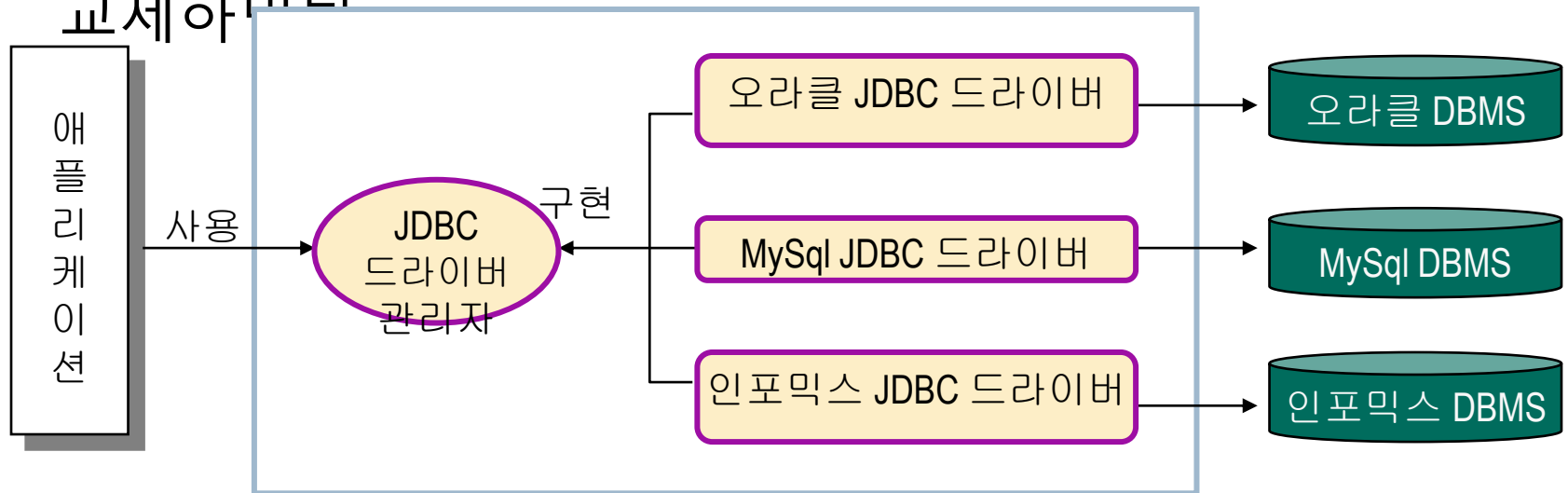


---

# 7. DataBase 연동

# JDBC

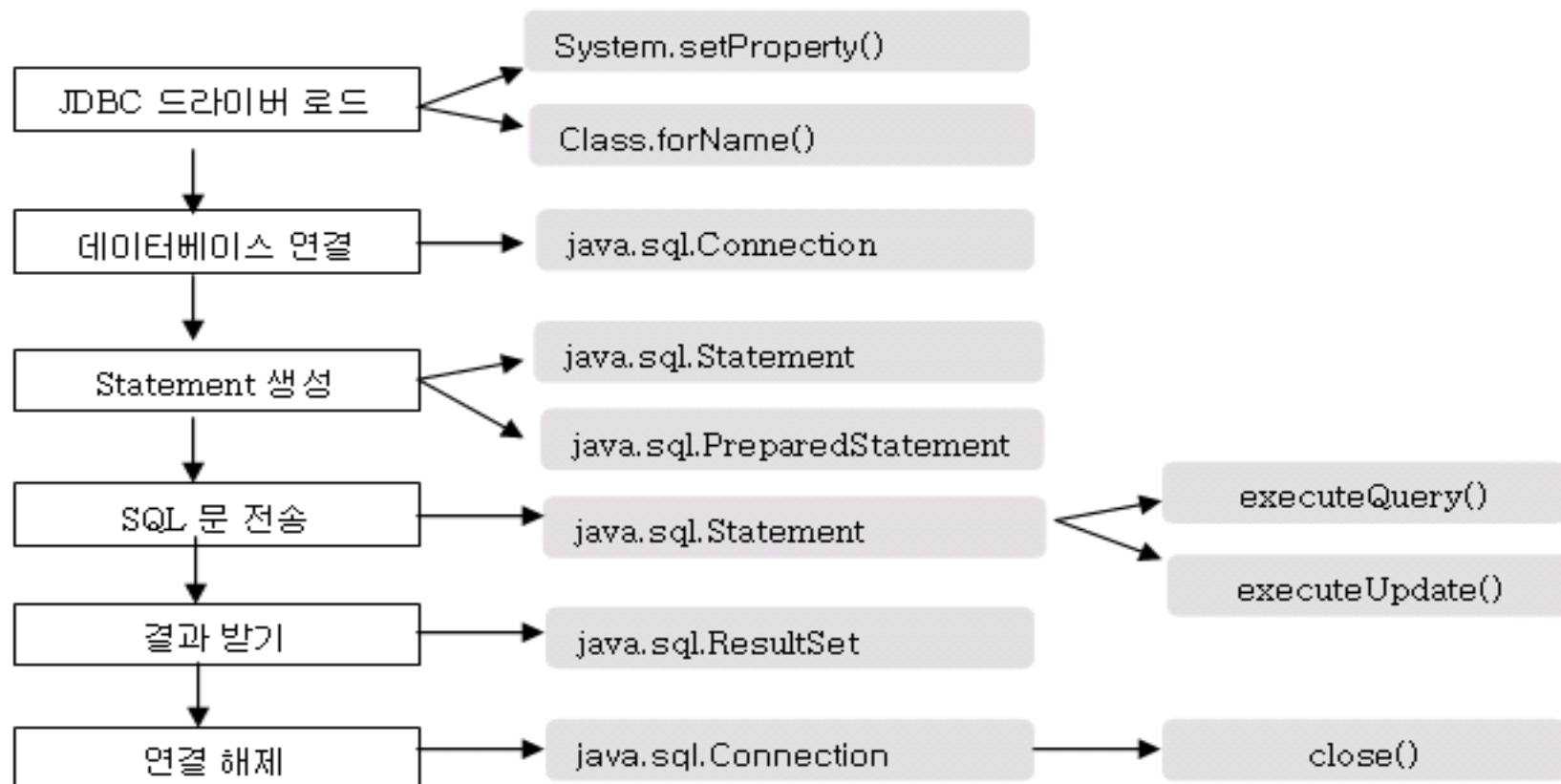
- ▶ 자바애플리케이션에서 표준화된 데이터베이스 접근 제공.
- ▶ 각 데이터베이스 접속에 대한 상세한 정보를 알 필요 없음.
- ▶ 개발된 애플리케이션에서 DB 변경시 JDBC 드라이버만 교체하면 된다.



