

8. 그룹함수

- 여러 행 또는 테이블 전체에 대해 함수가 적용되어 하나의 결과를 가져오는 함수

함 수	설 명
COUNT	행의 개수를 카운트한다.
SUM	해당 열의 총 행의 합계를 구한다.
AVG	해당 열의 평균을 구한다.
MIN	해당 열의 총 행중에 최소 값을 구한다.
MAX	해당 열의 총 행중에 최대 값을 구한다.

- 그룹함수에서는 null값을 무시하고 결과를 산출함
- 합을 구하는 SUM 함수

```
SELECT SUM(salary)
FROM employees;
```

- ① 문제 : 총 상여금(보너스)을 구하세요.

- 평균을 구하는 AVG 함수

```
SELECT AVG(salary)
FROM employees;
```

- ② 문제 : 프로그래머들의 월급 평균을 구하세요.

- 최대 값을 구하는 MAX 함수

- ③ 문제 : 회계사(Accountant)중 최대 연봉이 얼마인지 알아보세요.

■ 최소 값을 구하는 MIN 함수

- ④ 미국 South San Francisco에 근무하는 직원 중 최소 급여를 받는 직원이 누구인지 알아보세요.

■ 로우의 개수를 세는 COUNT 함수

```
SELECT COUNT(job_id)
FROM employees;
```

- ⑤ 문제 : 직업의 개수를 구하여라.

```
select count(*)
from employees
where to_char(hire_date, 'yyyy') = 1997
```

■ 그룹함수와 일반 칼럼과의 관계

```
SELECT MAX(salary), employee_id
FROM employees;
```

9. GROUP BY 절

■ 문법

```
SELECT 컬럼명 (별칭)..
FROM 테이블명
WHERE 조건(연산자)
[GROUP BY 그룹화필드명]
ORDER BY 컬럼명(별칭, 순서)
```

■ GROUP BY 절에는 별칭을 사용할 수 없음(컬럼명만)

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
```

⑦ 부서별로 전체인원수와 커미션을 받는 직원들의 수를 계산하세요.

■ 하나 이상의 컬럼을 그룹화

```
SELECT department_id ,job_id ,COUNT(*),SUM(salary)
FROM employees
GROUP BY department_id,job_id
order by 1;
```

⑧ 각 부서별로 인원수, 급여의 평균, 최저급여, 최대급여, 급여의 합을 구하여
급여의 합이 많은 순으로 출력하세요.

■ HAVING 절

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
HAVING AVG(salary) >= 8000
```

⑨ 부서인원이 4명보다 많은 부서의 부서번호, 인원수, 급여의 합을 구하세요.

10. SQL의 실행순서

■ from -> where -> group by -> having -> select

■ 테이블이 모두 메모리상에 올라옴

■ where조건에 맞는 열(레코드)들만 분류되어 다시 가상 테이블 생성

- group by가 있을 경우 열 묶음
- having 조건 분류
- 그 중 select에 있는 열만 화면에 출력

11. 서브쿼리의 문법과 종류

- 서브쿼리는 2개이상의 쿼리를 한 쿼리문으로 작성하는 것

1) 이름이 Steven이고 성이 King인 사람의 부서명을 알고 싶다면

```
SELECT first_name, last_name, department_id
FROM employees
WHERE last_name = 'King' AND first_name = 'Steven';
```

결과

FIRST_NAME	LAST_NAME	DEPARTMENT_ID
Steven	King	90

2) department_id를 알고 다시 쿼리를 수행해야 함

```
SELECT department_name
FROM departments
WHERE department_id = 90;
```

결과 : Executive

- 서브쿼리를 이용하여 부서명 구하기

```
SELECT department_name
FROM departments
WHERE department_id = (SELECT department_id
                        FROM employees
                        WHERE last_name = 'King' AND first_name = 'Steven');
```

- ① department_id가 60인 부서의 도시명을 알아내세요.
- ② 사번이 107인 사람과 부서가 같고, 167번의 급여보다 많은 직원들의 사번, 이름, 성, 급여를 출력하세요.
- ③ 급여 평균보다 연봉을 적게 받는 직원들 중 커미션을 받는 직원들의 사번, 이름, 성, 급여, 커미션 퍼센트를 출력하세요.
- ④ 각 부서의 최소급여가 20번 부서의 최소급여보다 많은 부서의 번호와 그 부서의 최소급여를 출력하세요.
- ⑤ 직업 중 급여의 평균이 가장 적은 직업을 출력하세요.

