

동적으로 생성된 SQL문에서 데이터베이스 스키마 재구성하기

제9회 소프트웨어무결점연구센터 워크샵
한양대학교
프로그래밍언어연구실
조소희, 도경구

예제

```

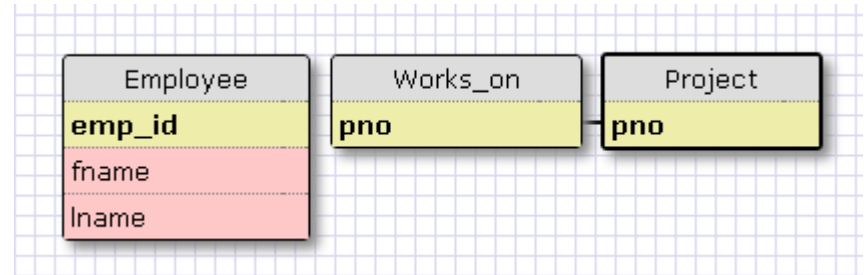
public String getEmpWorkIn(String location) throws SQLException {
    String name = "";
    String sql = "SELECT fname, lname \n"
        + "FROM Employee \n";
    if (!"".equals(location)) {
        sql += "WHERE emp_id IN \n"
            + "    (SELECT emp_id \n"
            + "    FROM Works_on JOIN Project \n"
            + "    USING(pno) \n"
            + "    WHERE plocation = ?)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, deptName);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            name += rs.getString("fname")
                + " "
                + rs.getString("lname");
        }
    }
    return name;
}
  
```

```

SELECT fname, lname
FROM Employee
WHERE emp_id IN
    (SELECT emp_id
    FROM Works_on JOIN Project
    Using(pno)
    WHERE plocation = ?)
  
```

SQL문과 예상 스키마

```
SELECT fname, lname
FROM Employee
WHERE emp_id IN
  (SELECT emp_id
   FROM Works_on JOIN Project
    USING(pno)
   WHERE plocation = ?)
```



동적으로 생성된 SQL문에서 데이터베이스 스키마 재구성하기

데이터베이스 스키마 재구성을 위한 속성

컬럼모드(ColumnMode)

- 접근 범위 및 관계 조건 여부에 따라 분류

ColumnMode = { USUAL | JCOLUMN | JCOND | COND }

- USUAL
 - 기본 모드, from절에 명시된 모든 테이블의 컬럼명을 찾을 수 있음
 - 예> `SELECT id FROM table1 ORDER BY id;`
- JCOLUMN
 - 개발자가 직접 명시한 조인 대상 컬럼명을 찾을 수 있는 위치
 - 예> `SELECT * FROM table1 JOIN table2 USING (id);`
- JCOND
 - 조인 조건에 명시된 컬럼명을 찾을 수 있는 위치
 - 예> `SELECT * FROM table1 JOIN table2 ON table1.id=table2.id;`
- COND
 - 조인 조건으로 사용될 가능성이 있는 위치
 - 예> `SELECT * FROM table1, table2 where table1.id=table2.id;`

하위질의문 중첩 구조

- 하위질의문 중첩 구조에 따라 테이블과 컬럼의 관계유추

```
SELECT fname, lname  
FROM Employee  
WHERE emp_id IN  
  (SELECT emp_id  
   FROM Works_on JOIN Project  
    USING(pno)  
   WHERE plocation = ?)
```

```
SELECT fname, lname  
FROM Employee e,  
  (SELECT emp_id, plocation  
   FROM Works_on JOIN Project  
    USING(pno)) p  
WHERE e.emp_id = p.emp_id  
AND plocation = ?
```

스칼라, 행 서브쿼리

테이블 서브쿼리

하위질의문 중첩 구조

- 하위질의문 중첩 구조에 따라 테이블과 컬럼의 관계유추

0

```
SELECT fname, lname
FROM Employee
WHERE emp_id IN
```

```
(SELECT emp_id
FROM Works_on JOIN Project
    USING(pno)
WHERE plocation = ?)
```

1, R

스칼라, 행 서브쿼리

0

```
SELECT fname, lname
FROM Employee e,
```

```
(SELECT emp_id, plocation
FROM Works_on JOIN Project
    USING(pno)) p
```

```
WHERE e.emp_id = p.emp_id
AND plocation = ?
```

1, T

테이블 서브쿼리

조인 중첩 구조

- 조인 중첩 구조에 따라 테이블과 컬럼의 관계유추

```
SELECT fname, lname  
FROM Employee  
WHERE emp_id IN  
  (SELECT emp_id  
   FROM Works_on JOIN Project  
    USING(pno)  
   WHERE plocation = ?)
```