# JDBC 작업단계

## JDBC 프로그래밍 단계

## 사용 클래스



# JDBC 작업단계

```
Select * from xxx where city='서울';
```

데이터베이스

①

② 검색결과 100건

③ rs

ResultSet

④ rs.next()  
rs.next()

1

2

...

name

email

city

홍길동

[test@test.net](mailto:test@test.net)

서울

아무개

[kk@kk.net](mailto:kk@kk.net)

서울

길동이

[tt@tt.net](mailto:tt@tt.net)

부산

몰라요

[hh@hh.net](mailto:hh@hh.net)

인천

.....

....

# ConnectionPool

---

## ▶ 등장배경

- 사용하지 않는 커넥션으로 인한 데이터베이스 자원낭비.
- 애플리케이션에서 새로운 DB 연결을 만드는것은 시스템 부하의 한 요소.
- 대규모 시스템일수록 커넥션 관리는 중요함.

## ▶ ConnectionPool

- 일정 수 만큼의 Connection을 pool로 유지.
- 일정수준이상의 커넥션을 유지하기 때문에 최적의 성능을 보장.
- 사용하지 않는 커넥션의 자동관리 기능.
- 최대 접속에 대한 제한 기능.
- 여러 애플리케이션에 서비스 가능

# ConnectionPool

---

## ▶ ConnectionPool 관리방법

- 애플리케이션 서버가 startup 될때 일정수의 커넥션을 미리 생성한다.
- 애플리케이션의 요청에 따라 생성된 커넥션 객체를 전달한다.
- 일정수 이상의 커넥션이 사용되어 지면 새로운 커넥션을 만든다.
- 사용되지 않는 커넥션은 정리하고 최소수의 커넥션을 유지한다.

## ▶ 예전에는 별도로 구현된 커넥션 풀링 소스를 사용.

## ▶ JDBC 2.0 이상에서는 javax.sql.DataSource 인터페이스를 제공.

## ▶ 애플리케이션 서버에서 제공하는 DataSource 구현체를 권장함

# JNDI 와 DataSource

---

- ▶ JNDI(Java Naming and Directory Interface)
  - 표준화된 디렉토리 및 이름 서비스에 대한 자바 인터페이스
  - JNDI를 통해 객체참조가 가능함
  - 애플리케이션에서 DataSource를 JNDI를 통해 제공

# JNDI 와 DataSource

Property	Value	설명
JNDI Name	jdbc/Ora8	네이밍 서비스에서 사용할 이름을 입력 일반적으로 DataSource에는jdbc 접두어 사용
Data Source URL	jdbc:oracle:thin:@xxx.xx.xxx.xxx:1521:OSID	데이터베이스 접속을 위한 URL 정보를 입력 각 데이터베이스별 정의된 URL을 입력
JDBC Driver Class	oracle.jdbc.driver.OracleDriver	JDBC 클래스 로딩을 위한 클래스 정보
User Name	scott	데이터베이스 계정
Password	tiger	데이터베이스 계정의 비밀번호
Max. Active Connections	4	커넥션 풀에서 관리할 최대 동시 연결 수 지정 데이터베이스 사용규모에 따라 적절히 설정 커넥션풀은 Max 값 이상의 연결은 만들지 못함
Max Idle Connections	2	최대로 유지할 유휴 연결의 개수 설정값 이상의 커넥션은 자동으로 정리 →항상유지할 빈 연결 개수로 생각할 수 있음
Max Wait for Connection	50000	밀리세컨드 값(기본값은 50 초) 데이터베이스 연결을 구하기 위한 대기 시간 →시간이 지나치게 짧으면 클라이언트에 예외를 발생시킬 수 있음.



# JNDI 와 DataSource

---

```
import javax.naming.*;
```

```
.. 중략
```

```
Context initContext = new InitialContext();
```

```
Context envContext = (Context)initContext.lookup("java:/comp/env");
```

```
DataSource ds = (DataSource)envContext.lookup("jdbc/Ora8");
```

```
Connection conn = ds.getConnection();
```

```
.. 중략
```

```
conn.close();
```