

# DATABASE Homework\_2 MySQL Report

2018\_06\_07

## 14 MySQL Queries and Results in Company Database

20132651 이 성 재



국민대학교

## ~ Q1 ~

성과 이름이 *John B. Smith*인 사원의  
생년월일과 주소를 검색하라

The screenshot shows the SQL Developer interface with a query window titled 'Query 3'. The query is: `-- Q1. 성과 이름이 John B. Smith 인 사원의 생년월일과 주소를 검색하라`  
`SELECT Bdate , Address`  
`FROM employee`  
`WHERE Fname = 'John' AND Minit = 'B' AND Lname = 'Smith';`

The result grid shows the following data:

Bdate	Address
1965-01-09	731 Fondren, Houston, TX

## ~ Q2 ~

부서번호가 5인 부서에 근무하는 모든 사원의  
모든 애트리뷰트 값을 검색하라

The screenshot shows the SQL Developer interface with a query window titled 'Query 3'. The query is: `-- Q2. 부서번호가 5인 부서에 근무하는 모든 사원의 모든 애트리뷰트 값을 검색하라`  
`SELECT *`  
`FROM employee`  
`WHERE Dno = 5;`

The result grid shows the following data:

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000.00	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000.00	987654321	5
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## ~ Q3 ~

*Research* 부서에 근무하는 모든 사원의  
성과 이름, 주소를 검색하라

Query 3 results for Research department employees:

```
-- Q3. Research 부서에 근무하는 모든 사원의 성과 이름, 주소를 검색하라
SELECT Fname, Minit, Lname, Address
FROM employee EMP JOIN department DEP ON EMP.Dno = DEP.Dnumber
WHERE DEP.Dname = 'Research';
```

Fname	Minit	Lname	Address
John	B	Smith	731 Fondren, Houston, TX
Franklin	T	Wong	638 Voss, Houston, TX
Joyce	A	English	5631 Rice, Houston, TX
Ramesh	K	Narayan	975 Fire Oak, Humble, TX

## ~ Q4 ~

*Stafford* 에 위치한 모든 프로젝트에 대해  
프로젝트 번호, 담당부서 번호, 부서 관리자의 성,  
주소, 생년월일을 검색하시오

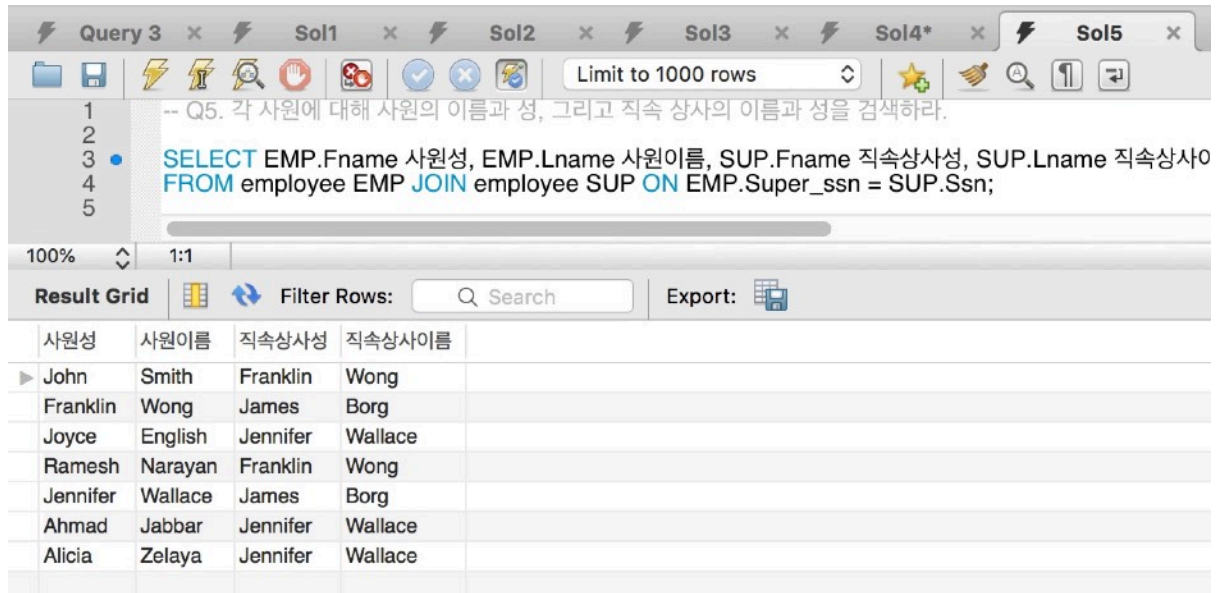
Query 4 results for projects in Stafford:

```
-- Q4. Stafford 에 위치한 모든 프로젝트에 대해서, 프로젝트 번호, 담당부서 번호, 부서 관리자의 성, 주소
SELECT Pnumber, Dnum, Fname, Address, Bdate
FROM (
  SELECT Pnumber, Dnum, Mgr_ssn
  FROM (
    SELECT Pnumber, Dnum, Plocation
    FROM project
    WHERE Plocation = 'Stafford'
  ) PRJ JOIN department DEPT ON PRJ.Dnum = DEPT.Dnumber
) MGR JOIN employee EMP ON MGR.Mgr_ssn = EMP.Ssn;
```

