**JDBC**

|  |  |
| --- | --- |
| 1 | 다음 설명 중 틀린 것은?  (1)자바언어로 데이터베이스를 연동하는 기술을 JDBC라 하며, java.sql 패키지에 관련 클래스들이 모여있다.  (2) JDBC를 이용하면 데이터베이스 기종에 따라 소스코드가 달라지는 일은 발생되지 않는다.  (3) 각 데이터베이스 제품에 알맞는 드라이버 제작 및 배포는 Sun 사가 담당한다.// **땡!! DB사에서 제공**  (4) 드라이버를 포함한 모든 jar 파일은 원칙상 시스템의 환경변수에 등록이 되어야 해당 클래스들을 사용할 수 있다. |
| 2 | 다음 설명 중 틀린 것은?  (1) SQLPlus를 통해 오라클에 접속할 경우 JDBC를 이용한 접속으로 볼 수 있다. **// 땡**  (2) SQLPlus가 데이터베이스 클라이언트 접속 프로그램이라면, JDBC는 자바언어를 통해 DB를 제어하기  위한 API로 볼 수 있다.  (3)SQLPlus 에서 sqlplus scott/tiger 와 같은 명령으로 접속을 시도하는 행위는 JDBC의**DriverManager** 객체의 getConnection() 메서드 호출에 비유할 수 있다. // 접속담당!!  (4)SQLPlus로 접속이 성공되면 개발자는 각종 쿼리문을 실행시킬 수 있는데, 이 쿼리행위를 JDBC로수행하려면 **PreparedStatement** 인터페이스를 이용해야 한다.  (5)PreparedStatement 객체로 쿼리 수행 시, 쿼리문의 종류가 select 라면 반환되는 결과는 표형태의 데이터이므로 **ResultSet** 인터페이스로 처리해야 한다. |
| 3 | 다음 설명 중 틀린 것은?  (1) **Connection** 객체는 데이터베이스 접속을 시도하는 객체이다. // **땡!** **사후객체!** 접속정보를 갖고있다!  (2) **PreparedStatement** 객체는 쿼리문을 수행하는 객체이다  (3) **ResultSet** 은 select문 수행결과로 반환되는 테이블을 담는 객체이다.  (4) **DriverManager**는 접속을 시도하는 객체이다. |
| 4 | 다음은 무엇에 대한 설명인가?  자바개발자들은 데이터베이스 제품에 상관없이 접속정보는 Connection 객체를, 쿼리수행은 PreparedStatement  를, select문 수행결과 집합은 ResultSet을 이용하고 있으며 메서드 명칭도 제품변경에 의해 변하지 않는다.  즉 어떠한 데이터베이스 제품을 사용하더라도 사용되는 객체 및 메서드에는 변함이 없으므로 제품의 변경에 따른 소스코드의 변경이 발생하지 않는다.  위와 같은 개발이 가능한 이유는 Sun에서 Connection, PreparedStatement, ResultSet 객체등을 몸체없는 인터페이스로 정의만 해두고, 이 객체를 완성할 의무는 데이터베이스 벤더사에 위임하였기 때문인데, 데이터베이스를 제작한 벤더는 Sun에서 명시한 인터페이스 객체들의 몸체를 자신들만의 기술로 완성하되, 메서드 명칭이나 객체명칭을 그대로 유지함으로써 자바 개발자들의 소스코드의 변경을 초래하지 않는다. 이렇게 완성된 객체집합은 jar로 묶어서 배포되는데, 이러한 클래스 집합을 가리켜 무엇이라 하는가? ( JDBC 드라이버 ) |
| 5 | 다음 설명 중 틀린 것은?  (1) 수행하고자 하는 문장이 insert 문이면 PreparedStatement 객체의 executeUpdate() 메서드를 수행한다.  (2) 수행하고자 하는 문장이 update 문이면 PreparedStatement 객체의 executeQuery() 메서드를 수행한다. **// 땡**  (3) 수행하고자 하는 문장이 delete 문이면 PreparedStatement 객체의 exeucteUpdate() 메서드를 수행한다.  (4) 수행하고자 하는 문장이 select문이면 PreparedStatement 객체의 executeQuery() 메서드를 수행한다. |
| 5 | 다음 설명 중 틀린 것은?  (1) insert 문을 executeUpdate() 메서드로 수행할 경우, 결과값은 해당 쿼리문에 의해 반영된 레코드수를  반환하므로 성공할 경우 1이 반환된다.  (2) DML은 executeUpdate() 메서드로 수행시킬 수 있다  (3) select 문은 executeQuery() 메서드로 수행시킬 수 있다.  (4) update와 delete 문을 executeUpdate() 메서드로 수행할 경우, 성공하면 언제나 1이 반환된다. |
| 6 | 다음은 무엇에 대한 설명인가?  데이터베이스도 컴퓨터에서 실행되는 소프트웨어이기 때문에 개발자가 쿼리문을 수행시킬때, 이 문자열은 사실상 컴퓨터가 이해할 수 없는 문자열이므로 0과 1의 이진수로 변환되어져야 한다. 또 이진수로 변환되기 전에 각종 문법검사, 객체권한 검사, 파싱등이 발생하므로 모든 쿼리문 수행시마다 이런 작업이 진행될 경우 성능에 문제가 발생할 수 있게 된다. 이러한 문제점을 해결하기 위해 쿼리문 수행시 예전에 수행되었던 문장의 경우 각종 검사 및 파싱 , 컴파일등 생략할 수 있는 방법이 필요하게 되었고, 쿼리문내에서 변경되는 데이터에 대해 새로운 문장으로 인식처리되지 않도록 해주는 변수를 가리켜 ( 바인드변수 )라 한다. |
| 7 | 다음 설명 중 틀린 것은?  (1) 데이터베이스를 다루기위한 최적의 스윙 컴포넌트는 JTable 이다.  (2) JTable은 디자인, 즉 보여주는 역할만 수행할 뿐 실제로 어떤 데이터를 몇건을 보여줄지 여부등은 TableModel 이 담당할 수 있도록 지원하고 있다.  (3) TableModel 은 인터페이스이므로 원칙상 구현 클래스를 개발자가 정의하여야 하지만, sun에서는 개발자대신  이미 TableModel 인터페이스를 구현한 클래스를 제공해주고 있고 그 클래스는 AbstractTableModel이다. |