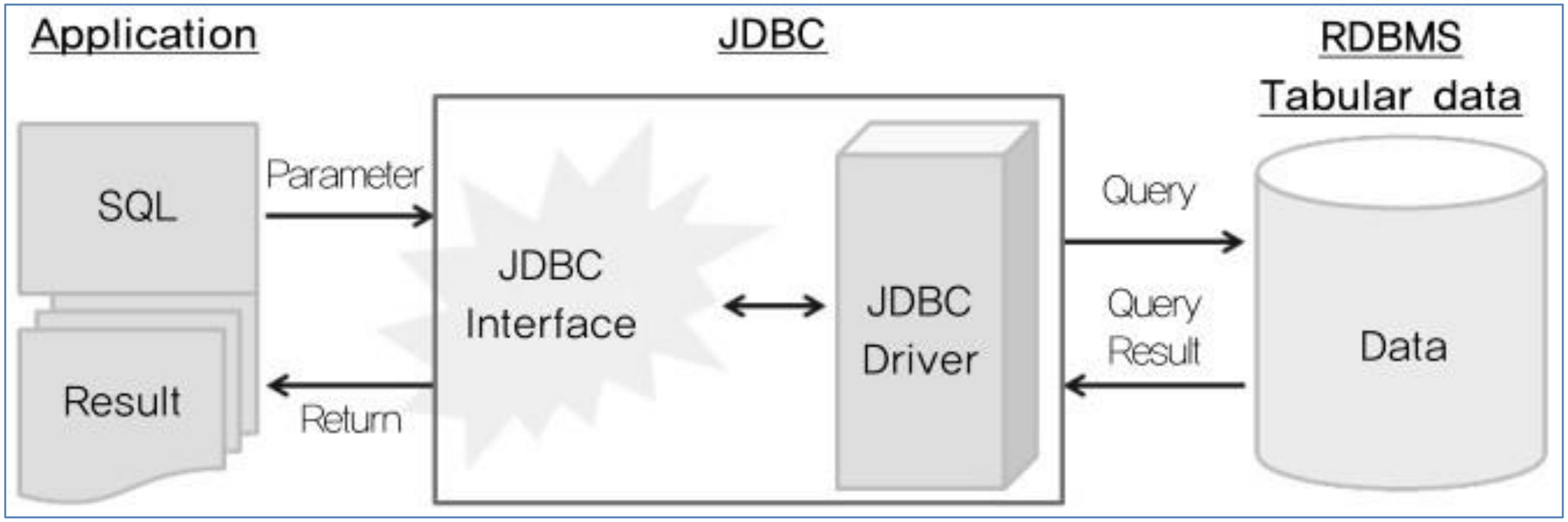
■ JDBC (Java Database Connectivity)

● 자바에서 데이터베이스에 접속할 수 있도록 하는 자바 API

● CRUD

- Create
- Read (Retrieve)
- Update
- Delete (Destroy)



■ JDBC (Java Database Connectivity)

● 데이터베이스 작업 흐름

1. 드라이버 로딩

```
Class.forName("org.mariadb.jdbc.Driver");
Class.forName("oracle.jdbc.driver.OracleDriver");
Class.forName("com.mysql.jdbc.Driver");
Class.forName("org.h2.Driver");
```

2. 데이터베이스 연결

```
Connection con = DriverManager.getConnection(url, id, pw);
String url = "jdbc:mariadb://localhost:3306/java";
String url = "jdbc:oracle:thin:@localhost:1521:xe";
String url = "jdbc:mysql://localhost:3306/java";
String url = "jdbc:h2:tcp://localhost/~/testdb";
```

■ JDBC (Java Database Connectivity)

● 데이터베이스 작업 흐름

3. SQL (Query) 실행 준비

```
String sql = "select * from emp";

String sql = "insert into (id, pw) values (?, ?);

PreparedStatement stmt = conn.prepareStatement(sql);
stmt.setString(1, "userid");
stmt.setString(2, "password");
```

4. SQL (Query) 실행

- insert, update, delete : stmt.executeUpdate();
- select : stmt.executeQuery();

5. (조회시) 조회결과 처리

```
while(rs.next()) {
    String empno = rs.getString(“empno");
    String ename = rs.getString(“ename");
    String job = rs.getString(“job");
}
```