Pnumber	Dnum	Fname	Address	Bdate
10	4	Jennifer	291 Berry, Bellaire, TX	1941-06-20
30	4	Jennifer	291 Berry, Bellaire, TX	1941-06-20

## ~ Q5 ~

각 사원에 대해 사원의 이름과 성,  
그리고 직속 상사의 이름과 성을 검색하라



The screenshot shows a SQL query editor with a query window titled 'Query 3' and a solution window titled 'Sol5'. The query is as follows:

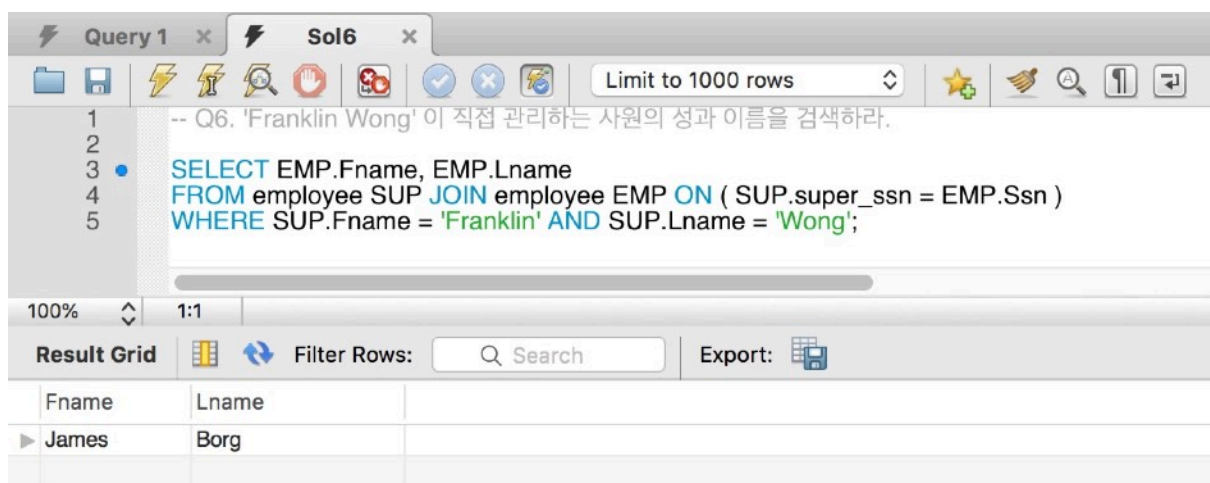
```
-- Q5. 각 사원에 대해 사원의 이름과 성, 그리고 직속 상사의 이름과 성을 검색하라.  
SELECT EMP.Fname 사원성, EMP.Lname 사원이름, SUP.Fname 직속상사성, SUP.Lname 직속상사이름  
FROM employee EMP JOIN employee SUP ON EMP.Super_ssn = SUP.Ssn;
```

