



# 직무부트캠프 강의자료(3주차)

SQL로 배우는 데이터베이스(DB) 실무



# 답안

## 과제 1)

EMP_NO	EMP_NM	EMP_GENDE R	COUPLE_YN	SPOUSE_N M	CHILD_NM	CHILD_GEN DER	CHILD_BIRT H_YMD	CHILD_AGE	TWIN_YN
직원번호	직원성명	직원성별	사내부부여 부	배우자성명	자녀성명	자녀성별	자녀생년 월일	자녀나이	쌍생아여부

## 과제 2) 빨간색 글씨와 같이 애매모호성 제거

- 기혼의 임직원 중 자녀가 1명 이상의 인 자 ( "기혼"의 제한을 없앴. 추출일자 기준, 재직자 중 부양 자녀가 1명 이상인 경우로 추출)
- 자녀 나이가 5~7세인자 (자녀 생년월일 기준 2012년 1월 1일 생 ~ 2015년 12월 31일 생 기준, 명확한 생년월일을 받는 것이 중요!)
- 배우자가 사내커플인 경우는 남편, 아내 중 한쪽만 가능 (사내커플인 경우 부를 대상자에 포함, 모는 제외)
- 배우자가 사내커플인 경우 배우자 명을 표시 (배우자가 사내커플인 경우 "부" 기준으로 "모"의 직원번호 및 배우자 명을 함께 추출)
- 해당자녀의 나이, 성별표시 (해당자녀의 나이, 생년월일, 성별을 추출)
- 쌍생아의 경우 대표 1명만 표시 (쌍생아의 경우 이름으로 오름차순 기준, 상위자로 추출, 쌍생아의 기준 - 관계 코드가 자녀이면서 상호 생년월일이 같은 경우

## 테이블 관련 설명

1. EMP\_C → 인사기본 원장 : 직원의 기본적인 정보 저장되어 있는 이 테이블의 데이터는 항상 최신의 상태로 관리가 됨. 직원번호는 UNIQUE 해야함.  
모든 관계 및 JOIN 에서 가장 기본이 되는 테이블.
2. → 직원 가족 원장, EMP\_C와 1:N 관계를 갖는 테이블로 인사원장과 EMP\_NO로 JOIN 가능. 특징으로 직원별로 한 사람 당 여러 개의 레코드를 가질 수 있음.  
FAM\_C 재직여부를 구분하는 방법 ( RETIRE\_YMD = '99991231' 일 경우 재직 아닐 경우 퇴직)  
가족 관계는 코드로 관리 (REL\_TYPE\_CD)  
CHILD\_SEQ 자녀 순번을 기록하는 용도로 NULL 일 경우도 있으니 유의!
3. ORG\_C → 조직원장 : 인사원장과 ORG\_CD로 JOIN 가능. 계층형 테이블 (AREA\_ORG\_CD, MGR\_ORG\_CD, SUPER\_ORG\_CD) 로 상위 관리조직코드가 ORG\_CD로 존재함.  
조직은 통폐합이 빈번하게 일어나므로 기준일자로 현재 조직이 살아있는 조직인지 알 수 있음 ( SYSDATE BETWEEN STA\_YMD AND END\_YMD)
4. FAM\_REL\_C → 가족 임직원 원장 : 가족 중 임직원이 있을 경우 별도 관리하기 위한 테이블, EMP\_NO와 EMP\_REL\_NO는 상호 가족임을 나타냄.  
관계는 코드로 관리 (REL\_TYPE\_CD)  
선분이력관리 (STA\_YMD, END\_YMD)로 특정 시점에 어떤 임직원이 있었는지 관리 가능 (인사 발령과 같은 업무에서 이력을 알고 싶을 경우 이러한 기법을 사용함)

# 답안

이번 과제의 의도는 데이터 추출 전에 반드시 해야 할 단계를 습득하는 것이며, 가장 중요한 현업의 요구사항을 파악하는 것입니다. 한번 전달된 요구사항에서 최대한 애매모호한 부분을 현업 담당자와 확실히 하여, 데이터 재추출을 1차, 2차 최대한 반복하지 않도록 하는 것이 업무를 효율적으로 하는 방법 중 하나입니다. 아무래도 재추출이 반복 된다면, 그만큼 다른 업무에 방해받는 요소가 많아, 효율적인 업무를 하는 것이 제한이 되겠죠. 누가 보아도 이해할 수 있을 정도로 구체화 및 명확화 하는 것이 포인트 입니다.

