



## 4장. SQL (1)

- ☐ SQL 언어
- ☐ SELECT 기초

## 4.1 SQL 언어

### □ SQL 개요

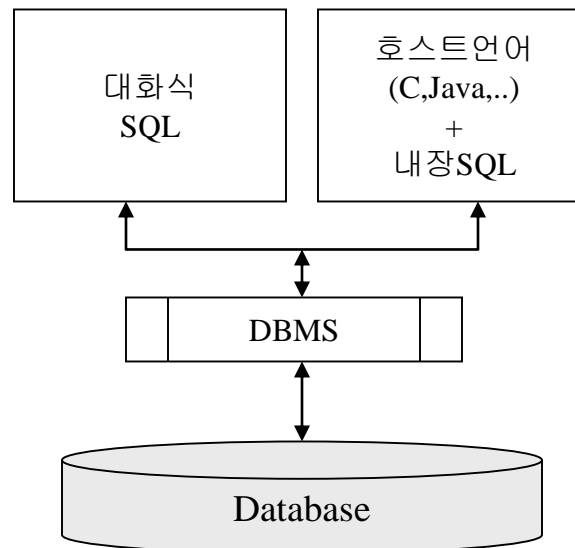
- 관계형 데이터베이스의 장점중의 하나는 사용자들이 쉽게 사용할 수 있는 SQL(Structured Query Language) 언어를 제공한다는 것
- 사용자는 간단한 SQL문을 사용하여 DBMS에게 작업을 요청
- 기본적인 SQL문들은 표준화되어 있기 때문에 거의 대부분의 DBMS 제품에 동일하게 적용
- SQL은 1974년 IBM 연구소에서 System R 프로젝트를 통해 개발되었고 1986년 미국 표준기구 ANSI에서 SQL 표준을 제정
- SQL 언어는 비절차적 언어(non-procedural language)
  - 사용자는 자신이 원하는 바만을 명시하며, 원하는 바를 DBMS가 어떻게 처리할지는 명시할 필요가 없음

## 4.1 SQL 언어

### □ SQL 개요

#### ➤ SQL의 두가지 사용 방식

- 대화식 SQL : DBMS 회사에서 제공하는 유틸리티 프로그램(예: ORACLE의 SQL\*Plus)을 이용하여 사용자가 직접 SQL문을 입력하고 실행결과를 확인하는 방식
- 내장 SQL : SQL문이 C, Java와 같은 프로그램 안에 포함되어져서 사용되는 방식