The result grid shows the following data:

사원성	사원이름	직속상사성	직속상사이름
John	Smith	Franklin	Wong
Franklin	Wong	James	Borg
Joyce	English	Jennifer	Wallace
Ramesh	Narayan	Franklin	Wong
Jennifer	Wallace	James	Borg
Ahmad	Jabbar	Jennifer	Wallace
Alicia	Zelaya	Jennifer	Wallace

## ~ Q6 ~

*Franklin Wong* 이 직접 관리하는 사원의  
성과 이름을 검색하라.



The screenshot shows a SQL query editor with a query window titled 'Query 1' and a solution window titled 'Sol6'. The query is as follows:

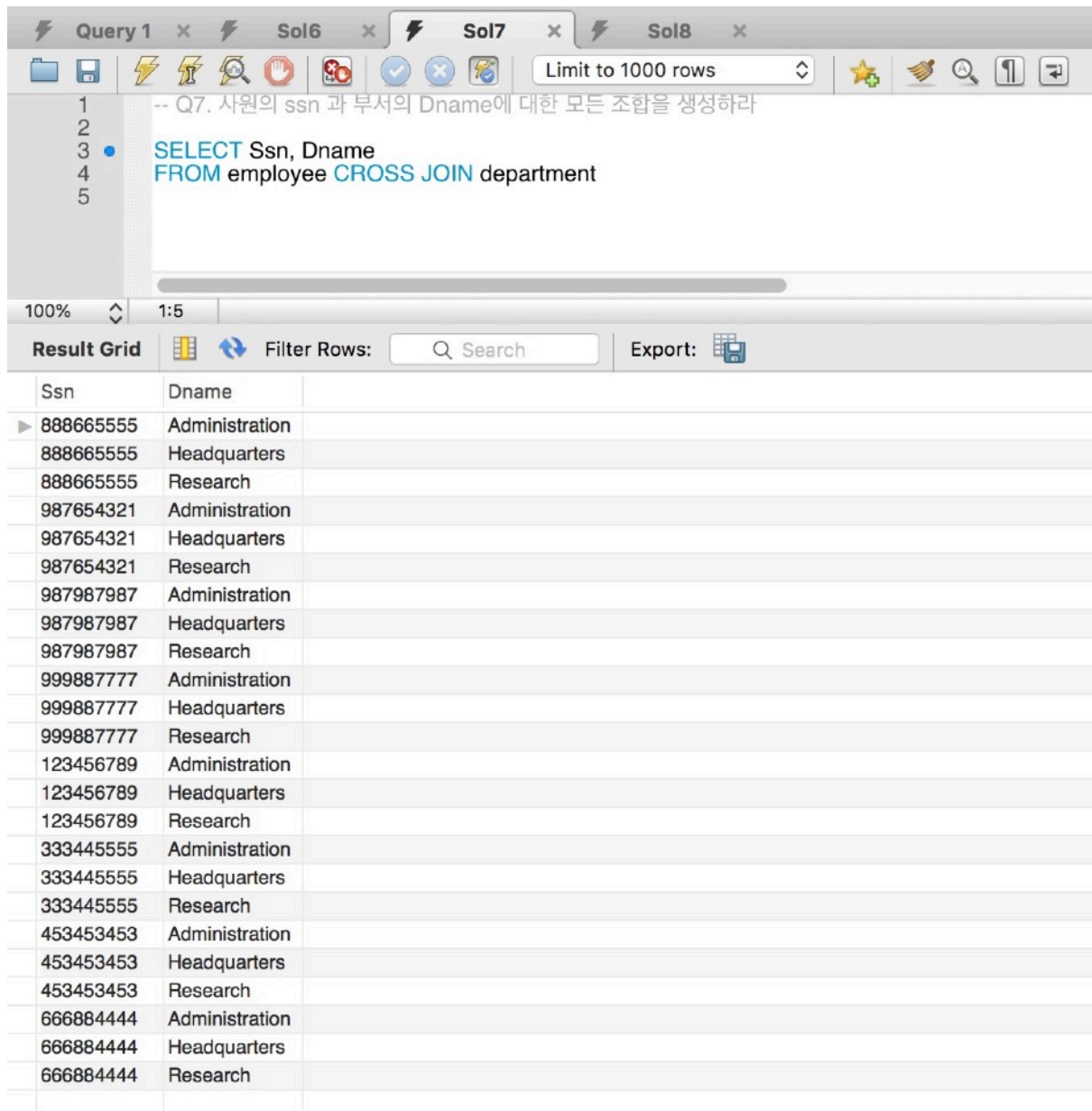
```
-- Q6. 'Franklin Wong' 이 직접 관리하는 사원의 성과 이름을 검색하라.  
SELECT EMP.Fname, EMP.Lname  
FROM employee SUP JOIN employee EMP ON ( SUP.super_ssn = EMP.Ssn )  
WHERE SUP.Fname = 'Franklin' AND SUP.Lname = 'Wong';
```