두번째로 데이터 추출 전, 전달 해야 할 데이터 형태에 대해서 고민하고, 이 데이터들을 현재 보유하고 있는 데이터 베이스에서 커버가 가능한지 테이블 및 컬럼을 탐색 하셔야 합니다. 무작정 SQL을 작성하다가, 추출불가한 항목이 있을 경우, 더 난감한 상황이 발생할 수 있으니, 전달할 데이터 형태를 확정하고, 어떤 테이블의 어떤 컬럼을 사용할 것인지 탐색을 반드시 하시기 바랍니다. (아래 표 참조)

	EMP_NO	EMP_NM	EMP_GENDE R	COUPLE_YN	SPOUSE_NM	CHILD_NM	CHILD_GEN DER	CHILD_BIRT H_YMD	CHILD_AGE	TWIN_YN
	직원번호	직원성명	직원 성별	사내부부 여부	배우자성명	자녀성명	자녀 성별	자녀 생년월일	자녀나이	쌍생아여부
테이블	EMP_C	EMP_C	EMP_C	FAM_REL_C	FAM_REL_C EMP_C	FAM_C	FAM_C	FAM_C	FAM_C	FAM_C
컬럼	EMP_NO	EMP_NM	GENDER_CD	EMP_REL_NO	EMP_REL_NO, EMP_NM	FAM_NM	GENDER_CD	BIRTH_YMD	BIRTH_YMD	REL_TYPE_CD

세번째로 테이블의 형태는 아래 4가지가 가장 많이 사용됩니다. (OLTP 기준)

1. 항상 최신의 상태로 관리되는 테이블 (인사원장)
2. 이력을 담고 있는 테이블 (조직원장), 그래서, STA\_YMD (시작일)과 END\_YMD(종료일)이 있어, 특정 시점에 데이터를 추출 할 수 있는 특징이 있습니다.
3. 순환관계 테이블 : 카테고리, 조직트리, 계층 등의 데이터를 관리할 때 많이 사용됩니다. 레코드에 조직코드 와 상위조직코드를 같은 레코드에 관리하여, 두 테이블을 조인하여 상위 조직코드가 어떤 조직인지 알 수 있습니다.  
기술적인 부분은 이 링크를 참고하세요 : <https://offbyone.tistory.com/109>
4. 1:N 관계 테이블 (FAM\_C, 가족원장), EMP\_NO는 여러 가족을 가질 수 있으므로, 1:N이 성립합니다.  
EMP\_NO 기준으로 여러 레코드를 가질 수 있는 특징이 있습니다.
5. N:M 관계 테이블 → 이 테이블은 추후 과제에서 다룰 예정입니다.

(세줄 요약)

1. 요구사항을 초딩이 봐도 이해할 수준으로 명확히 하라.
2. 추출 전 테이블과 컬럼을 탐색하라. 추출할 수 없는 데이터에 대해서는 현업과 상의하라.
3. 테이블의 형태와 특징을 알고 있어야 하며, 특징에 맞는 SQL을 작성해야 한다.