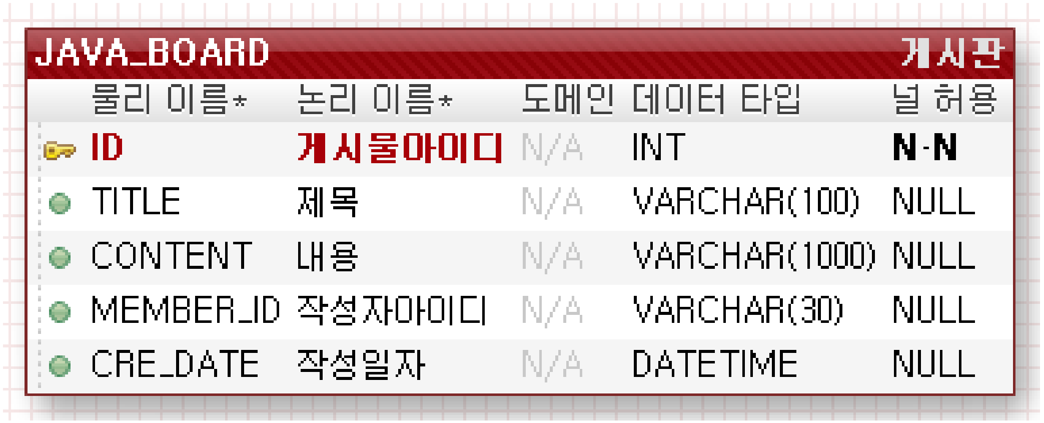
■ JDBC 기본 코드

ch18.JdbcExam

```
private static final String DRIVER_CLASS_NAME = "org.mariadb.jdbc.Driver";
private static final String URL = "jdbc:mariadb://15.164.153.191:3306/java";
private static final String USERNAME = "java";
private static final String PASSWORD = "1234";

public static void main(String[] args) throws ClassNotFoundException, SQLException {
  Class.forName(DRIVER_CLASS_NAME);
  Connection con = DriverManager.getConnection(URL, USERNAME, PASSWORD);

  String sql = "SELECT empno, ename, job FROM emp";

  PreparedStatement stmt = con.prepareStatement(sql);
  ResultSet rs = stmt.executeQuery();

  while(rs.next()) {
    int empno = rs.getInt("empno");
    String ename = rs.getString("ename");
    String job = rs.getString("job");

    System.out.printf(
        "empno : %s, ename : %s, job : %s\n", empno, ename, job);
  }
  rs.close();
  stmt.close();
  con.close();
}
```

■ JDBC CRUD 연습

● ERD (Entity Relationship Diagram)



JAVA_BOARD 게시판

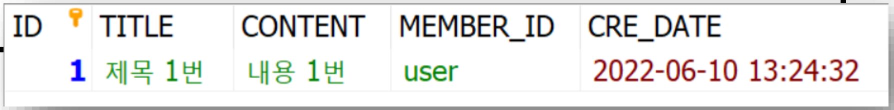
물리 이름*	논리 이름*	도메인	데이터 타입	널 허용
ID	게시물아이디	N/A	INT	N-N
TITLE	제목	N/A	VARCHAR(100)	NULL
CONTENT	내용	N/A	VARCHAR(1000)	NULL
MEMBER_ID	작성자아이디	N/A	VARCHAR(30)	NULL
CRE_DATE	작성일자	N/A	DATETIME	NULL

```
CREATE TABLE JAVA_BOARD (
    ID          INT             NOT NULL  COMMENT '게시물아이디',
    TITLE       VARCHAR(100)    NULL      COMMENT '제목',
    CONTENT     VARCHAR(1000)   NULL      COMMENT '내용',
    MEMBER_ID   VARCHAR(30)     NULL      COMMENT '작성자아이디',
    CRE_DATE    DATETIME        NULL      COMMENT '작성일자',
    PRIMARY KEY (ID)
)
```

■ JDBC CRUD 연습 - Create

ch18.JdbcCreate

```
Class.forName(DRIVER_CLASS_NAME);
Connection con = DriverManager.getConnection(URL, USERNAME, PASSWORD);

String sql = "INSERT INTO java_board " +
             " (id, title, content, member_id, cre_date) " +
             " VALUES (?, ?, ?, ?, now())";

PreparedStatement stmt = con.prepareStatement(sql);
stmt.setInt(1, 1);
stmt.setString(2, "제목 1번");
stmt.setString(3, "내용 1번");
stmt.setString(4, "user");

int result = stmt.executeUpdate();

stmt.close();
con.close();
```