The result grid shows the following data:

Fname	Lname
James	Borg

~ Q7 ~

사원의 SSN과 부서의 DNAME에 대한  
모든 조합을 생성하라



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor displays the following SQL statement:

```
-- Q7. 사원의 ssn 과 부서의 Dname에 대한 모든 조합을 생성하라
SELECT Ssn, Dname
FROM employee CROSS JOIN department
```

The result grid shows the output of the query, displaying all combinations of employee SSNs and department names. The columns are labeled 'Ssn' and 'Dname'. The data is as follows:

Ssn	Dname
888665555	Administration
888665555	Headquarters
888665555	Research
987654321	Administration
987654321	Headquarters
987654321	Research
987987987	Administration
987987987	Headquarters
987987987	Research
999887777	Administration
999887777	Headquarters
999887777	Research
123456789	Administration
123456789	Headquarters
123456789	Research
333445555	Administration
333445555	Headquarters
333445555	Research
453453453	Administration
453453453	Headquarters
453453453	Research
666884444	Administration
666884444	Headquarters
666884444	Research



## ~ Q8 ~

일반 직원 혹은 프로젝트 관리자 중에서  
성이 *Smith* 인 사원을 포함하는 모든 프로젝트 번호의  
리스트를 검색하라

Query 1 x Sol6 x Sol7 x Sol8 x

Limit to 1000 rows

-- Q8. 일반 직원 혹은 프로젝트 관리자 중에서 성이 Smith 인 사원을 포함하는 모든 프로젝트 번호의 리스트를 검색하라

```
1 SELECT Pnumber
2 FROM employee E JOIN project P ON E.Dno = P.Dnum
3 WHERE E.Lname = 'Smith'
4
5 UNION
6
7 SELECT Pnumber
8 FROM employee E JOIN department D ON ( E.Ssn = D.Mgr_ssn ) JOIN project P ON D.Dnumber = P.Dnum
9 WHERE E.Lname = 'Smith';
10
11
12
```

100% 1:12

Result Grid Filter Rows: Search Export:

Pnumber
1
2
3

## ~ Q9 ~

주소가 *Houston, TX* 인 모든 사원의  
성과 이름을 검색하라

Query 1 x Sol6 x Sol7 x Sol8 x Sol9 x

Limit to 1000 rows

-- Q9. 주소가 Houston, TX 인 모든 사원의 이름과 성을 검색하라

```
1 SELECT Fname, Lname
2 FROM employee
3 WHERE address LIKE '%Houston, TX'
```