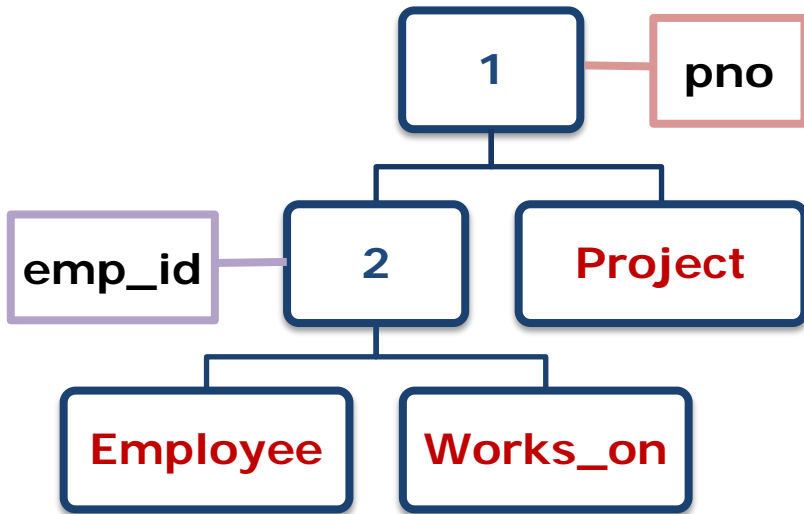
기존 질의문

```
SELECT fname, lname  
FROM (Employee JOIN Works_on  
      USING(emp_id))  
      JOIN Project  
      USING(pno)  
WHERE plocation = ?
```

중첩 조인 사용

조인 중첩 구조

- 조인 중첩 구조에 따라 테이블과 컬럼의 관계유추



```

SELECT fname, lname [2]
FROM (Employee JOIN Works_on
      USING(emp_id))
     JOIN Project
     USING(pno)
WHERE plocation = ? [1]
    
```

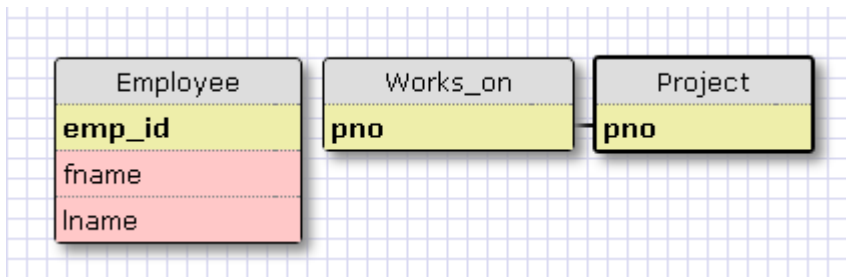
중첩 조인 사용

동적으로 생성된 SQL문에서 데이터베이스 스키마 재구성하기

재구성된 데이터베이스 스키마

SQL문에서 재구성된 DB스키마

```
SELECT fname, lname
FROM Employee
WHERE emp_id IN
    (SELECT emp_id
     FROM Works_on JOIN Project
     USING(pno)
     WHERE plocation = ?)
```

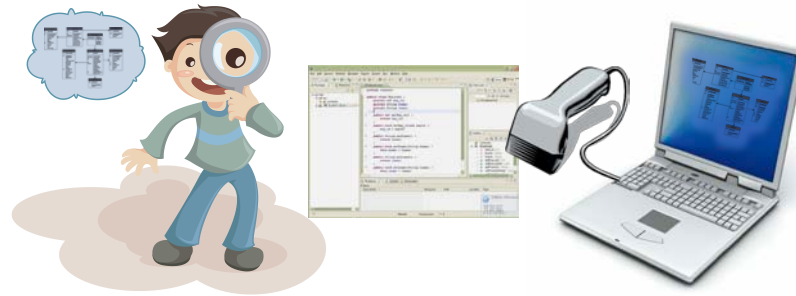


#	컬럼이름	컬럼모드	하위질의	조인중첩
1	fname	USUAL	0	0
2	lname	USUAL	0	0
3	emp_id	COND	0	0
4	emp_id	USUAL	1,R	0
5	pno	JCOLUMN	1,R	1
6	plocation	COND	1,R	0

테이블이름	하위질의	조인중첩
Employees	0	0
Works_on	1,R	1
Project	1,R	1

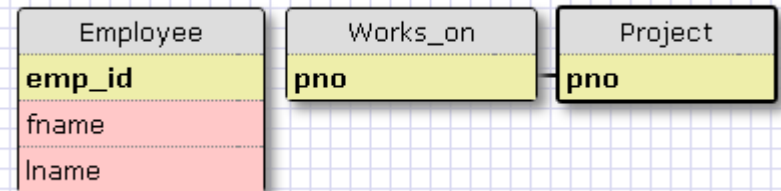
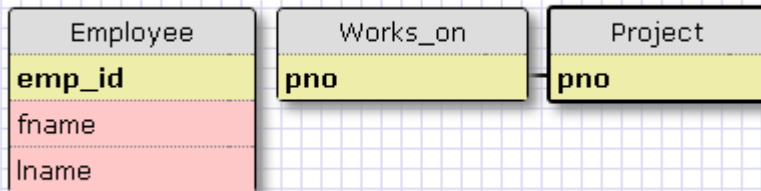
컬럼정보 1	컬럼정보 2
#5	#5

예상 스키마와의 비교



사람

컴퓨터



ROSAEC center

Research On Software Analysis for Error-free Computing

소프트웨어 무결점 연구센터 NRF ERC