■ 다중행 서브쿼리

종 류	의 미
IN	메인쿼리의 비교조건이 서브쿼리의 결과중에서 하나라도 일치시 참
ANY, SOME	메인쿼리의 비교조건이 서브쿼리의 결과와 하나 이상이 일치하면 참
ALL	메인쿼리의 비교조건이 서브쿼리의 결과와 모든 값이 일치하면 참
EXIST	메인쿼리의 비교조건이 서브쿼리의 결과중에서 만족하는 값이 하나라도 존재하면 참

■ IN 연산자

- ⑥ 각 부서별 최소 급여를 구하고, 그 급여들과 같은 급여를 받는 직원들을

구하세요.

■ FROM절에 서브쿼리

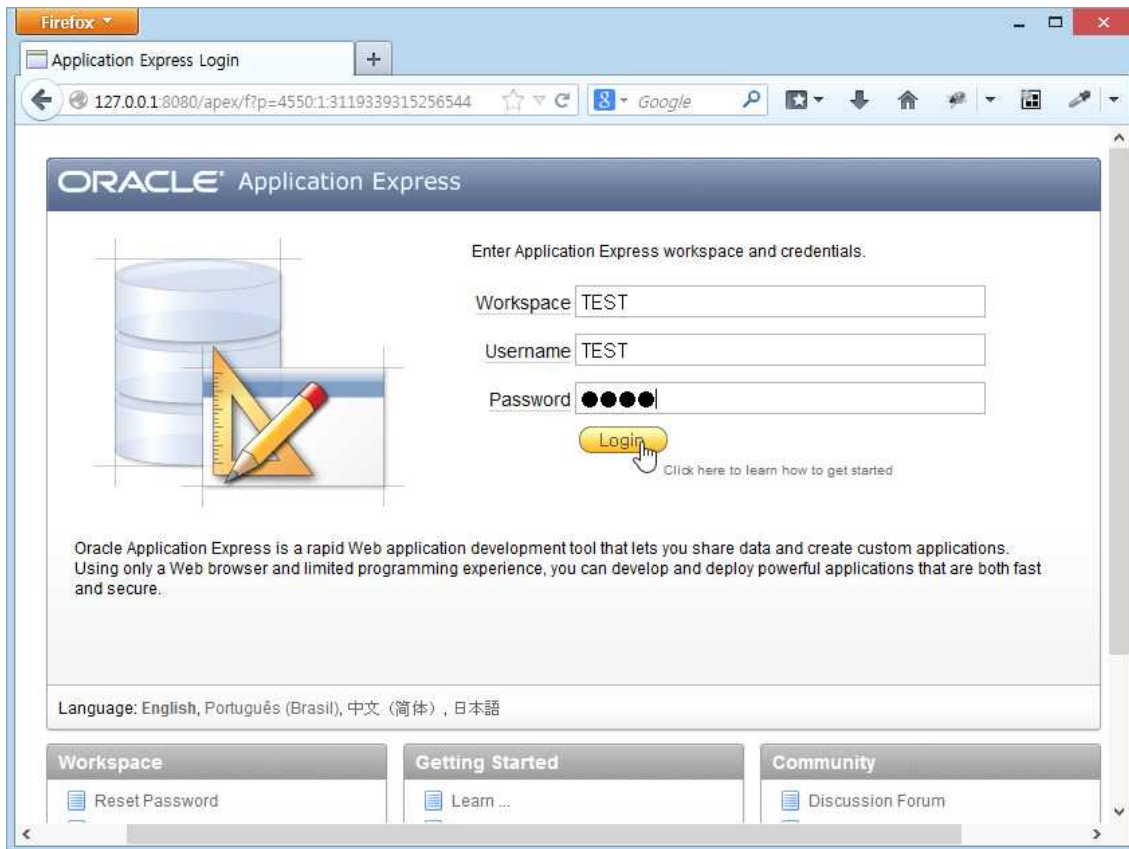
- ⑦ employees 테이블과 department 테이블, locations 테이블에서 업무가 IT_PROG인 사원의 정보를 이름, 성, 업무, 부서명, 근무도시를 출력하세요.

12. 계정 생성

The screenshot shows a Firefox browser window with the Oracle Application Express interface. The address bar shows the URL `/f?p=4950:7:245905741087170::`. The navigation tabs include Home, Storage, Sessions, Parameters, and Application Express (which is active). The breadcrumb trail shows Home > Oracle Application Express. The main content area is titled 'Create Application Express Workspace' and contains the following fields and options:

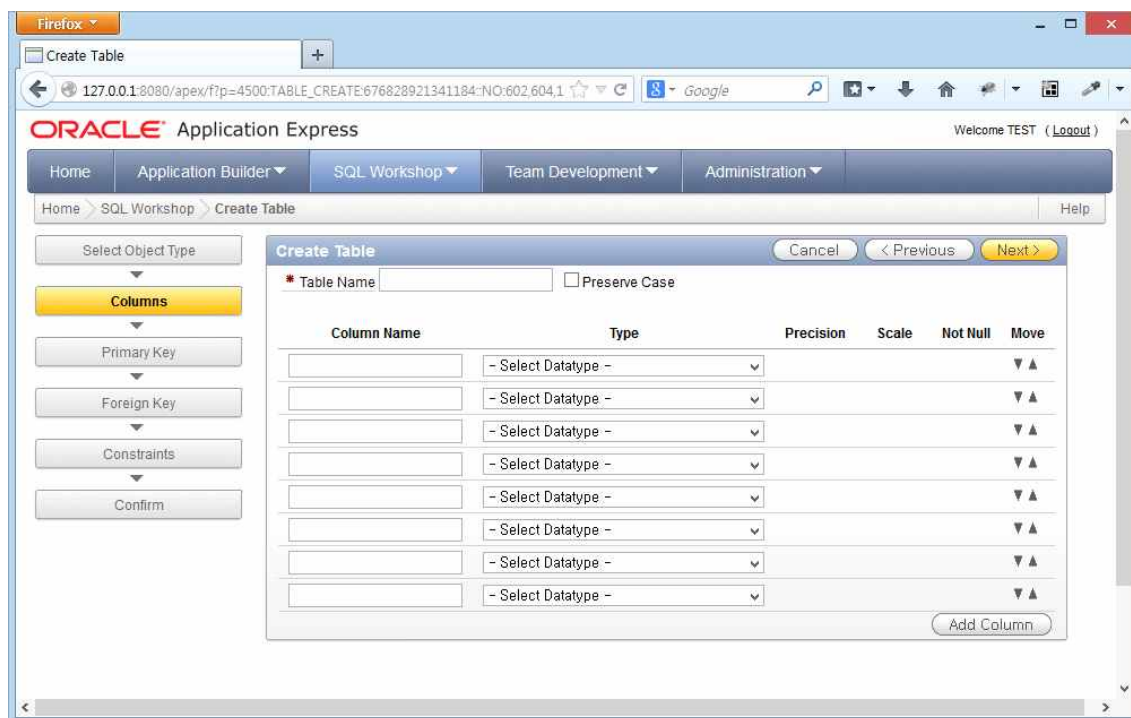
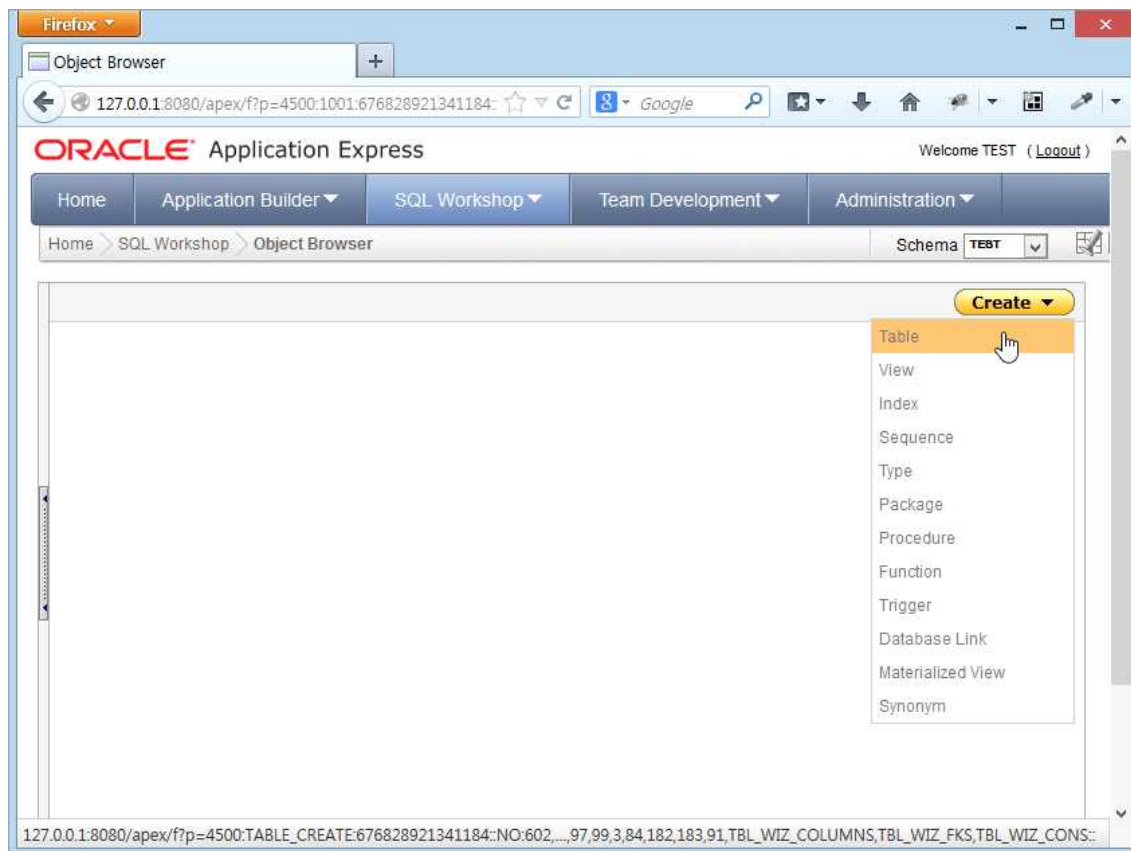
- Database User:** Radio buttons for 'Create New' (selected) and 'Use Existing'.
- * Database Username:** Text input field containing 'test'.
- * Application Express Username:** Text input field containing 'test'.
- * Password:** Password input field with masked characters (dots).
- * Confirm Password:** Password input field with masked characters (dots).