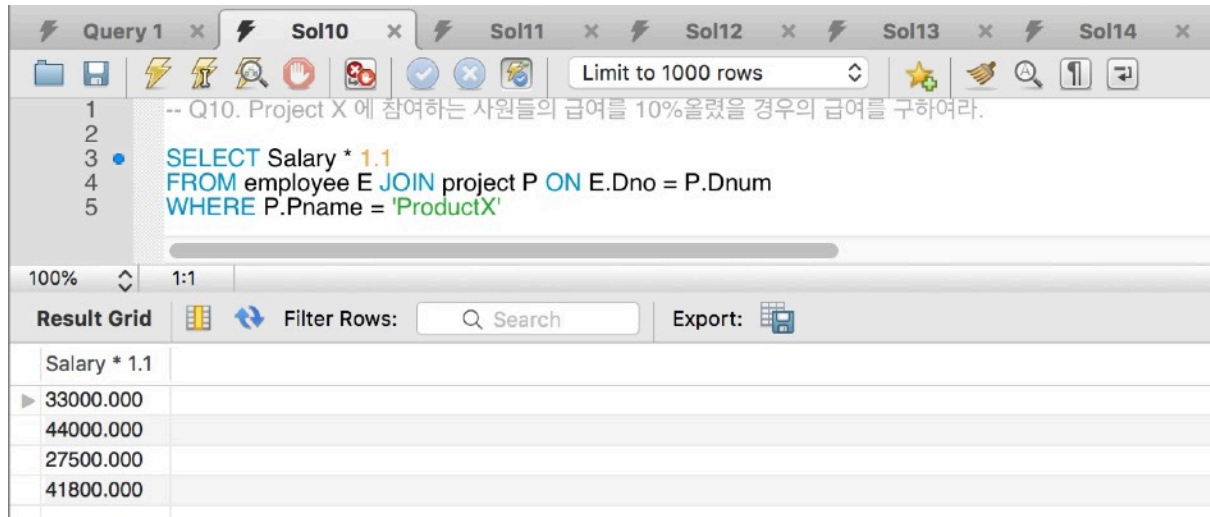
100% 1:1

Result Grid Filter Rows: Search Export:

Fname	Lname
John	Smith
Franklin	Wong
Joyce	English
James	Borg
Ahmad	Jabbar

## ~ Q10 ~

*ProductX* 프로젝트에 참여하는 직원들의 급여를  
10% 올렸을 경우의 급여를 구하여라



The screenshot shows a SQL IDE interface with multiple tabs. The active tab is 'Sol10'. The query editor contains the following SQL statement:

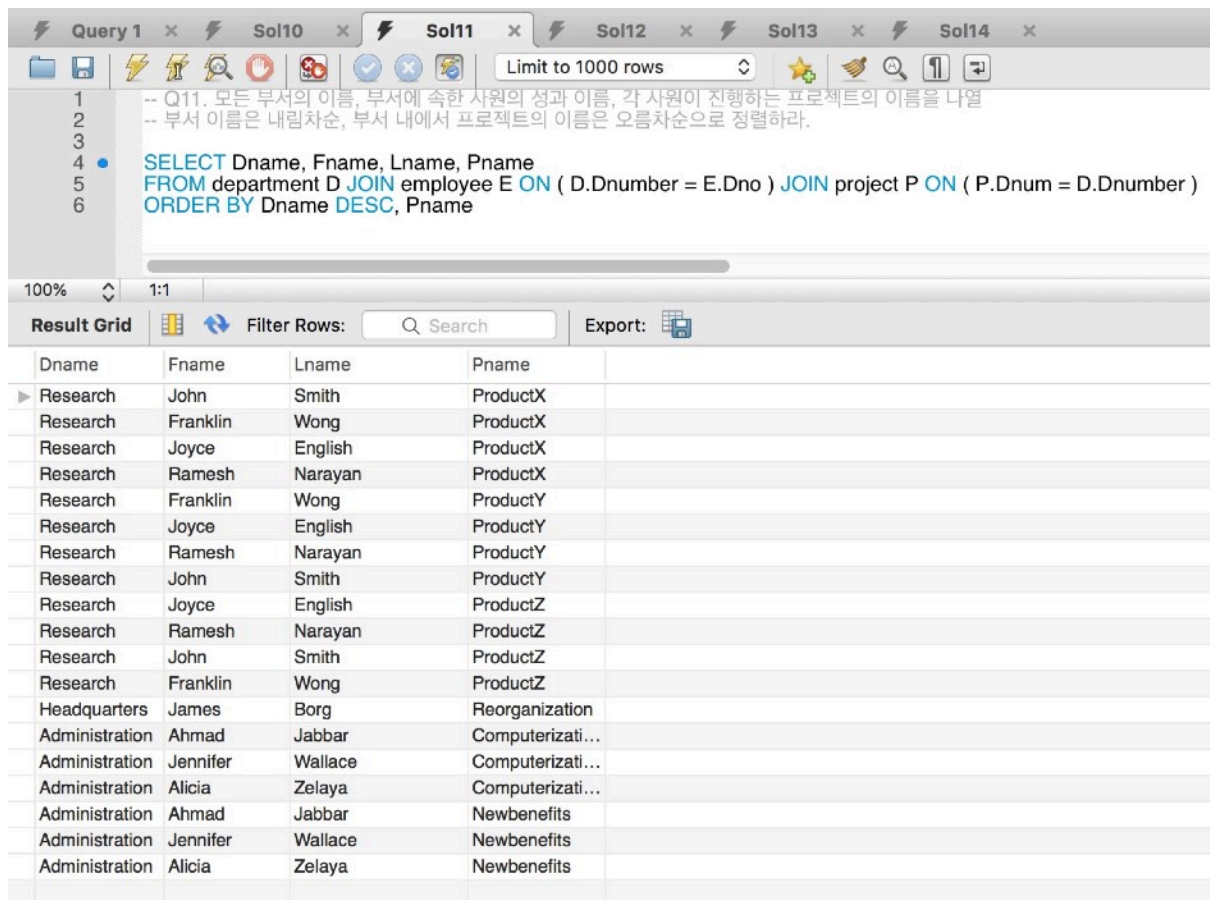
```
-- Q10. Project X에 참여하는 직원들의 급여를 10%올렸을 경우의 급여를 구하여라.  
SELECT Salary * 1.1  
FROM employee E JOIN project P ON E.Dno = P.Dnum  
WHERE P.Pname = 'ProductX'
```

