

0326 JDBC

JDBC
JAVA - DATE BASC - Connectivity
Connectivity : 연결(성)



자바에서 DB에 접근하려면 . 드라이버가 필요하다 이 드라이버는
각각 DBMS에 맞는 라이브러리를 받아와 Referenced Libraries으로 등록 후 가져와 사용한다
오라클은 OJDBC 메이븐 레파지토리에서 다운로드
SQL관련 자바 클래스는 java.sql으로 기본적으로 제공한다

Class.forName("oracle.jdbc.driver.OracleDriver");

- 어떻게 동작하는가: 드라이버와 같이 인스턴스가 별도로 관리하지 않는 클래스의 경우 **Static**블록을 통해 미리 생성하고 관리한다.
- 따라서 **class.forName()**은 드라이버의 생성과 초기화를 발생시킨다.

자바에서 DB로 연결할 드라이버를 라이브러리로 가져와 등록 했다면

- **자바와 DB를 연결할 스트림을 생성한다**
 1. 이때 스트림을 연결하기 위해 필요한 변수들을 먼저 만드는데
 1. 드라이브의 주소를 **Class.forName("드라이버 주소")**을 이용해 어디에 드라이버가 있는 지 인지시켜준다.
 2. **DriverManager.getConnection()**을 통해 드라이버를 얻어온다
 3. 이때 **getConnection**의 괄호 안에는
 1. 드라이버의 위치 + DB의 IP 주소 + DB의 포트번호 + DB의 이름
 2. DB사용자명 , 비밀번호 가 필요하다



위처럼 연결시 OracleDBMS와 연결될 통로 정보가 **String conn**변수에 저장되었다

conn변수는 **db와 연결정보를 가진 변수인 것이다**
이때 보내는 질의문은 **String** 형식의 변수를 **sql**의 구문의 형태를 띄어 전달한다 **이때 구문 안에 ';' (세미콜론을 보낼 경우 오류)



- 이제 `creatStatement`를 통해 SQL 질의문을 전달하고 `ResultSet`을 가져온다.
- 이때 sql 질의문을 먼저 작성 후 `Statement`객체에 적재한다



`createStatement` 클래스를 통해 `Statement`객체를 생성한다.

이 순간

`stmt` 변수에는 DB의 연결정보를 가지고 질의문을 보내고 가져올 수 있게 하게 된다

우선적으로 보낸 sql 질의문이 수행 성공 시 `resultSet`을 가지고 돌아오게되는데

행의 끝이 어디인지 알지 못하기때문에 `while`문을 통해 다음 '행'이 있는지 없는지를 통해 끝을 구분한다

`while(resultSet변수명.next()) { } :`

- `resultSet변수명.next()`: 참조하는 `resultSet`의 다음 행이 있을 경우 `True`를 반환한다

이렇게 가져온 값을 각각 가져온 데이터 형식에 맞는 변수에 대입 시켜 원하는 동작을 수행한다

- DB의 `int` 는 `number`등 java와 자료형이 다르기때문에 알맞게 대입시켜야 한다.ㄴ



이렇게 모든 통신을 수행하는데

이 모든 상황에서 발생할 수 있는 예외를 예외처리후

Finally 후 사용한 자원을 close 해야한다

```
try{ }  
catch(SQLException e){ } : DB와 연결 관련 예외의 최상위  
catch(ClassNotFoundException e){ } : 클래스 파일을 찾지못할 경우 발생하는 예외  
사용한 자원을 반환 할 때에는 finally 구문안에서 한번 더 try로 예외처리해야한다  
finally{  
    try{  
        NullPointerException을 방지하기 위한 if문을 작성한다  
        if(rs != null) rs.close();  
        if(stmt != null) stmt.close();  
        if(conn != null) conn.close();  
    }  
    닫을 때는 객체 생성 순서의 역순으로 수행한다.  
    }  
    catch(SQLException e){ }  
}
```