# 오늘 같이 해결할 과제는 아래와 같습니다.

주 차	내 용	중 점 과 제
1주	데이터 추출 작업 준비	실습환경 구성
2주	현업 요구사항 분석 데이터 추출 및 올바른 SQL 문장 작성	데이터 추출 요청서 분석 SQL 작성
3주	현업 요구사항에 맞게 데이터 추출하기	SQL작성
4주	다양한 형태의 데이터 추출하기	Oracle 내장함수 응용하기



# Divide and Conquer

- **분할정복법**
- 주어진 문제의 입력을 더 이상 나눌 수 없을 때까지 두 개 이상의 작은 문제로 순환적으로 분할하고, 이렇게 분할된 작은 문제들을 각각 해결한 후 그 해를 결합하여 원래 문제의 해를 구하는 방식.
- SQL을 작성할 때도, 해결할 문제를 분할하여 작성하고, 합쳐 나가는 습관을 갖는 것이 중요.



## ✓ 데이터 select

워크시트    질의 작성기

```
SELECT * FROM EMP_C EMP;
```

질의... x

SQL | 50개의 행이 인출됨(0,054초)

	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YMD	NOTE
1	10001389	이ㅇ혁	1	19390519	A100000	98/10/30	96/02/07	(null)
2	10001392	권ㅇ혁	1	19400611	A100000	98/02/26	96/02/07	(null)
3	10001404	황ㅇ남	1	19410407	A100000	99/02/27	96/02/07	(null)
4	10001418	윤ㅇ학	1	19410415	A100000	98/02/26	96/02/07	(null)
5	10001421	이ㅇ영	1	19411127	A100000	99/02/27	96/02/07	(null)
6	10001449	권ㅇ철	1	19420108	A100000	98/02/26	96/02/07	(null)
7	10001452	윤ㅇ조	1	19411004	A100000	99/01/02	96/02/23	(null)
8	10001466	김ㅇ태	1	19420115	A100000	99/01/02	96/02/23	(null)
9	10001972	김ㅇ철	1	19510910	A100000	04/02/29	00/01/17	(null)
10	10002431	정ㅇ백	1	19551029	A100000	04/02/29	03/08/01	(null)

✓ 데이터 정렬 - order by

워크시트

질의 작성기

```

SELECT * FROM EMP_C EMP
ORDER BY EMP_NM;

```

질의 결과 x

SQL | 50개의 행이 인출됨(0,015초)

	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YMD	NOTE
1	11500067	가ㅇ	1	19720301	A123240	17/01/23	99/09/01	(null)
2	11500036	가ㅇ	1	19300125	AK02443	84/05/23	63/06/01	(null)
3	11500053	가ㅇ란	2	19671112	AK00688	92/03/07	89/10/10	(null)
4	11500019	가ㅇ수	1	19641227	A109100	96/06/03	92/09/14	(null)
5	11500040	가ㅇ옥	2	19680505	A500000	98/04/13	86/03/11	(null)
6	11500022	가ㅇ원	2	19630506	A912840	92/02/12	81/12/16	(null)
7	11500071	가ㅇ정	2	19780906	A122105	99/12/31	97/02/17	(null)
8	11500084	간ㅇ섭	1	19691016	A501000	02/03/06	88/02/15	(null)
9	11500098	갈ㅇ일	1	19460521	AK02459	00/06/10	75/10/06	(null)
10	11500100	갈ㅇ희	2	19690401	A602000	94/04/25	90/11/19	(null)

## ✓ 데이터 그룹핑 - group by

워크시트    질의 작성기	
<pre>SELECT EMP_NM, COUNT(*) FROM EMP_C EMP GROUP BY EMP_NM;</pre>	
질의 결과 x	
SQL   50개의 행이 인출됨(0초)	
EMP_NM	COUNT(*)
1 정ㅇ현	1
2 황ㅇ희	1
3 박ㅇ원	3
4 강ㅇ숙	60
5 강ㅇ희	64
6 강ㅇ현	10
7 강ㅇ준	7
8 강ㅇ오	2
9 강ㅇ자	22
10 이ㅇ재	3




## ✓ 데이터 존재여부 확인 - exists

워크시트

질의 작성기

```
SELECT * FROM EMP_C EMP
WHERE EXISTS
    (SELECT 1 FROM EMP_C EMP2 WHERE EMP2.EMP_NO = EMP.EMP_NO
      AND EMP2.EMP_NM LIKE '이%');
```

