[] tit = {"사번", "이름", "업무", " 입사일"};
      System.out.println(tit[0]+"\t"+tit[1]+"\t"+tit[2]+"\t\t"+tit[3]);
      System.out.println("-----------------------------------");
      try { Class.forName("oracle.jdbc.driver.OracleDriver");
           conn = DriverManager.getConnection(url,"scott","tiger");
           stmt = conn.createStatement();   rs = stmt.executeQuery(query);
           while(rs.next()) {
              int empno = rs.getInt(1);  String ename = rs.getString(2);
              String job = rs.getString(3);    Date date = rs.getDate(4);
              if (job.length() > 7) {
                     System.out.print(empno + "\t"+ ename + "\t");
                     System.out.println(job + "\t"+date.toString());
               }  else {  System.out.print(empno + "\t"+ ename + "\t");
                     System.out.println(job + "\t\t"+date.toString()); }
              }
             rs.close(); stmt.close(); conn.close();
       }catch(Exception e) { System.err.println("error sql = " + e); }
   }
}
```

Mysql

```
import java.sql.*;
public class MysqlConnection1 {
    public static void main(String[] args) throws SQLException {
        String driver = "com.mysql.cj.jdbc.Driver";
        String url =
        "jdbc:mysql://localhost:3306/test?useSSL=false&serverTimezone=UTC";
        Connection conn=null;
        try {
            Class.forName(driver);
            conn = DriverManager.getConnection(url,"root","mysql");
                System.out.println("MySql 연결 성공");
        }catch(Exception e) {
                System.out.println("MySql DB연결 실패 ㅠㅠ");
                System.out.println(e.getMessage());
        } finally {
                conn.close();
        }
    }
}
```

DML과 executeUpdate() 메소드

executeUpdate() 메소드

insert 문, update 문, delete 문, create 문과 같이 데이터베이스 파일의 내용을 변경하는 SQL 문을 실행할 때 사용한다.

executeUpdate() 메소드는 변경된 행(레코드)의 수를 반환해 주기에 리턴형이 Integer이다.

키보드에서 읽어온 변수에 저장된 값으로 행을 추가할 경우에는 어떻게 해야 할까?

문자열 상수는 반드시 단일 따옴표로 둘러 싸주어야 하기에 다음과 같이 복잡한 형태의 문장을 만들어야 한다.

```
String sql = "insert into employee(name, jobGrade, department” ;
sql += " , email) values (’ " + sname + "’ , ’ " + sjobGrade;
sql += "’ , " + " ndepartment + ", ’ " + semail + "’ )";
```

DML과 executeUpdate() 메소드

```
import java.sql.*;   import java.util.Scanner;
public class OracleInsert {
      public static void main(String[] args) throws SQLException {
            String driver = "oracle.jdbc.driver.OracleDriver";
            String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
            Connection conn = null; Statement stmt = null;
            Scanner sc = new Scanner(System.in);
            System.out.println("부서코드를 입력하세요");
            String deptno = sc.nextLine();
            System.out.println("부서명를 입력하세요");
            String dname = sc.nextLine();
            System.out.println("근무지를 입력하세요");
            String loc = sc.nextLine();
            String sql = String.format("insert into dept values(" +
                                    "%s,'%s','%s')", deptno, dname, loc);
            try { Class.forName(driver);
              conn = DriverManager.getConnection(url,"scott","tiger");
                      stmt = conn.createStatement();
                      stmt.executeUpdate(sql);
                      System.out.println("입력 성공");
            }catch (Exception e) { System.out.println(e.getMessage());   }
                      stmt.close();  conn.close();
            }
}
```

DML과 executeUpdate() 메소드

