



데이터베이스(jdbc)



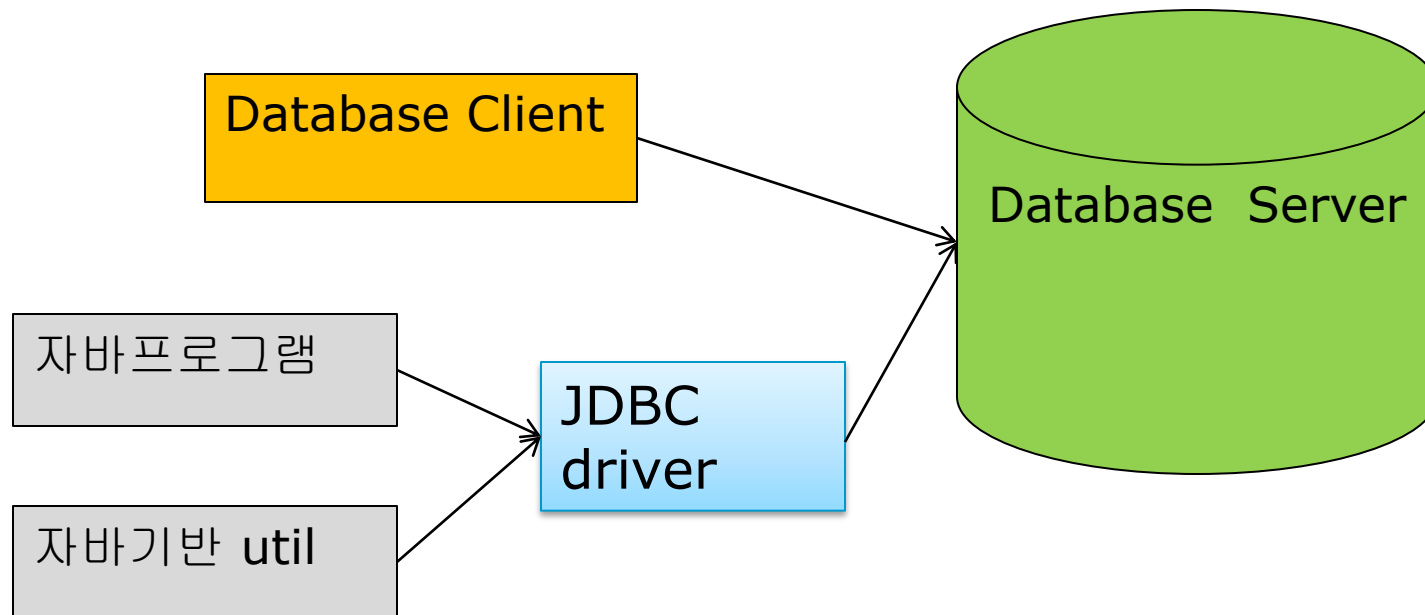
- **jdbc**의 개념과 자바에서 활용되는 **DB** 처리 프로세스에서 명확하게 파악한다.
 - **VO(value object), DAO(data access object), DTO(data tranfer object)**를 구분해서 **DB**처리시 활용 할 수 있도록 한다.
 - 데이터베이스의 내용을 조회, 등록, 수정, 삭제 시, 처리할 모듈을 구성할 수 있다.
-



생각해봅시다 :

- **java**에서 데이터베이스 서버는 어떻게 연결이 될까?
 - 데이터를 조회할 때, 필요로 하는 요소들이 어떤 것이 있을까?
 - **java**에서 데이터베이스와 연결 후, 자원은 어떻게 관리를 할까?
 - **DB** 연결 시, 발생한 예외 내용은 어떤 것이 있을까?
 - **SQL**을 통해 동적 **query**은 어떻게 처리하여야 할까?
-

- JDBC(**J**ava **D**ata**B**ase **C**onnectivity)란 자바로 데이터베이스에 접근할 수 있게 하는 프로그램 **API**를 말한다.





jdbc를 이용해서 DB 접근 :