질의 결과 x

 SQL | 50개의 행이 인출됨(0.016초)

	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YMD	NOTE
1	10001389	이오혁	1	19390519	A100000	98/10/30	96/02/07	(null)
2	10001421	이오영	1	19411127	A100000	99/02/27	96/02/07	(null)
3	10002413	이오남	2	19471111	A100000	04/04/20	03/03/21	(null)
4	10002400	이오영	1	19521225	A100000	04/02/29	03/02/05	(null)
5	10001816	이오주	1	19391122	A100000	03/03/21	99/02/27	(null)
6	10001760	이오민	1	19430617	A100000	02/03/22	99/01/02	(null)
7	10004354	이오호	1	19650403	A121730	10/01/08	08/01/01	(null)
8	10005238	이오재	1	19590124	A125910	12/01/01	10/08/03	(null)
9	10005133	이오기	1	19570423	A124200	10/03/26	10/03/25	(null)
10	10005091	이오철	1	19480220	A124200	10/03/25	08/09/29	(null)

## ✓ 데이터 검색 - not exists

워크시트

질의 작성기

## ✓ 데이터 여러 조건 검색 - in

워크시트

질의 작성기

<



## ✓ 데이터 여러 조건 검색 - not in

워크시트

질의 작성기

SELECT \* FROM EMP\_C EMP  
WHERE EMP\_NO NOT IN  
(SELECT EMP\_NO FROM EMP\_C EMP2 WHERE EMP2.EMP\_NO = EMP.EMP\_NO  
AND EMP2.EMP\_NM LIKE 'O|\*');

질의 결과 x

SQL | 50개의 행이 인출됨(0초)

	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YMD	NOTE
1	10001392	권ㅇ혁	1	19400611	A100000	98/02/26	96/02/07	(null)
2	10001404	황ㅇ남	1	19410407	A100000	99/02/27	96/02/07	(null)
3	10001418	윤ㅇ학	1	19410415	A100000	98/02/26	96/02/07	(null)
4	10001449	권ㅇ철	1	19420108	A100000	98/02/26	96/02/07	(null)
5	10001452	윤ㅇ조	1	19411004	A100000	99/01/02	96/02/23	(null)
6	10001466	김ㅇ태	1	19420115	A100000	99/01/02	96/02/23	(null)
7	10001972	김ㅇ철	1	19510910	A100000	04/02/29	00/01/17	(null)
8	10002431	정ㅇ백	1	19551029	A100000	04/02/29	03/08/01	(null)
9	10001986	김ㅇ훈	1	19420313	A100000	04/03/19	00/03/18	(null)
10	10001990	윤ㅇ현	1	19450701	A100000	02/03/18	00/04/06	(null)

## ✓ 데이터 여러 조건 검색 - not in

워크시트

질의 작성기

SELECT \* FROM EMP\_C EMP  
WHERE ( EMP\_NO, EMP\_NM) NOT IN  
(SELECT EMP\_NO, EMP\_NM FROM EMP\_C EMP2 WHERE EMP2.EMP\_NO = EMP.EMP\_NO  
AND EMP2.EMP\_NM LIKE 'O|S');

질의 결과 x

SQL | 50개의 행이 인출됨(0초)

	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YMD	NOTE
1	10001392	권ㅇ혁	1	19400611	A100000	98/02/26	96/02/07	(null)
2	10001404	황ㅇ남	1	19410407	A100000	99/02/27	96/02/07	(null)
3	10001418	윤ㅇ학	1	19410415	A100000	98/02/26	96/02/07	(null)
4	10001449	권ㅇ철	1	19420108	A100000	98/02/26	96/02/07	(null)
5	10001452	윤ㅇ조	1	19411004	A100000	99/01/02	96/02/23	(null)
6	10001466	김ㅇ태	1	19420115	A100000	99/01/02	96/02/23	(null)
7	10001972	김ㅇ철	1	19510910	A100000	04/02/29	00/01/17	(null)
8	10002431	정ㅇ백	1	19551029	A100000	04/02/29	03/08/01	(null)
9	10001986	김ㅇ훈	1	19420313	A100000	04/03/19	00/03/18	(null)
10	10001990	윤ㅇ현	1	19450701	A100000	02/03/18	00/04/06	(null)