```
import java.sql.*;      import java.util.Scanner;
public class OracleUpdate {
        public static void main(String[] args) throws SQLException {
                String driver = "oracle.jdbc.driver.OracleDriver";
                String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
                Connection conn = null; Statement stmt = null;
                ResultSet rs = null;
                Scanner sc = new Scanner(System.in);
                System.out.println("부서코드를 입력하세요");
                String deptno = sc.nextLine();
                System.out.println("부서명를 입력하세요");
                String dname = sc.nextLine();
                String sql = String.format("update dept set dname='%s' " +
                        " where deptno = %s", dname, deptno);
                String sql2 = String.format("select * from dept where " +
                        " deptno = %s ", deptno);
                try { Class.forName(driver);
                   conn = DriverManager.getConnection(url,"scott","tiger");
                        stmt = conn.createStatement();
                        stmt.executeUpdate(sql);
                        System.out.println("수정 성공");
                        rs = stmt.executeQuery(sql2);
```

DML과 executeUpdate() 메소드

```
                if (rs.next()) {
                    int dno = rs.getInt("deptno");
                    String dna = rs.getString("dname");
                    System.out.println("no = " +  dno + "\t name = "
                                                + dna);
                }
            }catch (Exception e) {
                System.out.println(e.getMessage());   }
                stmt.close();  conn.close();
    }
}
```

DML과 executeUpdate() 메소드

```
import java.sql.*;    import java.util.Scanner;
public class OracleDelete {
        public static void main(String[] args) throws SQLException {
                String driver = "oracle.jdbc.driver.OracleDriver";
                String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
                Connection conn = null; Statement stmt = null;
                Scanner sc = new Scanner(System.in);
                System.out.println("삭제할 부서코드를 입력하세요");
                String deptno = sc.nextLine();
                String sql = String.format("delete from dept where " +
                        "deptno=%s ", deptno);
                try { Class.forName(driver);
                        conn =

        DriverManager.getConnection(url,"scott","tiger");
                        stmt = conn.createStatement();
                        stmt.executeUpdate(sql);
                        System.out.println("삭제 성공");
                }catch (Exception e) {
                        System.out.println(e.getMessage());  }
                stmt.close();  conn.close();
        }
}
```

PreparedStatement 인터페이스

PreparedStatement를 생성

PreparedStatement 객체를 생성하기 위해서는 Connection 인터페이스의 preparedStatement() 메소드를 호출한다.

예	PreparedStatement pstmt= con.prepareStatement(sql);

preparedStatement() 메소드의 인자로 사용되는 SQL 문은 다음과 같이 ? 기호를 사용해서 다음과 같이 표현할 수 있다.

예	String sql ="insert into employee values(?, ?, ?, ?, ?)"; ① ② ③ ④ ⑤

?로 지정된 인자에 값을 준다.

형식	setXXX(int 순서, 실제 데이터나 변수);

PreparedStatement 인터페이스

```
import java.sql.*;  import java.util.*;
public class PreparedOracle {
    public static void main(String[] args) throws SQLException {
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
        Connection conn = null; PreparedStatement pstmt = null;
        Scanner sc = new Scanner(System.in);
        System.out.println("부서코드를 입력하세요");
        String deptno = sc.nextLine();
        System.out.println("부서명를 입력하세요");
        String dname = sc.nextLine();
        String sql = "update dept set dname=? where deptno =?";
        try { Class.forName(driver);
                conn =  DriverManager.getConnection(url,"scott","tiger");
                pstmt = conn.prepareStatement(sql);
                pstmt.setString(1, dname);
                pstmt.setString(2, deptno);
                pstmt.executeUpdate();
                System.out.println("수정 성공");
        }catch (Exception e) {
                System.out.println(e.getMessage());  }
        pstmt.close();  conn.close();
    }
}
```

Stored Procedure와 CallableStatement