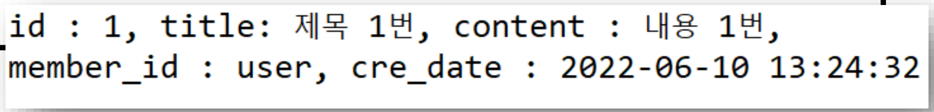


ID	TITLE	CONTENT	MEMBER_ID	CRE_DATE
1	제목 1번	내용 1번	user	2022-06-10 13:24:32

■ JDBC CRUD 연습 - Read

ch18.JdbcRead

```
Class.forName(DRIVER_CLASS_NAME);
Connection con = DriverManager.getConnection(URL, USERNAME, PASSWORD);

String sql = "SELECT id, title, content, member_id, cre_date " +
             "  FROM java_board";

PreparedStatement stmt = con.prepareStatement(sql);
ResultSet rs = stmt.executeQuery();
while(rs.next()) {
  int id = rs.getInt("id");
  String title = rs.getString("title");
  String content = rs.getString("content");
  String member_id = rs.getString("member_id");
  String cre_date = rs.getString("cre_date");

  System.out.printf(
      "id : %s, title: %s, content : %s,\n" +
      "member_id : %s, cre_date : %s\n",
      id, title, content, member_id, cre_date);
}
rs.close();
stmt.close();
con.close();
```

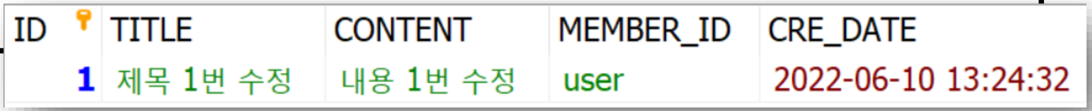


```
id : 1, title: 제목 1번, content : 내용 1번,
member_id : user, cre_date : 2022-06-10 13:24:32
```

■ JDBC CRUD 연습 - Update

ch18.JdbcUpdate

```
Class.forName(DRIVER_CLASS_NAME);
Connection con = DriverManager.getConnection(URL, USERNAME, PASSWORD);

String sql = "UPDATE java_board SET " +
             " title = ?, content = ? WHERE id = ?";

PreparedStatement stmt = con.prepareStatement(sql);
stmt.setString(1, "제목 1번 수정");
stmt.setString(2, "내용 1번 수정");
stmt.setInt(3, 1);

int result = stmt.executeUpdate();

stmt.close();
con.close();
```

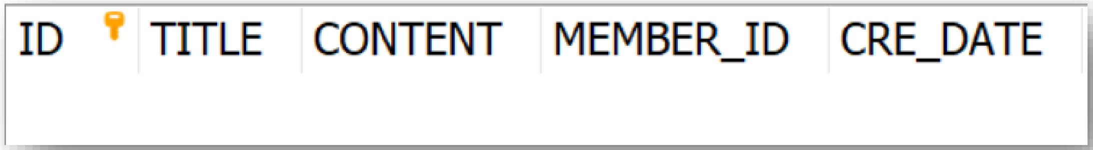


ID	TITLE	CONTENT	MEMBER_ID	CRE_DATE
1	제목 1번 수정	내용 1번 수정	user	2022-06-10 13:24:32

■ JDBC CRUD 연습 - Delete

ch18.JdbcDelete

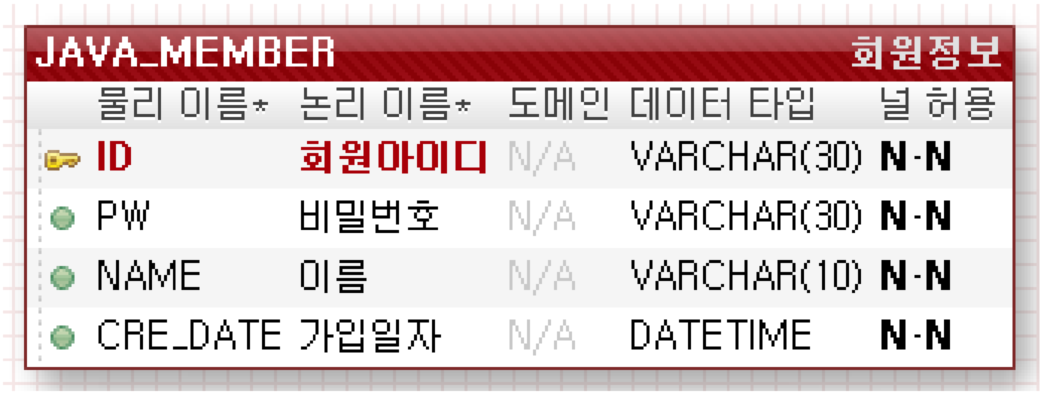
```
Class.forName(DRIVER_CLASS_NAME);
Connection con = DriverManager.getConnection(URL, USERNAME, PASSWORD);

String sql = "DELETE FROM java_board WHERE id = ?";

PreparedStatement stmt = con.prepareStatement(sql);
stmt.setInt(1, 1);

int result = stmt.executeUpdate();

stmt.close();
con.close();
```