✓ 데이터 검색 - like

워크시트

질의 작성기

```

SELECT * FROM EMP_C EMP
WHERE EMP_NM LIKE '0|%'

```

질의 결과 x

SQL | 50개의 행이 인출됨(0.016초)

	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YMD	NOTE
1	10001389	이ㅇ혁	1	19390519	A100000	98/10/30	96/02/07	(null)
2	10001421	이ㅇ영	1	19411127	A100000	99/02/27	96/02/07	(null)
3	10002413	이ㅇ남	2	19471111	A100000	04/04/20	03/03/21	(null)
4	10002400	이ㅇ영	1	19521225	A100000	04/02/29	03/02/05	(null)
5	10001816	이ㅇ주	1	19391122	A100000	03/03/21	99/02/27	(null)
6	10001760	이ㅇ민	1	19430617	A100000	02/03/22	99/01/02	(null)
7	10004354	이ㅇ호	1	19650403	A121730	10/01/08	08/01/01	(null)
8	10005238	이ㅇ재	1	19590124	A125910	12/01/01	10/08/03	(null)
9	10005133	이ㅇ기	1	19570423	A124200	10/03/26	10/03/25	(null)
10	10005091	이ㅇ철	1	19480220	A124200	10/03/25	08/09/29	(null)



✓ 데이터 검색 - not like

워크시트
질의 작성기

```
SELECT * FROM EMP_C EMP
WHERE EMP_NM NOT LIKE 'O|*';
```

▶
질의 결과 x

📌
📄
🔗
✖
SQL | 50개의 행이 인출됨(0.015초)

	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YMD	NOTE
1	10001392	권ㅇ혁	1	19400611	A100000	98/02/26	96/02/07	(null)
2	10001404	황ㅇ남	1	19410407	A100000	99/02/27	96/02/07	(null)
3	10001418	윤ㅇ학	1	19410415	A100000	98/02/26	96/02/07	(null)
4	10001449	권ㅇ철	1	19420108	A100000	98/02/26	96/02/07	(null)
5	10001452	윤ㅇ조	1	19411004	A100000	99/01/02	96/02/23	(null)
6	10001466	김ㅇ태	1	19420115	A100000	99/01/02	96/02/23	(null)
7	10001972	김ㅇ철	1	19510910	A100000	04/02/29	00/01/17	(null)
8	10002431	정ㅇ백	1	19551029	A100000	04/02/29	03/08/01	(null)
9	10001986	김ㅇ훈	1	19420313	A100000	04/03/19	00/03/18	(null)
10	10001990	윤ㅇ현	1	19450701	A100000	02/03/18	00/04/06	(null)

## ✓ 데이터 순번부여 - rownum

워크시트    질의 작성기

```
SELECT ROWNUM, EMP.* FROM EMP_C EMP  
WHERE EMP_NM NOT LIKE '이*';
```

질의 결과 x

SQL | 50개의 행이 인출됨(0초)

	ROWNUM	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YM
1	1	10001392	권ㅇ혁	1	19400611	A100000	98/02/26	96/02/07
2	2	10001404	황ㅇ남	1	19410407	A100000	99/02/27	96/02/07
3	3	10001418	윤ㅇ학	1	19410415	A100000	98/02/26	96/02/07
4	4	10001449	권ㅇ철	1	19420108	A100000	98/02/26	96/02/07
5	5	10001452	윤ㅇ조	1	19411004	A100000	99/01/02	96/02/23
6	6	10001466	김ㅇ태	1	19420115	A100000	99/01/02	96/02/23
7	7	10001972	김ㅇ철	1	19510910	A100000	04/02/29	00/01/17
8	8	10002431	정ㅇ백	1	19551029	A100000	04/02/29	03/08/01
9	9	10001986	김ㅇ훈	1	19420313	A100000	04/03/19	00/03/18
10	10	10001990	윤ㅇ현	1	19450701	A100000	02/03/18	00/04/06