```
CREATE TABLE member (
        id VARCHAR2(12),
        passwd VARCHAR2(12),
        name VARCHAR2(12),
        age NUMBER,
        addr VARCHAR2(50),
        email VARCHAR2(30)
);
INSERT INTO member VAIUES('aaa','aaa','오정원',22,'서울시','a@a.com');
INSERT INTO member VAIUES('bbb','bbb','김개똥',21,'개똥시','b@b.com');
INSERT INTO member VAIUES('ccc','ccc','김말숙',22,'말숙시','c@c.com');
INSERT INTO member VAIUES('ddd','ddd','박말숙',22,'이상시','d@d.com');
INSERT INTO member VAIUES('eee','eee','오말숙',22,'우리시','e@e.com');
COMMIT;

CREATE OR REPLACE PROCEDURE user_insert(user_id VARCHAR2,
user_passwd VARCHAR2,user_name VARCHAR2,user_age NUMBER,
user_addr VARCHAR2,user_email VARCHAR2)
IS
BEGIN
INSERT INTO member
VALUES(user_id,user_passwd,user_name,user_age,user_addr,user_email);
END;
/
```

```
import java.sql.CallableStatement;  import java.sql.Connection;
import java.sql.DriverManager;      import java.sql.PreparedStatement;
import java.sql.ResultSet;          import java.sql.SQLException;
public class ProcedureTest {
   Connection con;
   static{
      try{  Class.forName("oracle.jdbc.driver.OracleDriver");
      }catch(ClassNotFoundException cne){ cne.printStackTrace();}
   }
   public void connect(){
     try{  con = DriverManager.getConnection(
                    "jdbc:oracle:thin:@localhost:1521:xe","scott", "tiger");
           System.out.println("Connection Success!");
      }catch(SQLException se){ se.printStackTrace(); }
   }
   public void select(){
       String sql = "SELECT * FROM member";
       PreparedStatement pstmt = null;   ResultSet rs = null;
       try{ connect();   pstmt = con.prepareStatement(sql);
            rs = pstmt.executeQuery();
            while(rs.next()){
                System.out.println("아이디 : " + rs.getString(1) + ", 비밀번호 : "
                     + rs.getString(2) + ", 이름 : " + rs.getString(3) + ", 나이 : "
                     + rs.getInt(4) + ", 주소 : " + rs.getString(5) + ", 이메일 : "
                     + rs.getString("email"));
            }
       } catch(SQLException se){se.printStackTrace();
```

```
        } finally{   try{ pstmt.close();  con.close();
                     }catch(Exception e){e.printStackTrace(); }
        }
    }
    public void insertMember(){        CallableStatement cs = null;
        try{   connect();
               String sql = "{call user_insert(?,?,?,?,?,?)}";
               cs = con.prepareCall(sql);
               cs.setString(1, "protest");    cs.setString(2, "1111");
               cs.setString(3, "김다슬");      cs.setInt(4, 19);
               cs.setString(5, "강원도");      cs.setString(6, "pro@pro.com");
               int count = cs.executeUpdate();
               if(count > 0){  System.out.println("insert success");
               }else{ System.out.println("insert fail");}
        }catch(Exception e){e.printStackTrace();
        }finally{  try{  cs.close();   con.close();
                   }catch(Exception e){e.printStackTrace(); }
        }
    }
    public static void main(String[] args) {
         ProcedureTest pt = new ProcedureTest();
         System.out.println("프로시저 호출 전 데이터");  pt.select();
         System.out.println("프로시저 호출 후 데이터");  pt.insertMember();
         pt.select();
     }
}
```

Stored Procedure와 CallableStatement

```
CREATE OR REPLACE PROCEDURE Emp_Info
  ( p_empno IN emp.empno%TYPE,
       p_ename out emp.ename%TYPE,p_sal out emp.ename%TYPE)

IS
   -- %TYPE 데이터형 변수 선언
   v_empno emp.empno%TYPE;
   v_ename emp.ename%TYPE;
   v_sal emp.sal%TYPE;