At the top right of the form area are 'Cancel' and 'Create Workspace' buttons. The 'Create Workspace' button is highlighted in red.



13. 테이블 생성

```
CREATE TABLE 테이블명 (  
    컬럼명 자료형 (제약조건),  
    .....  
)
```



■ 자료형

자료형의 종류	의 미
VARCHAR2(n)	가변 길이 문자 데이터 (1~4000byte)
CHAR(n)	고정 길이 문자 데이터(1~2000byte)
NUMBER(p,s)	전체 p자리 중 소수점 이하 s자리(p:1~38, s:-127까지)
DATE	날자 데이터(7byte)
LONG	가변 길이 문자 데이터(1~2GByte)
CLOB	단일 바이트 가변 길이 문자 데이터(1~4GByte)
BLOB	가변 길이 이진 데이터(1~4GByte)

■ 테이블 생성시 제약 조건

- 1) 부적절한 자료가 입력되는 것을 방지하기 위하여 CONSTRAINT를 명시
- 2) 제약 조건은 테이블 레벨에서 규칙을 적용함
- 3) 제약조건은 종속성이 존재할 경우 테이블 삭제를 방지함
- 4) 테이블에서 행이 삽입, 갱신, 삭제될 때마다 테이블에서 규칙을 적용
- 5) CONSTRAINT와는 별도로 각 컬럼들에게 제약 조건을부여할 수 있는데 이것을 컬럼레벨이라 함

■ 무결성 제약조건의 조건

제약조건	의 미
PRIMARY KEY	유일하게 테이블의 각 행을 구별(NOT NULL과 UNIQUE조건)
FOREIGN KEY	열과 참조된 열 사이의 외래키 관계를 적용(다른 테이블의 기본 키를 참조)
UNIQUE KEY	테이블의 모든 행을 유일하게 하는 값을 가지는 열(NULL 허용)
NOT NULL	열의 값이 NULL값을 포함 할 수 없다.
CHECK	참이어야 하는 조건을 지정 함

⑧ 회원 테이블을 생성해보세요.

아이디(ID) 정보

아이디(ID)	<input type="text"/> @hanmail.net 중복확인 3~15자의 영문소문자, 숫자, '-', '_', '.'만 가능합니다. (단, '.'은 아이디 맨 처음과 끝에 사용하지 않습니다.)
Daum이름(닉네임)	<input type="text"/> 1~8자의 한글, 2~16자의 영문대소문자, 숫자, '-', '_'만 가능합니다.
비밀번호	<input type="password"/> 6~32자의 영문, 숫자, 특수문자 조합으로 이용 가능합니다. ID와 개인정보, 추측하기 쉬운 비밀번호는 사용 불가능합니다. [도움말보기]
비밀번호 확인	<input type="password"/>

개인 정보

이름(한글)	<input type="text"/>
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여
생년월일	<input type="text"/> 년 <input type="text"/> 월 <input type="text"/> 일 <input checked="" type="radio"/> 양력 <input type="radio"/> 음력
연락처	휴대폰 010 - <input type="text"/> - <input type="text"/> 전화번호 <input type="text"/> 02 <input type="text"/> - <input type="text"/> - <input type="text"/> 휴대폰, 전화번호 중 하나만 입력하셔도 됩니다.
거주지역	<input type="button" value="국내"/> 국내, <input type="button" value="국외"/> 국외 버튼을 클릭하세요. <input type="text"/>

14. 테이블에 데이터 추가, 수정

■ INSERT 문

```
INSERT INTO 테이블명[(컬럼명)] VALUES (값)
```

⑨ 만든 테이블에 데이터를 입력해 보세요.

■ UPDATE 문

```
UPDATE 테이블명 SET 컬럼명 = 변경할 값,... WHERE 조건
```

⑩ 데이터를 변경해 보세요.

■ DELETE 문

```
DELETE FROM 테이블명 WHERE 조건
```

⑪ 데이터를 삭제해 보세요.