The results are displayed in a table with the following data:

Salary * 1.1
33000.000
44000.000
27500.000
41800.000

## ~ Q11 ~

모든 부서의 이름, 부서에 속한 사원의 성과 이름,  
각 사원이 진행하는 프로젝트의 이름을 나열하라  
이 때 부서 이름은 내림차순, 부서 내에서  
프로젝트의 이름은 오름차순으로 정렬하라.



The screenshot shows a SQL query editor with a query window titled 'Sol11'. The query is as follows:

```
-- Q11. 모든 부서의 이름, 부서에 속한 사원의 성과 이름, 각 사원이 진행하는 프로젝트의 이름을 나열
-- 부서 이름은 내림차순, 부서 내에서 프로젝트의 이름은 오름차순으로 정렬하라.

SELECT Dname, Fname, Lname, Pname
FROM department D JOIN employee E ON ( D.Dnumber = E.Dno ) JOIN project P ON ( P.Dnum = D.Dnumber )
ORDER BY Dname DESC, Pname
```

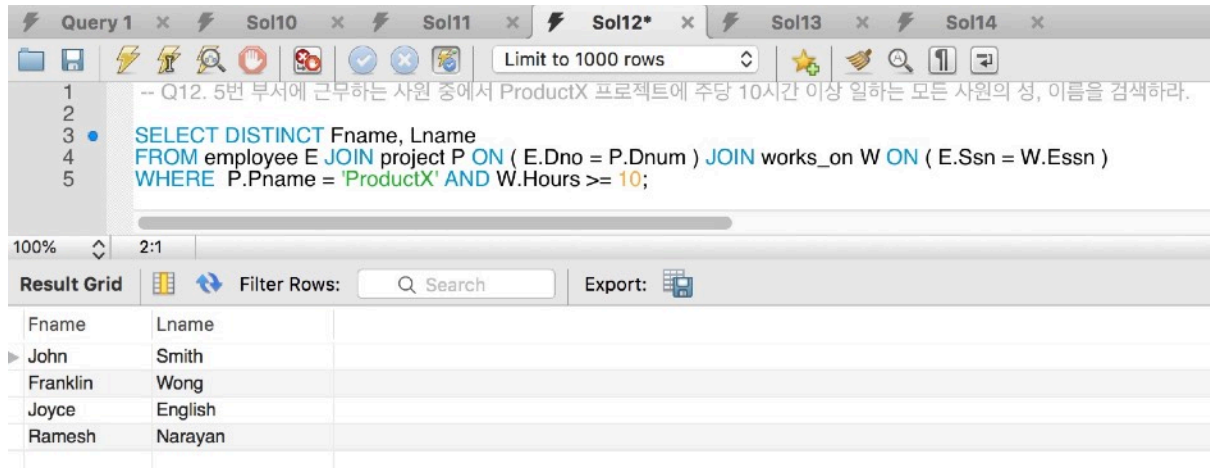
The results are displayed in a table with the following columns: Dname, Fname, Lname, and Pname. The data is sorted by Dname in descending order, and then by Pname in ascending order within each department.

