

JDBC

▼ LIKE 연산자

ename like 'A%' : A로 시작하는

enmae like 'A_' : A로 시작하고 뒤에 임의의 문자가 하나 있는 단어만

ename like '%A' : A로 끝나는

ename like '%A%' : A를 포함하는

▼ CONCAT 연산자

여러 문자열을 하나로 합쳐준다.

concat(format(sal, 1), '달러') fsal → sal을 소숫점 1자리수까지 출력하며 1,000단위, + 뒤에 달러를 붙여서 출력 → 5,000.0달러

▼ IS, IF

IS NULL : null인 경우

IF NULL : null일 경우

▼ Subquery

조회하고자하는 내용이 한 테이블에 모두 있는 경우 JOIN 보다는 Subquery가 더 유용하다

```
select ename, job, sal, deptno from emp where deptno =(select deptno from dept where dname = 'SALES');
```

▼ blob

mediumblob : db에 이미지를 보관할 때 지정하는 타입

▼ Statement 사용

```
Statement stmt = conn.createStatement();
```

```
stmt.executeUpdate("INSERT INTO student VALUES('둘리', 100) "); → 둘리와 100을 student 컬럼에 삽입한다.
```

```
int delNum = stmt.executeUpdate("delete from student where name = '둘리'"); → 이름이 둘리인 데이터를 컬럼에서 지운다.
```

```
int updateNum = stmt.executeUpdate ("update student set score = 100 where name = '둘리'"); → set 컬럼명 = '값' : 해당 컬럼의 값을 변경한다.
```

executeQuery()는 SELECT 명령의 실행 결과를 참조하는 ResultSet 객체를 반환하여 추출된 결과 집합을 행단위로 옮겨가며 각 열의 값을 정해진 메서드로 추출하지만 executeUpdate()는 int 값을 return한다. return 되는 값의 의미는 주어진 SQL 명령에 의해 테이블에서 변화된 행의 개수를 의미한다.

executeQuery()	ResultSet		
executeUpdate()	int	INSERT	1
		DELETE	삭제된 행의 개수
		UPDATE	수정된 행의 개수
		DDL	0

▼ PreparedStatement 사용

설정하고 싶은 데이터의 타입에 맞춰서 메서드를 호출할 수 있다.

동일한 sql명령의 값만 바뀌서 한다 / 동적으로 sql명령을 구성한다 → PreparedStatement가 유용하다.

```
PreparedStatement pstmt = conn.prepareStatement("insert into student values (?, ?)");
pstmt.setString(1, name);
pstmt.setInt(2, score);
pstmt.executeUpdate();
// PreparedStatement에서 executeUpdate를 할 때 ()안에 sql을 준다면 error.

PreparedStatement pstmt = conn.prepareStatement("delete from student where name = ?")
pstmt.setString(1, name);
int delNum = pstmt.executeUpdate();

PreparedStatement pstmt = conn.prepareStatement("update student set score = ? where name = ?");
pstmt.setInt(1, score);
pstmt.setString(2, name);
int updateNum = pstmt.executeUpdate();
```