BEGIN
   -- %TYPE 데이터형 변수 사용
   SELECT empno, ename, sal INTO v_empno, v_ename, v_sal
     FROM emp   WHERE empno = p_empno ;

   p_ename := v_ename;
   p_sal := v_sal;
END;
/
```

Stored Procedure와 CallableStatement

```
import java.sql.*;    import java.util.Scanner;
public class Call03 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("사번을 입력 ?");
        int empno = sc.nextInt();
        String driver ="oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
        String sql = "{call Emp_Info(?,?,?)}";
        try {
            Class.forName(driver);
            Connection conn = DriverManager.getConnection(url,"scott","tiger");
            CallableStatement cs = conn.prepareCall(sql);
            cs.setInt(1,empno);
            cs.registerOutParameter(2, java.sql.Types.VARCHAR);
            cs.registerOutParameter(3, java.sql.Types.INTEGER);
            cs.execute();
            System.out.println("부서명 : " + cs.getString(2));
            System.out.println("급여 : " + cs.getInt(3));
        } catch(Exception e) { System.out.println(e.getMessage());}
        sc.close();
    }
}
```

Mysql procedure

```
delimiter //
create procedure division_insert (
   vdno int, vdname varchar(20),
   vphone varchar(20),  vposition varchar(20))
 begin
      insert into division
          values(vdno,vdname,vphone,vposition);
  end;
//

drop procedure division_insert;
```

실행

```
sql> call division_insert(50,'대박','010-1111-1111','서울');
```

Mysql procedure

```
public class MyPro1 {
	public static void main(String[] args) throws SQLException {
		String
url="jdbc:mysql://127.0.0.1:3306/test?useSSL=false&serverTimezone=UTC";
		String driver = "com.mysql.cj.jdbc.Driver";
		String sql = "{call dept_insert(?,?,?)}";
		Scanner sc = new Scanner(System.in);
		Connection  conn = null;  CallableStatement cs = null;
		System.out.println("부서코드");
		int deptno=Integer.parseInt(sc.nextLine());
		System.out.println("부서명"); String dname = sc.nextLine();
		System.out.println("근무지"); String loc = sc.nextLine();
		try {Class.forName(driver);
	conn=DriverManager.getConnection(url,"root","mysql");
				cs = conn.prepareCall(sql);
				cs.setInt(1, deptno); cs.setString(2, dname);
				cs.setString(3, loc);
				int result = cs.executeUpdate();
				if (result > 0) System.out.println("입력 성공 대박");
				else System.out.println("입력 실패 쪽박");
		}catch(Exception e) {System.out.println(e.getMessage());}
		sc.close(); conn.close(); cs.close();
	}
}
```

Blob데이터 입력 및 조회

- create table test (id varchar2(15) primary key,photo blob);

```
public class BlobInsert {
        public static void main(String[] args)  {
                String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
                String driver = "oracle.jdbc.driver.OracleDriver";
                String sql = "insert into img_test values ('a',?)";
                Connection conn = null; PreparedStatement pstmt = null;
                try{    File fileName  = new File("ioi.jpg");
                        int size = (int)fileName.length();
                        InputStream is = new FileInputStream(fileName);
                        Class.forName(driver);
                        conn=DriverManager.getConnection(url,"scott","tiger");
                        pstmt = conn.prepareStatement(sql);
                        pstmt.setBinaryStream(1, is, size);
                        int result = pstmt.executeUpdate();
                        if (result >0) System.out.println("입력 성공");
                        else System.out.println("입력 실패");
                        is.close();
                }catch (Exception e) { System.out.println(e.getMessage());
                }finally {
                        try {if (pstmt != null) pstmt.close();
                                if (conn  != null) conn.close();
                        }catch(Exception e) {}
                }
                                        // [byte배열을 이용하는 방법]
                                            /* byte[] buffer = new byte[fileSize];
                                               is.read(buffer);
                                               psmt.setBytes(1, buffer); */
        }
}
```