ID	TITLE	CONTENT	MEMBER_ID	CRE_DATE

■ 회원가입 / 로그인

● ERD (Entity Relationship Diagram)



JAVA_MEMBER 회원정보

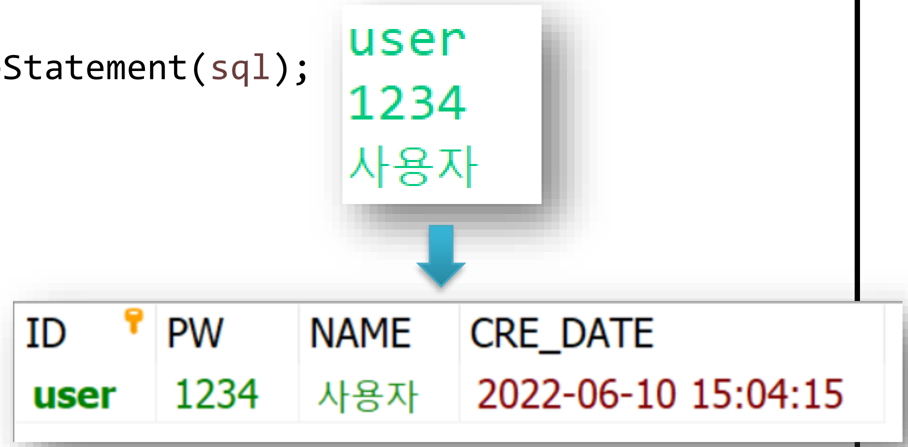
	물리 이름*	논리 이름*	도메인	데이터 타입	널 허용
🔑	ID	회원아이디	N/A	VARCHAR(30)	N-N
●	PW	비밀번호	N/A	VARCHAR(30)	N-N
●	NAME	이름	N/A	VARCHAR(10)	N-N
●	CRE_DATE	가입일자	N/A	DATETIME	N-N

```
CREATE TABLE JAVA_MEMBER (
    ID        VARCHAR(30) NOT NULL COMMENT '회원아이디',
    PW         VARCHAR(30) NOT NULL COMMENT '비밀번호',
    NAME      VARCHAR(10) NOT NULL COMMENT '이름',
    CRE_DATE DATETIME    NOT NULL COMMENT '가입일자',
    PRIMARY KEY (ID)
)
```

■ 회원가입

ch18.SignUp

```
Scanner scanId = new Scanner(System.in);
Scanner scanPw = new Scanner(System.in);
Scanner scanName = new Scanner(System.in);

String id = scanId.nextLine();
String pw = scanPw.nextLine();
String name = scanName.nextLine();

scanId.close();  scanPw.close();  scanName.close();

Class.forName(DRIVER_CLASS_NAME);
Connection con = DriverManager.getConnection(URL, USERNAME, PASSWORD);

String sql = "INSERT INTO java_member (id, pw, name, cre_date) " +
             " VALUES (?, ?, ?, now())";

PreparedStatement stmt = con.prepareStatement(sql);
stmt.setString(1, id);
stmt.setString(2, pw);
stmt.setString(3, name);

int result = stmt.executeUpdate();

stmt.close();
con.close();
```



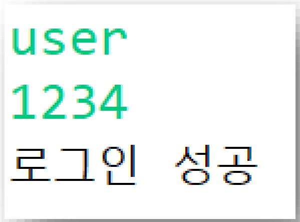
```
user
1234
사용자
```

ID	PW	NAME	CRE_DATE
user	1234	사용자	2022-06-10 15:04:15

■ 로그인

ch18.SignIn

```
Scanner scanId = new Scanner(System.in);
Scanner scanPw = new Scanner(System.in);

String id = scanId.nextLine();
String pw = scanPw.nextLine();

scanId.close();  scanPw.close();

Class.forName(DRIVER_CLASS_NAME);
Connection con = DriverManager.getConnection(URL, USERNAME, PASSWORD);

String sql = "SELECT * FROM java_member WHERE id = ? AND pw = ?";

PreparedStatement stmt = con.prepareStatement(sql);
stmt.setString(1, id);
stmt.setString(2, pw);
ResultSet rs = stmt.executeQuery();

String result = "";
if(rs.next()) result = "로그인 성공";
else result = "로그인 실패";

System.out.println(result);

rs.close(); stmt.close(); con.close();
```



```
user
1234
로그인 성공
```