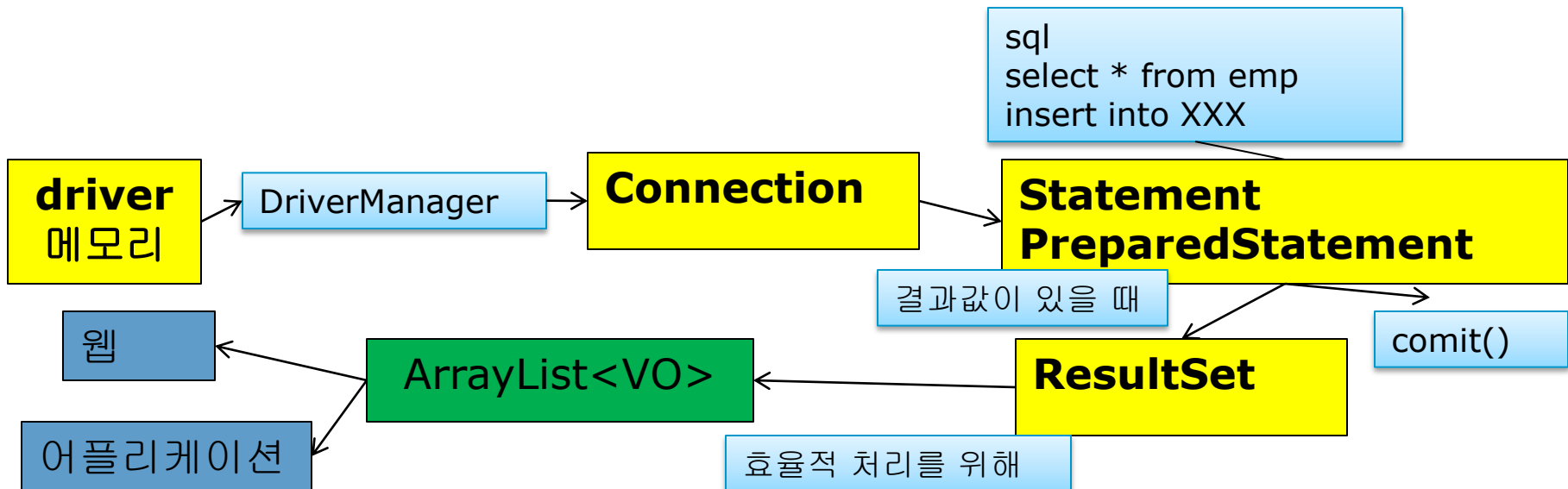
- driver 다운로드 및 lib에 위치 지정
 - 오라클 서버
 - C:\oracle\app\oracle\product\11.2.0\server\jdbc\lib\ojdbc6.jar
 - Web(jsp기반) 프로그램
 - \WebContent\WEB-INF\lib 에 해당 lib파일을 위치
 - 드라이버 로드
 - 클래스를 jvm 메모리에 로딩
 - lib에 있는 package명.클래스
 - **oracle.jdbc.driver.OracleDriver.class**
 - Class.forName(driver명)
-



jdbc를 이용해서 DB 접근 :

■ DB와 연결

- url : jdbc driver(thin):IP정보:port정보:SID
 - jdbc:oracle:thin:@localhost:1521:xe
- Connection **con** =
DriverManager.getConnection(url, id, pass);





jdbc를 이용해서 DB 접근 :

- 대화하기(Statement, PreparedStatement)
 - sql문 작성
 - select * from emp
 - 연결된 객체(con)과 Statement 연결
 - Statement **stmt**=**con**.createSatement();
 - 실제 sql문 처리..결과이 있을 때, ResultSet
 - ResultSet **rs** = **stmt**.executeQuery(sql문)
- 결과값 처리(ResultSet) – select 문일 때.
 - next() : 데이터 있을 때까지 호출(다음 row에 데이터가 있는지 여부boolean)
 - while(rs.next())
 - getXXX("sql의 title명");
 - rs.getString("name")



jdbc를 이용해서 DB 접근 :

- 결과값이 없는 sql문
 - con.commit() 호출 : 등록, 수정, 삭제 확정
 - con.rollback() 호출 : 에러
- ResultSet → ArrayList<VO> 변환 처리.
 - sql의 list와 맞는 VO 객체 생성(java)
 - ArrayList<Emp> list = new ArrayList<Emp>();
 - Emp e=null;
 - while(rs.next()){ // rs.next() : row단위로 반복
 - e = new Emp();
 - e.setEmpno(rs.getInt("empno"));
 - ...
 - ...
 - list.add(e);
 - }



정리 및 확인하기 :



감사합니다 !