```
public class BlobSelect {
    public static void main(String[] args) {
        String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
        String driver = "oracle.jdbc.driver.OracleDriver";
        String sql = "select photo from img_test where name='a'";
        Connection conn = null; PreparedStatement pstmt = null;
        try {       File fileName = new File("kk.jpg");
                    Class.forName(driver);
                    conn=DriverManager.getConnection(url,"scott","tiger");
                    pstmt = conn.prepareStatement(sql);
                    ResultSet rs = pstmt.executeQuery();
                    if (rs.next()) {
                            Blob blob = rs.getBlob("photo");
                            InputStream is = blob.getBinaryStream();
                            FileOutputStream fos = new FileOutputStream(fileName);
                            byte[] buffer = new byte[1024];
                            int i = 0;
                            while((i=is.read(buffer)) != -1) {
                                    fos.write(buffer, 0, i);
                            }
                            is.close(); fos.close();
                            System.out.println("출력 성공");
                    }else System.out.println("데이터 없네");
        }catch(Exception e){ System.out.println(e.getMessage());
        }finally {
                    try{    if (pstmt != null) pstmt.close();
                            if (conn  != null) conn.close();
                    }catch(Exception e) {}
        }
    }
}
```

고객관리

강사 강 병준

Customer

```
package customer;
import java.util.Date;
public class Customer {
	private String id;		private String pass;
	private String email;	private String name;
	private Date reg_date;
	public Customer() {};
	public Customer(String id,String pass,String email,String
name,Date reg_date) {
		this.id=id; this.pass = pass; this.email = email;
		this.name = name; this.reg_date = reg_date;
	}
	public String toString() {
		return "아이디:"+id+",암호:"+pass+",이메일:"+email+
				",이름:"+name+", 가입일:"+reg_date;
	}
	public boolean passChk(String pass, String confirmPass) {
		return pass.equals(confirmPass);
	}
	public String getId() {		return id;		}
	public void setId(String id) { this.id = id;	}
```

Customer

```
public String getPass() {
        return pass;
}
public void setPass(String pass) {
        this.pass = pass;
}
public String getEmail() {          return email;     }
public void setEmail(String email) {
        this.email = email;
}
public String getName() {           return name;      }
public void setName(String name) {
        this.name = name;
}
public Date getReg_date() {         return reg_date;  }
public void setReg_date(Date reg_date) {
        this.reg_date = reg_date;
}
}
```

CustomerDaoImpl

```
package customer2;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.Collection;
public class CustomerDaoImpl  {
        private static  Connection conn = null;
        public Connection getConnection() {
                String url = "jdbc:oracle:thin:@127.0.0.1:1521:xe";
                String driver = "oracle.jdbc.driver.OracleDriver";
                        try {
                                Class.forName(driver);

        conn=DriverManager.getConnection(url,"scott","tiger");
                        } catch (Exception e) {
                                System.out.println(e.getMessage());
                        }
                        return conn;
        }
```

CustomerDaoImpl

```
public Customer select(String id) {
        Customer customer = null;
        PreparedStatement pstmt = null;
        ResultSet rs = null;
        String sql = "select * from customer where id=?";
        Connection conn= getConnection();
        try {
                pstmt = conn.prepareStatement(sql);
                pstmt.setString(1,id);
                rs = pstmt.executeQuery();
                if (rs.next()) {
                        customer = new Customer();
                        customer.setId(rs.getString("id"));
                        customer.setName(rs.getString("name"));
                        customer.setEmail(rs.getString("email"));
                        customer.setPass(rs.getString("pass"));

customer.setReg_date(rs.getDate("reg_date"));
                }
```

CustomerDaoImpl

```
}catch (Exception e) {
                        System.out.println(e.getMessage());
                }finally {
                        try {
                                if (rs != null) rs.close();
                                if (pstmt != null) pstmt.close();
                                if (conn != null)  conn.close();
                        } catch (Exception e) {                 }
                }
                return customer;
        }
```

CustomerDaoImpl