15. 조인(JOIN)

- 하나 이상의 테이블로부터 자료를 검색하기 위하여 조인을 사용함
- 기본키(Primary Key)와 외부키(Foreign Key)를 사용하여 Join하는 경우가 대부분

기본키란(Primary Key)?

관계 데이터베이스(RDB)에서 관계(데이터베이스 테이블) 내의 특정 튜플(열)을 일의적으로 식별할 수 있는 키 필드. 주 키(major key)라고도 한다. 파일에서 특정 레코드를 검색하거나 레코드들을 정렬할 때 우선적으로 참조된다. 관계 내의 키 필드가 하나밖에 없을 때에는 자동적으로 그 관계의 일차 키가 된다. 그러나 하나의 관계 내에 복수의 키가 있을 때에는 그중의 하나를 일차 키로 지정해야 한다.

외부키란(Foreign Key)?

관계를 맺고 있는 릴레이션 R1, R2에서 릴레이션 R1이 참조하고 있는 릴레이션 R2의 기본키와 같은 R1 릴레이션의 속성

외래키로 지정되면 참조 테이블의 기본키에 없는 값은 입력할 수 없음

- 조인은 보통 WHERE절에 조인 조건을 기술함
- 테이블을 조인하는 SELECT 문장을 작성할 경우 명확성을 위하여 또는 데이터베이스의 Performance향상을 위해 열 이름 앞에 테이블 명을 붙임
- 각 테이블에 똑 같은 필드 명이 존재하는 경우 반드시 필드명 앞에 테이블 명을 기술해야 한다.

■ EQUI JOIN

- 1) 가장 많이 사용되는 조인 방법으로 두 테이블에서 공통적으로 존재하는

칼럼의 값이 일치되는 행을 연결하여 결과를 생성하는 조인 방법

```
SELECT *  
FROM employees, departments  
WHERE employees.department_id = departments.department_id
```

■ 컬럼명의 모호성 해결

```
SELECT employee_id, first_name, last_name, employees.department_id,  
departments.department_id  
FROM employees, departments  
WHERE employees.department_id = departments.department_id
```

■ 테이블에 별칭 부여

```
SELECT employee_id, first_name, last_name, e.department_id, d.department_id  
FROM employees e, departments d  
WHERE e.department_id = d.department_id
```

⑩ 아래와 같이 출력하세요

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	CITY
100	Steven	King	90	90	Seattle
101	Neena	Kochhar	90	90	Seattle
102	Lex	De Haan	90	90	Seattle
103	Alexander	Hunold	60	60	Southlake
104	Bruce	Ernst	60	60	Southlake
105	David	Austin	60	60	Southlake
106	Valli	Pataballa	60	60	Southlake
107	Diana	Lorentz	60	60	Southlake

⑪ Seattle에서 근무하는 사원의 이름, 성, 부서번호를 출력하세요.

■ NON EQUI JOIN

- 1) 특정 범위 내에 있는지를 조사하기 위해 '=' 연산자 이외의 비교 연산자 사용

■ SELF JOIN

```
SELECT emp.first_name||' '||emp.last_name|| '의 매니저는 ' ||  
manager.first_name||' '||manager.last_name || '입니다.'  
FROM employees emp, employees manager  
WHERE emp.manager_id = manager.employee_id
```

■ OUTER JOIN

- 1) 두 테이블간에 일치한 값만 출력하는 EQUI JOIN과는 다르게 일치하지 않는 모든 자원들도 같이 출력하고자 할 때 Outer 조인을 사용함
- 2) 위 예제에서 사장은 결과가 나오지 않으나 표현하고 싶을 때 사용
- 3) Outer Join연산자(+)를 조인 조건에 사용시 조인 조건을 만족하지 않는 행들도 나타남