Dname	Fname	Lname	Pname
Research	John	Smith	ProductX
Research	Franklin	Wong	ProductX
Research	Joyce	English	ProductX
Research	Ramesh	Narayan	ProductX
Research	Franklin	Wong	ProductY
Research	Joyce	English	ProductY
Research	Ramesh	Narayan	ProductY
Research	John	Smith	ProductY
Research	Joyce	English	ProductZ
Research	Ramesh	Narayan	ProductZ
Research	John	Smith	ProductZ
Research	Franklin	Wong	ProductZ
Headquarters	James	Borg	Reorganization
Administration	Ahmad	Jabbar	Computerizati...
Administration	Jennifer	Wallace	Computerizati...
Administration	Alicia	Zelaya	Computerizati...
Administration	Ahmad	Jabbar	Newbenefits
Administration	Jennifer	Wallace	Newbenefits
Administration	Alicia	Zelaya	Newbenefits



## ~ Q12 ~

5번 부서에 근무하는 사원 중에서  
*Product X* 프로젝트에 주당 10시간 이상 일하는  
모든 사원의 성과 이름을 검색하라.

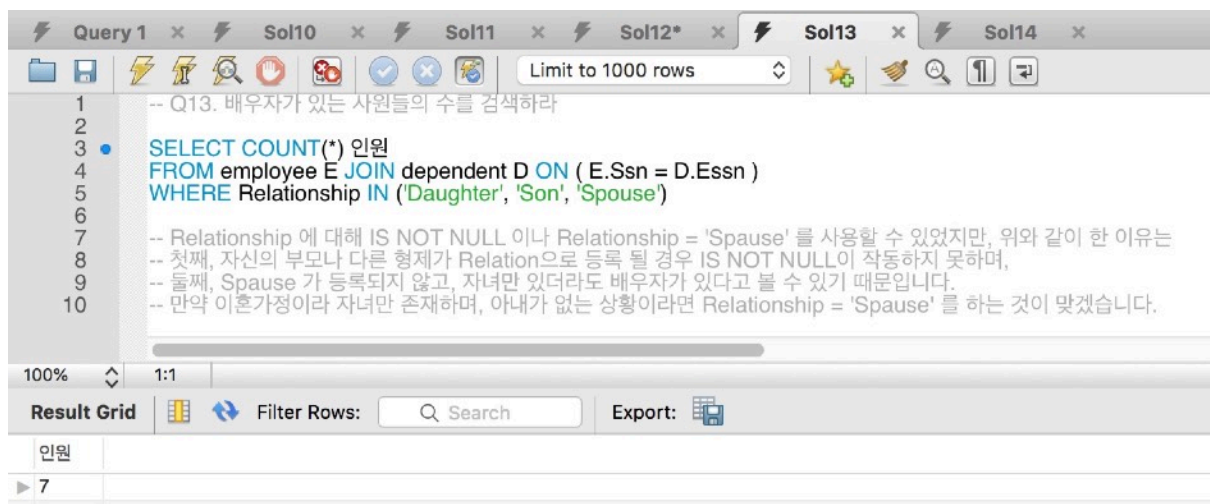


```
-- Q12. 5번 부서에 근무하는 사원 중에서 ProductX 프로젝트에 주당 10시간 이상 일하는 모든 사원의 성, 이름을 검색하라.
SELECT DISTINCT Fname, Lname
FROM employee E JOIN project P ON ( E.Dno = P.Dnum ) JOIN works_on W ON ( E.Ssn = W.Essn )
WHERE P.Pname = 'ProductX' AND W.Hours >= 10;
```