```
public int insert(Customer customer) {
        int result = 0;
        PreparedStatement pstmt = null;
        String sql = "insert into customer values (?,?,?,?,sysdate)";
        Connection conn= getConnection();
        try {   pstmt = conn.prepareStatement(sql);
                pstmt.setString(1, customer.getId());
                pstmt.setString(2, customer.getPass());
                pstmt.setString(3, customer.getEmail());
                pstmt.setString(4, customer.getName());
                result = pstmt.executeUpdate();
        }catch (Exception e) {
                System.out.println(e.getMessage());
        }finally {  try {
                        if (pstmt != null) pstmt.close();
                        if (conn != null)  conn.close();
                } catch (Exception e) {                 }
        }
        return result;
}
```

CustomerDaoImpl

```
public Collection<Customer> list() {
        Collection<Customer> list = new ArrayList<>();
        PreparedStatement pstmt = null;
        ResultSet rs = null;
        String sql = "select * from customer order by id";
        Connection conn= getConnection();
        try {
                pstmt = conn.prepareStatement(sql);
                rs = pstmt.executeQuery();
                while(rs.next()) {
                        Customer customer = new Customer();
                        customer.setId(rs.getString("id"));
                        customer.setName(rs.getString("name"));
                        customer.setEmail(rs.getString("email"));
                        customer.setPass(rs.getString("pass"));

customer.setReg_date(rs.getDate("reg_date"));
                        list.add(customer);
                }
```

CustomerDaoImpl

```
}catch (Exception e) {
                        System.out.println(e.getMessage());
                }finally {
                        try {
                                if (rs != null) rs.close();
                                if (pstmt != null) pstmt.close();
                                if (conn != null)  conn.close();
                        } catch (Exception e) {                 }
                }
                return list;
        }
```

CustomerDaoImpl

```
public int update(Customer customer) {
        int result = 0;
        PreparedStatement pstmt = null;
        String sql = "update customer set pass=?,email=?,name=?
where id=?";
        Connection conn= getConnection();
        try {   pstmt = conn.prepareStatement(sql);
                pstmt.setString(4, customer.getId());
                pstmt.setString(1, customer.getPass());
                pstmt.setString(2, customer.getEmail());
                pstmt.setString(3, customer.getName());
                result = pstmt.executeUpdate();
        }catch (Exception e) { System.out.println(e.getMessage());
        }finally {                      try {
                        if (pstmt != null) pstmt.close();
                        if (conn != null)  conn.close();
                } catch (Exception e) {                 }
        }
        return result;
}
```

CustomerDaoImpl

```
public int delete(String id) {
        int result = 0;
        PreparedStatement pstmt = null;
        String sql = "delete customer where id=?";
        Connection conn= getConnection();
        try {   pstmt = conn.prepareStatement(sql);
                pstmt.setString(1, id);
                result = pstmt.executeUpdate();
        }catch (Exception e) {
System.out.println(e.getMessage());
        }finally {try {
                        if (pstmt != null) pstmt.close();
                        if (conn != null)  conn.close();
                } catch (Exception e) {         }
        }
        return result;
}
}
```

CustomerServiceImpl

```
package customer;
import java.util.Collection;
public class CustomerServiceImpl {
	private static CustomerDaoImpl md = new CustomerDaoImpl();
	public int insert(Customer customer) {
		int result = 0;
		Customer ct = md.select(customer.getId());
		if (ct == null) {
			result = md.insert(customer);
		}else System.out.println("이미 있는 데이터 입니다");
		return result;
	}
	public Customer select(String id) {
		return md.select(id);
	}
	public Collection<Customer> list() {
		return md.list();
	}
```

CustomerServiceImpl