## ✓ 데이터 순위부여 - rank

워크시트    질의 작성기

```
SELECT RANK() OVER (ORDER BY EMP_NM ASC) AS RN , EMP.* FROM EMP_C EMP;
```

질의 결과 x

SQL | 50개의 행이 인출됨(0.034초)

	RN	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRE_YMD	M
1	1	11500067	가ㅇ	1	19720301	A123240	17/01/23	99/09/01	(nu
2	1	11500036	가ㅇ	1	19300125	AK02443	84/05/23	63/06/01	(nu
3	3	11500053	가ㅇ란	2	19671112	AK00688	92/03/07	89/10/10	(nu
4	4	11500019	가ㅇ수	1	19641227	A109100	96/06/03	92/09/14	(nu
5	5	11500040	가ㅇ옥	2	19680505	A500000	98/04/13	86/03/11	(nu
6	6	11500022	가ㅇ원	2	19630506	A912840	92/02/12	81/12/16	(nu
7	7	11500071	가ㅇ정	2	19780906	A122105	99/12/31	97/02/17	(nu
8	8	11500084	간ㅇ섭	1	19691016	A501000	02/03/06	88/02/15	(nu
9	9	11500098	갈ㅇ일	1	19460521	AK02459	00/06/10	75/10/06	(nu
10	10	11500100	갈ㅇ희	2	19690401	A602000	94/04/25	90/11/19	(nu



## ✓ 데이터 코드변환 - decode

워크시트

질의 작성기

```
SELECT DECODE(GENDER_CD, '1', '남', '여') AS 성별, EMP.* FROM EMP_C EMP;
```

질의 결과 x

SQL | 50개의 행이 인출됨(0초)

성별	EMP_NO	EMP_NM	GENDER_CD	BIRTH_YMD	ORG_CD	RETIRE_YMD	HIRES_YMD	
1 남	10001389	이○혁	1	19390519	A100000	98/10/30	96/02/07	(r
2 남	10001392	권○혁	1	19400611	A100000	98/02/26	96/02/07	(r
3 남	10001404	황○남	1	19410407	A100000	99/02/27	96/02/07	(r
4 남	10001418	윤○학	1	19410415	A100000	98/02/26	96/02/07	(r
5 남	10001421	이○영	1	19411127	A100000	99/02/27	96/02/07	(r
6 남	10001449	권○철	1	19420108	A100000	98/02/26	96/02/07	(r
7 남	10001452	윤○조	1	19411004	A100000	99/01/02	96/02/23	(r
8 남	10001466	김○태	1	19420115	A100000	99/01/02	96/02/23	(r
9 남	10001972	김○철	1	19510910	A100000	04/02/29	00/01/17	(r
10 남	10002431	정○백	1	19551029	A100000	04/02/29	03/08/01	(r

## ✓ 데이터 조건부 - case when

워크시트

질의 작성기

## ✓ 데이터 합치기 - union all

워크시트    질의 작성기

```
SELECT EMP_NO FROM EMP_C WHERE EMP_NO = '10001389'  
UNION ALL  
SELECT EMP_NO FROM EMP_C WHERE EMP_NO = '10001389';
```

질의 결과 x

SQL | 인출된 모든 행: 2(0초)

	EMP_NO
1	10001389
2	10001389



## ✓ 데이터 합치기 - union

워크시트    질의 작성기

```
SELECT EMP_NO FROM EMP_C WHERE EMP_NO = '10001389'  
UNION  
SELECT EMP_NO FROM EMP_C WHERE EMP_NO = '10001389';
```

질의 결과 x

SQL | 인출된 모든 행: 1(0초)