Fname	Lname
John	Smith
Franklin	Wong
Joyce	English
Ramesh	Narayan

## ~ Q13 ~

배우자가 있는 사원들의 수를 검색하라



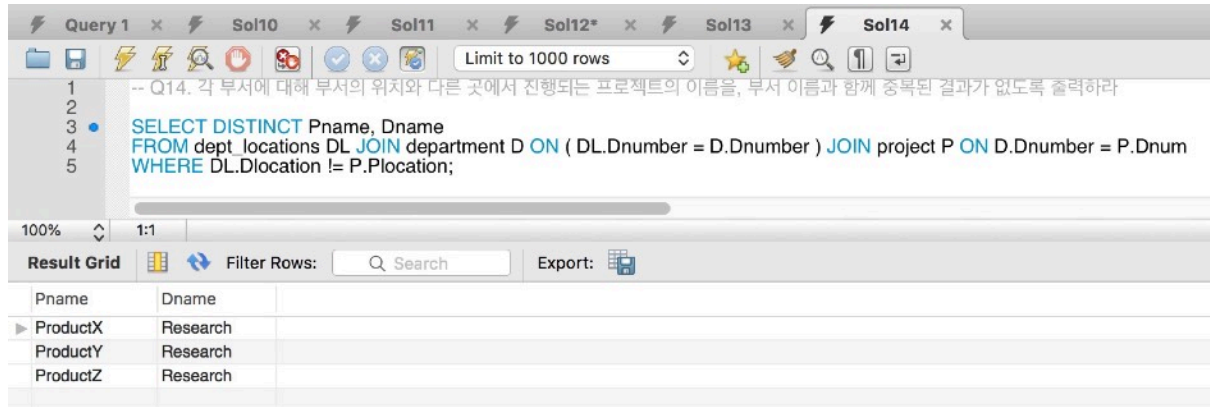
```
-- Q13. 배우자가 있는 사원들의 수를 검색하라
SELECT COUNT(*) 인원
FROM employee E JOIN dependent D ON ( E.Ssn = D.Essn )
WHERE Relationship IN ('Daughter', 'Son', 'Spouse')

-- Relationship 에 대해 IS NOT NULL 이나 Relationship = 'Spouse' 를 사용할 수 있었지만, 위와 같이 한 이유는
-- 첫째, 자신의 부모나 다른 형제가 Relation으로 등록 될 경우 IS NOT NULL이 작동하지 못하며,
-- 둘째, Spouse 가 등록되지 않고, 자녀만 있더라도 배우자가 있다고 볼 수 있기 때문입니다.
-- 만약 이혼가정이라 자녀만 존재하며, 아내가 없는 상황이라면 Relationship = 'Spouse' 를 하는 것이 맞겠습니다.
```

인원
7

## ~ Q14 ~

각 부서에 대해 부서의 위치와 다른 곳에서  
진행되는 프로젝트의 이름을  
부서 이름과 함께 중복이 없도록 출력하라



The screenshot shows a SQL IDE with multiple tabs. The active tab is 'Sol14'. The query editor contains the following SQL code:

```
-- Q14. 각 부서에 대해 부서의 위치와 다른 곳에서 진행되는 프로젝트의 이름을, 부서 이름과 함께 중복된 결과가 없도록 출력하라
SELECT DISTINCT Pname, Dname
FROM dept_locations DL JOIN department D ON ( DL.Dnumber = D.Dnumber ) JOIN project P ON D.Dnumber = P.Dnum
WHERE DL.Dlocation != P.Plocation;
```

The 'Result Grid' shows the following data:

Pname	Dname
ProductX	Research
ProductY	Research
ProductZ	Research