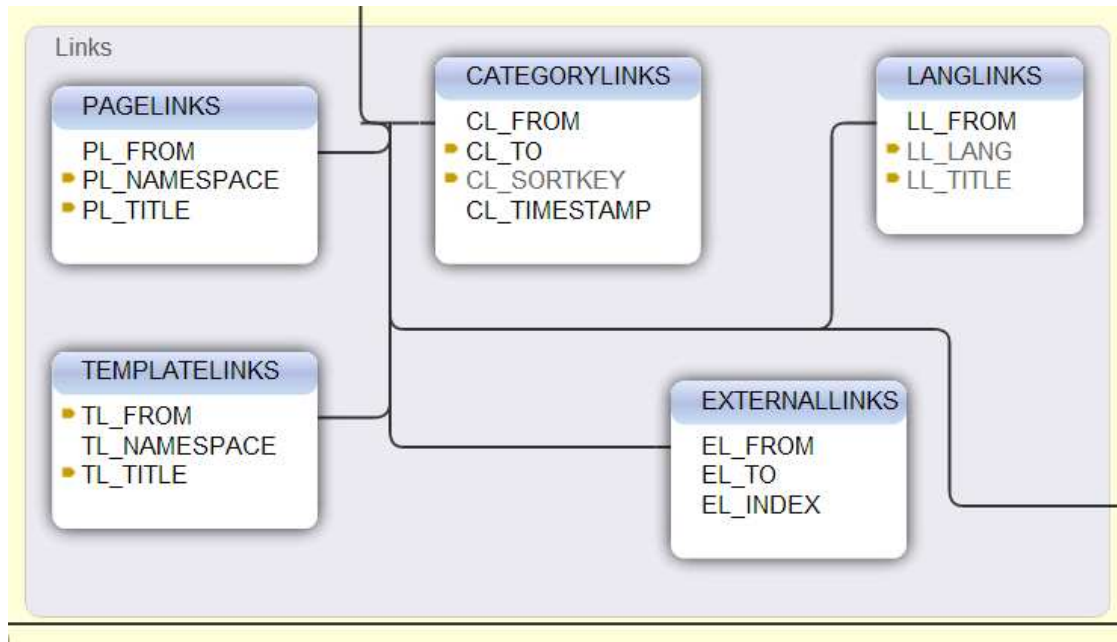
```
SELECT emp.first_name||' '||emp.last_name|| '의 매니저는 ' ||  
manager.first_name||' '||manager.last_name || '입니다.'  
FROM employees emp, employees manager  
WHERE emp.manager_id = manager.employee_id(+)
```

- 4) (+)연산자는 한 개이상의 NULL행을 생성하고, 정보가 충분한 테이블의 한 개이상의 행들이 NULL값에 조인됨

```
SELECT *  
FROM employees emp,departments dept  
WHERE dept.department_id = emp.department_id(+);
```

14. 데이터베이스 모델링

■ 스키마



■ 데이터베이스 단위



■ 데이터베이스의 구성요소

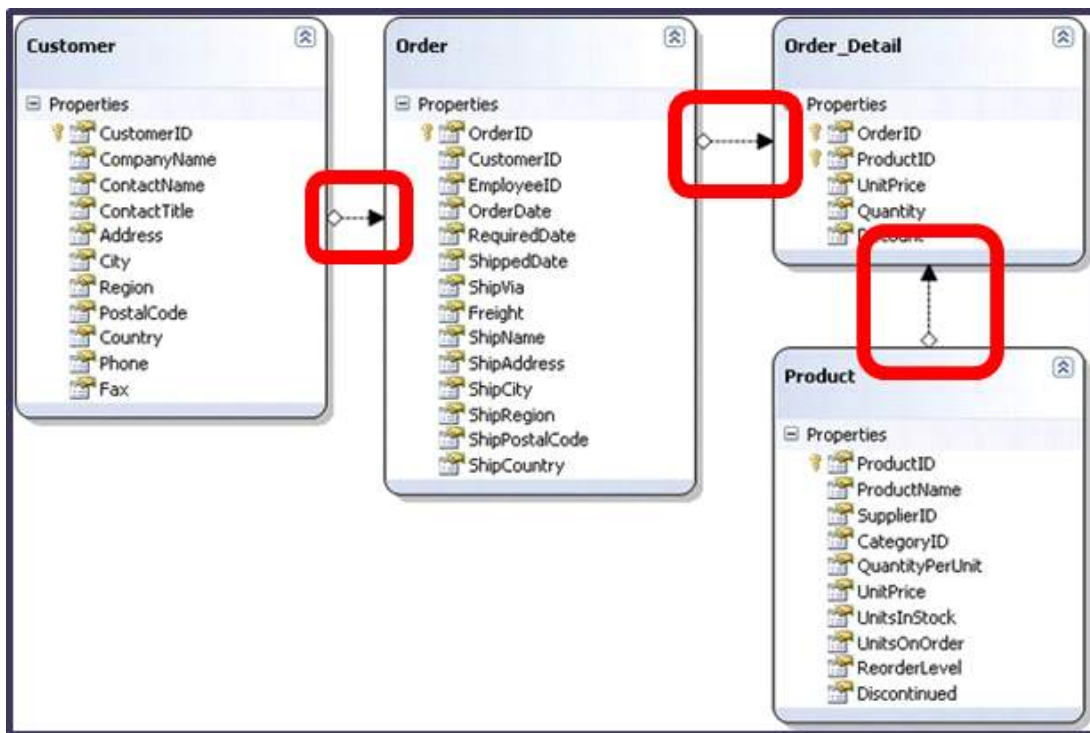
1) 개체(Entity)