```
	public int update(Customer customer) {
		int result = 0;
		Customer ct = md.select(customer.getId());
		if (ct != null) {
			result = md.update(customer);
		}else System.out.println("없는 데이터는 수정 못합니다");
		return result;
	}
	public int delete(String id) {
		int result = 0;
		Customer customer = md.select(id);
		if (customer != null) {
			result = md.delete(id);
		}else System.out.println("없는 데이터는 삭제 못합니다");
		return result;
	}
}
```

Ex02

```
public class Ex02 {
	private static CustomerServiceImpl ms = new CustomerServiceImpl();
	private static Scanner sc;
	public static void main(String[] args) {
		sc = new Scanner(System.in);
		while(true) {
			help();
			String command = sc.nextLine();
			if (command.equals("6")) break;
			else if (command.equals("1")) {
				insert();
			}else if (command.equals("3")) {
				select();
			} else if (command.equals("5"))
				list();
			else if (command.equals("2"))
				update();
			else if (command.equals("4"))
				delete();
			else help();
		}
		sc.close();
		System.out.println("프로그램 종료");
	}
```

Ex02

```
private static void insert() {
        System.out.println("ID를 입력하시오");
        String id = sc.nextLine();
        System.out.println("이메일을 입력하시오");
        String email = sc.nextLine();
        System.out.println("이름을 입력하시오");
        String name = sc.nextLine();
        System.out.println("암호를 입력하시오");
        String pass = sc.nextLine();
        System.out.println("암호를 한번 더 입력하시오");
        String confirmPass = sc.nextLine();
        Customer customer = new Customer();
        customer.setId(id);     customer.setEmail(email);
        customer.setName(name);   customer.setPass(pass);
        customer.setReg_date(new Date());
        if (!customer.passChk(pass,confirmPass)) {
                System.out.println("암호와 암호 확인이 다릅니다");
                return;
        }
        int result = ms.insert(customer);
        if (result > 0) System.out.println("회원등록 완료");
}
```

Ex02

```
private static void update() {
		System.out.println("ID를 입력하시오");
		String id = sc.nextLine();
		System.out.println("이메일을 입력하시오");
		String email = sc.nextLine();
		System.out.println("이름을 입력하시오");
		String name = sc.nextLine();
		System.out.println("암호를 입력하시오");
		String pass = sc.nextLine();
		System.out.println("암호를 한번 더 입력하시오");
		String confirmPass = sc.nextLine();
		Customer customer = new Customer();

		customer.setId(id);     customer.setEmail(email);
		customer.setName(name);   customer.setPass(pass);
		customer.setReg_date(new Date());
		if (!customer.passChk(pass,confirmPass)) {
			System.out.println("암호와 암호 확인이 다릅니다");
			return;
		}
		int result = ms.update(customer);
		if (result > 0) System.out.println("회원수정 완료");
}
```

Ex02

```
private static void delete() {
        System.out.println("삭제할 ID를 입력하시오");
        String id = sc.nextLine();
        int result = ms.delete(id);
        if (result > 0) System.out.println("삭제 됐네");
}

private static void list() {
        Collection<Customer> list = ms.list();
        if (list == null || list.size() == 0)
                System.out.println("없는 데이터입니다");
        else
                for(Customer customer : list) {
                        System.out.println(customer);
                }
}
```

Ex02

```
	private static void select() {
		System.out.println("조회할 ID를 입력하시오");
		String id = sc.nextLine();
		Customer customer = ms.select(id);
		if (customer == null) System.out.println("없는 데이터 입니다");
		else System.out.println(customer);
	}

	private static void help() {
		System.out.println();
		System.out.println("다음 명령어 중에서 하나를 사용하세요");
		System.out.println("1. 입력");
		System.out.println("2. 수정");
		System.out.println("3. 조회");
		System.out.println("4. 삭제");
		System.out.println("5. 전체조회");
		System.out.println("6. 끝내기");
		System.out.println();
		System.out.println("명령번호");
	}
}
```