	EMP_NO
1	10001389





## ✓ 데이터 집계 - sum, avg, count

워크시트

질의 작성기

```
SELECT SUM(EMP_NO), AVG(EMP_NO), COUNT(EMP_NO) FROM EMP_C EMP;
```

질의 결과 x



SQL | 인출된 모든 행: 1(0.043초)

	SUM(EMP_NO)	AVG(EMP_NO)	COUNT(EMP_NO)
1	22044555256	11049902.38395989974937343358395989974937	1995

## ✓ 데이터 집계 - min, max

워크시트		질의 작성기	
		<pre>SELECT MAX(EMP_NO), MIN(EMP_NO) FROM EMP_C EMP;</pre>	
질의 결과 x		SQL   인출된 모든 행: 1(0초)	
		MAX(EMP_NO)	MIN(EMP_NO)
1		11513883	10001389



## ✓ 데이터 집계 - having

워크시트    질의 작성기	
<pre>SELECT EMP_NM, COUNT(*) FROM EMP_C EMP GROUP BY EMP_NM HAVING COUNT(*) &gt; 10;</pre>	
질의 결과 x	
SQL   인출된 모든 행: 32(0,004초)	
EMP_NM	COUNT(*)
1 강○속	60
2 강○희	64
3 강○자	22
4 고○수	11
5 강○주	24
6 강○경	31
7 강○성	25
8 강○영	41
9 고○희	14
10 강○남	13

## ✓ 데이터 조인 - inner join

워크시트    질의 작성기

```
SELECT  
EMP.EMP_NO, FAM.FAM_NM, REL_TYPE_CD  
FROM EMP_C EMP , FAM_C FAM  
WHERE EMP.EMP_NO = FAM.EMP_NO --INNER JOIN  
AND FAM.REL_TYPE_CD = 'A27';
```

질의 결과 x

SQL | 50개의 행이 인출됨(0.006초)

	EMP_NO	FAM_NM	REL_TYPE_CD
1	10001452	윤ㅇ현	A27
2	10001483	한ㅇ환	A27
3	10001561	미ㅇ기	A27
4	10001588	성ㅇ진	A27
5	10001588	성ㅇ영	A27
6	10001651	안ㅇ호	A27
7	10001665	김ㅇ선	A27
8	10001679	김ㅇ식	A27
9	10001739	조ㅇ교	A27
10	10001739	조ㅇ호	A27

## ✓ 데이터 조인 - left outer join

워크시트    질의 작성기

```
SELECT  
EMP.EMP_NO, FAM.FAM_NM, REL_TYPE_CD  
FROM EMP_C EMP , FAM_C FAM  
WHERE EMP.EMP_NO = FAM.EMP_NO (+) --LEFT OUTER JOIN  
AND FAM.REL_TYPE_CD (+) = 'A27'  
AND FAM_NM IS NULL;
```

질의 결과 x

SQL | 50개의 행이 인출됨(0.002초)

	EMP_NO	FAM_NM	REL_TYPE_CD
1	11505155	(null)	(null)
2	11501656	(null)	(null)
3	11509070	(null)	(null)
4	11512459	(null)	(null)
5	11509402	(null)	(null)
6	11509681	(null)	(null)
7	10006784	(null)	(null)
8	10002492	(null)	(null)
9	10002504	(null)	(null)
10	11504115	(null)	(null)



## ✓ 데이터 조인 - nvl (null값 처리)

워크시트

질의 작성기

SELECT

EMP.EMP\_NO, FAM.FAM\_NM, REL\_TYPE\_CD, NVL(FAM.FAM\_NM , '자녀없음')

FROM EMP\_C EMP , FAM\_C FAM

WHERE EMP.EMP\_NO = FAM.EMP\_NO (+) --LEFT OUTER JOIN

AND FAM.REL\_TYPE\_CD (+) = 'A27'

AND FAM\_NM IS NULL;

질의 결과 x

SQL | 50개의 행이 인출됨(0.034초)