번 호	이 름	나 이	키
1	김연아	50	190
2	손흥민	21	167
3	류현진	34	174
4	박지성	28	182

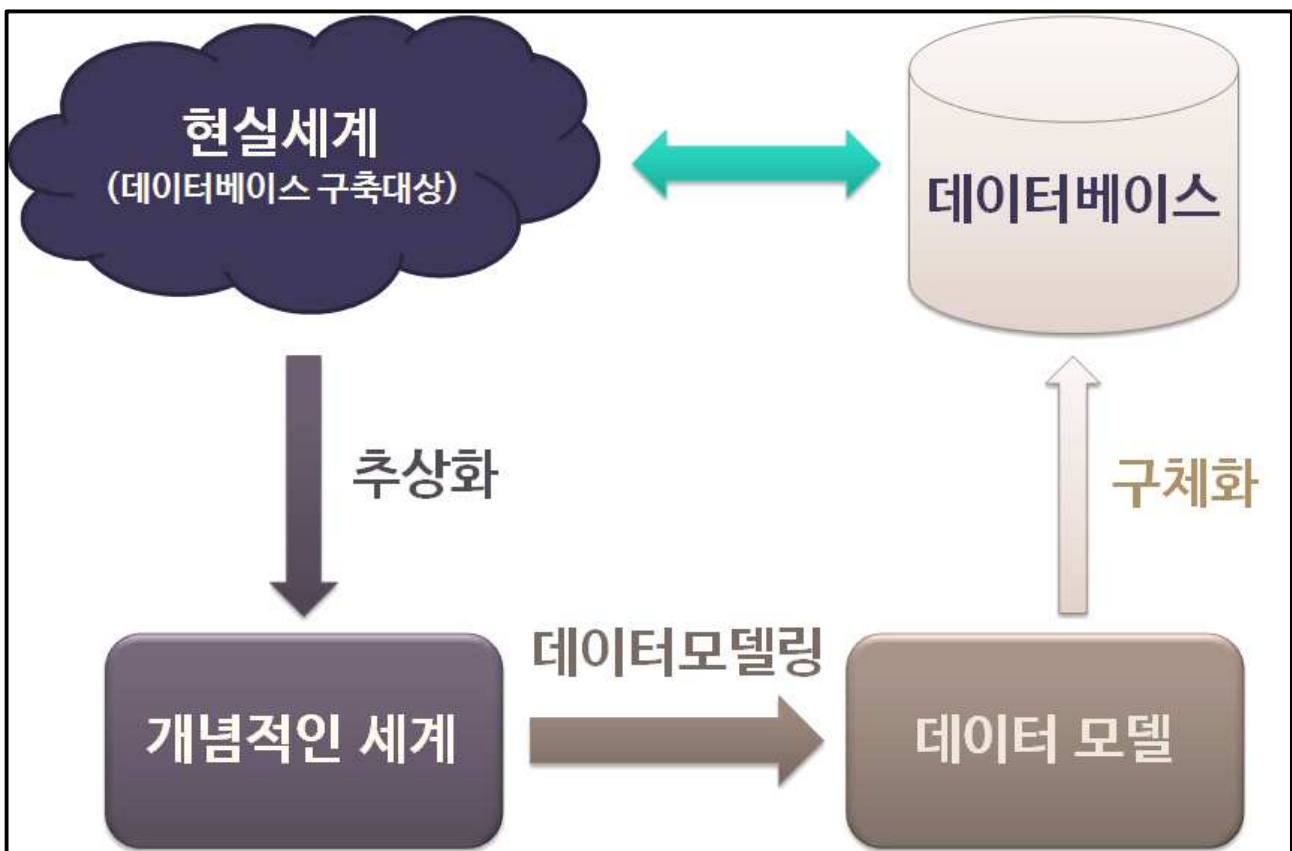
2) 속성(Attribute)

번 호	이 름	나 이	키
1	김연아	50	190
2	손흥민	21	167
3	류현진	34	174
4	박지성	28	182

■ 관계(Relationship)



■ 데이터모델링



■ 데이터모델링의 종류

개념적 데이터 모델링

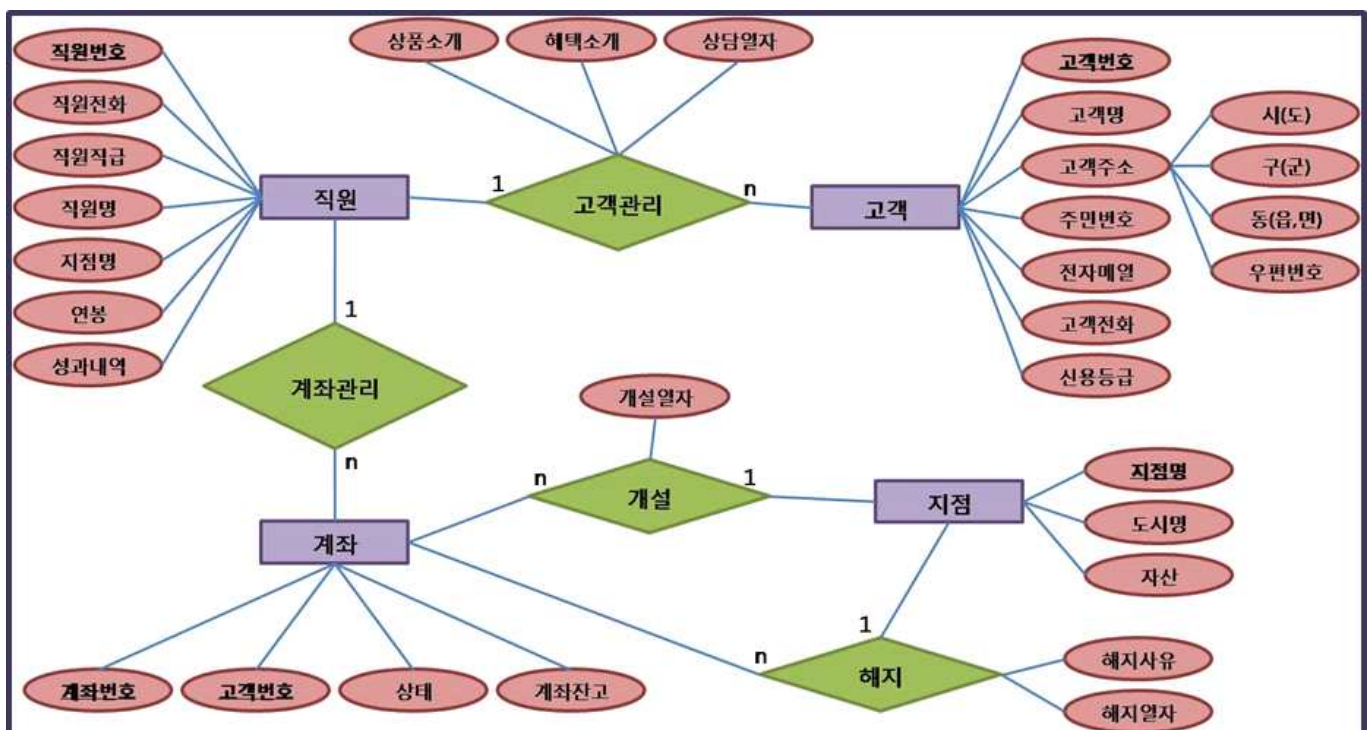


논리적 데이터 모델링

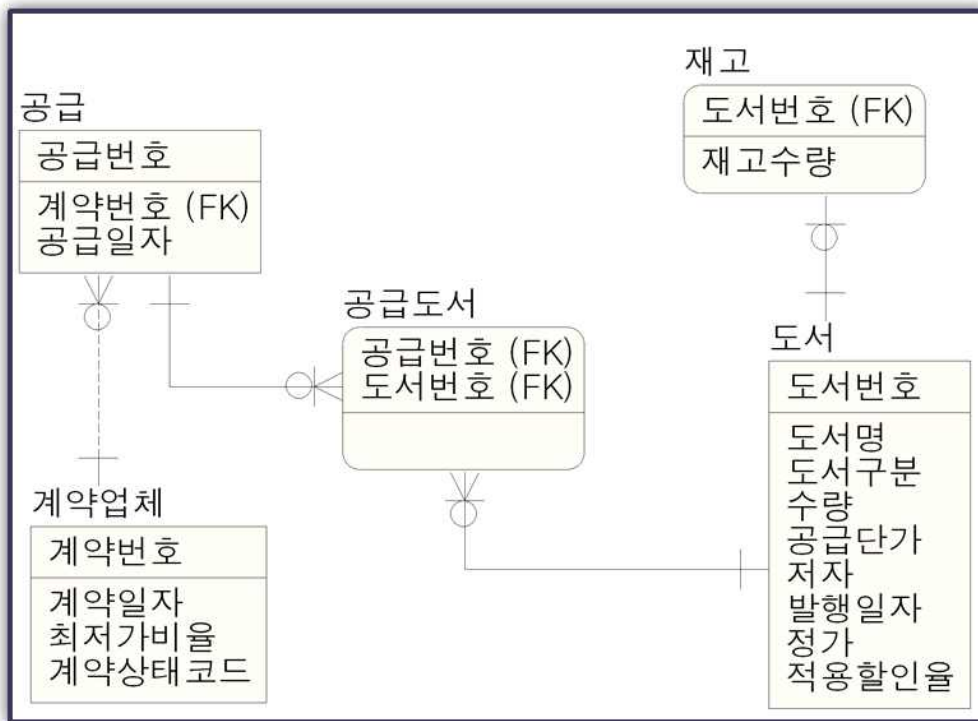


물리적 데이터 모델링

■ ER다이어그램



■ 논리적 데이터모델링



■ 물리적 데이터모델링