	EMP_NO	FAM_NM	REL_TYPE_CD	NVL(FAM,FAM_NM,'자녀없음')
1	11505155	(null)	(null)	자녀없음
2	11501656	(null)	(null)	자녀없음
3	11509070	(null)	(null)	자녀없음
4	11512459	(null)	(null)	자녀없음
5	11509402	(null)	(null)	자녀없음
6	11509681	(null)	(null)	자녀없음
7	10006784	(null)	(null)	자녀없음
8	10002492	(null)	(null)	자녀없음
9	10002504	(null)	(null)	자녀없음
10	11504115	(null)	(null)	자녀없음

## ✓ 데이터 행렬변환 - listagg

워크시트    질의 작성기	
<pre>SELECT EMP.EMP_NO, LISTAGG(FAM.FAM_NM, ',') WITHIN GROUP (ORDER BY EMP.EMP_NO) FROM EMP_C EMP , FAM_C FAM WHERE EMP.EMP_NO = FAM.EMP_NO (+) --LEFT OUTER JOIN AND FAM.REL_TYPE_CD (+) = 'A27' --AND EMP.EMP_NO = '10003791' GROUP BY EMP.EMP_NO;</pre>	
질의 결과 x	
SQL   50개의 행이 인출될(0.007초)	
EMP_NO	LISTAGG(FAM,FAM_NM,',')WITHINGROUP(ORDERBYEMP,EMP_NO)
10001389	(null)
10001392	(null)
10001404	(null)
10001418	(null)
10001421	(null)
10001449	(null)
10001452	윤ㅇ웅, 윤ㅇ현
10001466	김ㅇ우, 김ㅇ정
10001470	민ㅇ홍, 민ㅇ홍
10001483	한ㅇ진, 한ㅇ환, 한ㅇ환, 한ㅇ훈

```
SELECT
EMP.EMP_NO AS 직원번호,
LISTAGG(FAM_NM, ',') WITHIN GROUP (ORDER
BY EMP.EMP_NO) AS 자녀성명
FROM EMP_C EMP
INNER JOIN FAM_C FAM
ON EMP.EMP_NO = FAM.EMP_NO
AND FAM.REL_TYPE_CD = 'A27'
WHERE EMP.EMP_NO = '10001483'
GROUP BY EMP.EMP_NO
;
```


## ✓ pivot 과 inline view

워크시트

질의 작성기

```
SELECT * FROM (  
    SELECT * FROM (  
        SELECT SUBSTR(BIRTH_YMD,0,4) AS BIRTH_YM, ORG_CD,EMP_NM FROM  
    )  
    ) PIVOT ( COUNT(EMP_NM) FOR BIRTH_YM IN ('1950','1951','1952','1953') );
```

질의 결과 x

 SQL | 50개의 행이 인출됨(0.002초)

	ORG_CD	'1950'	'1951'	'1952'	'1953'
1	A100200	1	0	0	0
2	A183710	0	0	1	1
3	A419316	0	0	0	0
4	A125540	0	0	0	0
5	A125002	0	0	0	0
6	A131610	0	0	0	0
7	A161140	0	0	0	0
8	A155030	0	1	0	0
9	A162050	0	0	0	0
10	A161810	0	0	1	0



- 문제1) 직원번호 10004141 의 이름은? 힌트 : emp\_c에 있음
- 문제2) 직원번호 10004141 는 몇 명의 가족이 있나요? 힌트 fam\_c
- 문제3) 직원번호 10004141 의 소속 부서 코드는? 힌트 emp\_c
- 문제4) 직원번호 10004141 의 소속 부서명은? 힌트 org\_c
- 문제5) 남자 / 여자 직원이 몇 명인지 한번에 알 수 있는 sql을 작성하세요. (group by gender\_cd)
- 문제6) 직원별 자녀가 몇 명인지 한번에 알 수 있는 sql을 작성하세요. (rel\_type\_cd가 A27이면 자녀)
- 문제7) 생년월일이 1970년 1월 1일 이전인 직원의 수를 구하는 sql을 작성하세요.
- 문제8) 현재 재직 중인 직원의 전체 수를 구하는 sql을 작성하세요.