<그림> SQL을 이용하는  
두가지 방법

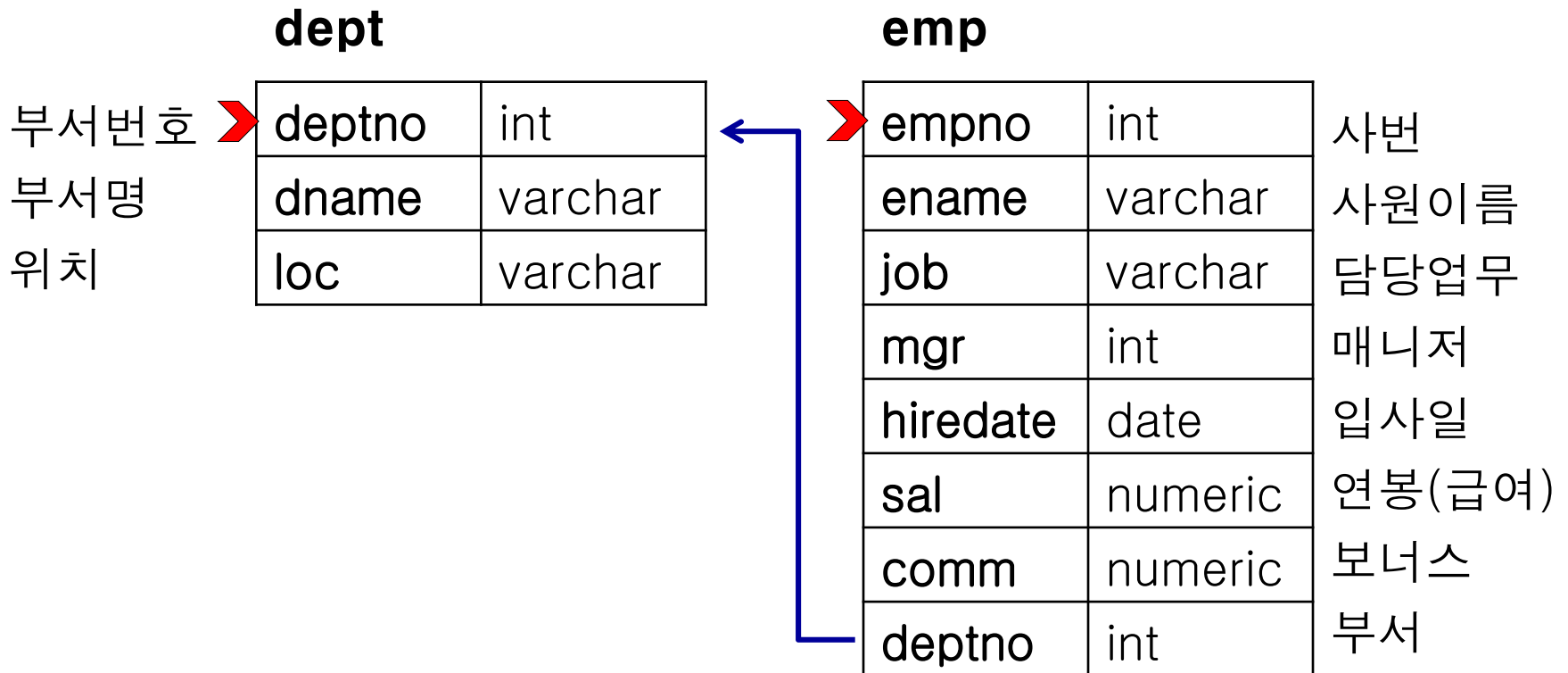
## 4.1 SQL 언어

### □ SQL 개요

- CREATE
- INSERT
- UPDATE
- DELETE
- SELECT
- DROP

## 4.1 SQL 언어

### □ 실습 테이블



## 4.1 SQL 언어

DEPT	+-----+			+-----+			+-----+		
	DEPTNO		DNAME		LOC				
	+-----+			+-----+			+-----+		
	10		ACCOUNTING		NEW YORK				
	20		RESEARCH		DALLAS				
	30		SALES		CHICAGO				
	40		OPERATIONS		BOSTON				
	+-----+			+-----+			+-----+		

EMP	+-----+								+-----+								+-----+							
	EMPNO	ENAME	JOB		MGR	HIREDATE	SAL		COMM	DEPTNO														
	+-----+																							
	7369	SMITH	CLERK		7902	1980-12-17	800.00		NULL	20														
	7499	ALLEN	SALESMAN		7698	1981-02-20	1600.00		300.00	30														
	7521	WARD	SALESMAN		7698	1981-02-22	1250.00		500.00	30														
	7566	JONES	MANAGER		7839	1981-04-02	2975.00		NULL	20														
	7654	MARTIN	SALESMAN		7698	1981-08-28	1250.00		1400.00	30														
	7698	BLAKE	MANAGER		7839	1981-05-01	2850.00		NULL	30														
	7782	CLARK	MANAGER		7839	1981-06-09	2450.00		NULL	10														
	7788	SCOTT	ANALYST		7566	1982-12-09	3000.00		NULL	20														
	7839	KING	PRESIDENT		NULL	1981-11-17	5000.00		NULL	10														
	7844	TURNER	SALESMAN		7698	1981-08-08	1500.00		0.00	30														
	7876	ADAMS	CLERK		7788	1983-01-12	1100.00		NULL	20														
	7900	JAMES	CLERK		7698	1981-12-03	950.00		NULL	30														
	7902	FORD	ANALYST		7566	1981-12-03	3000.00		NULL	20														
	7934	MILLER	CLERK		7782	1982-01-23	1300.00		NULL	10														
	+-----+																							

## 4.2 SELECT (기초)

### □ 기본 문법

```
select <컬럼 이름>  
from <테이블 이름>  
where <검색조건>
```

### □ 예

```
mysql> select sal  
-> from emp  
-> where empno = 7788 ;
```

## 4.2 SELECT (기초)

- ❑ 모든 부서의 이름을 보이시오

```
select dname  
from dept ;
```

- ❑ 'SALES' 부서는 어느 지역에 있는가

```
select loc  
from dept  
where dname = 'SALES';
```

- ❑ 사원번호가 7369 인 사원의 이름, 연봉, 담당 업무를 보이시오

```
select ename, sal, job  
from emp  
where empno = 7369;
```



## 4.2 SELECT (기초)

- ❑ 담당업무의 목록을 보이시오

<중복 포함>

```
select job  
from emp;
```

<중복 제거>

```
select distinct job  
from emp;
```

## 4.2 SELECT (기초)

- ❑ 'SCOTT' 사원의 모든 정보를 보이시오

```
select *  
from emp  
where ename = 'SCOTT';
```

- ❑ 담당업무가 SALESMAN 인 사원중에 연봉을 1500 보다 많이 받는 사람의 이름과 연봉을 보이시오

```
select ename, sal  
from emp  
where job = 'SALESMAN' and sal > 1500 ;
```

## 4.2 SELECT (기초)

- ❑ 담당업무가 CLERK 이거나 연봉을 3000 이하로 받는 사람의 이름과 연봉, 담당업무를 보이시오

```
select ename, sal, job  
from emp  
where job = 'CLERK' or sal <= 3000 ;
```

```
select ename, sal, job  
from emp  
where (job = 'CLERK') or (sal <= 3000) ;
```

비교연산자 : =, >, <, <=, >=, <>

## □ 값의 종류 (자료형)에 따른 비교 연산자 적용

- 숫자형 : `sal > 5000`
- 문자형 : `job <> 'clerk'`
- 날짜형 : `hiredate > '1981-05-01'`
- Null 은 컬럼의 자료형에 상관 없이 어디나 저장될 수 있다
  - `comm is null`
  - `comm is not null`

## 4.2 Select (기초)

### □ 조건절에 쓸 수 있는 연산자들

- 비교 연산자 : =, <>, >, >=, <, <=
- 기타
  - BETWEEN .. AND ..
  - NOT BETWEEN .. AND ..
  - IN ( , , )
  - NOT IN ( , , )
  - LIKE
  - %
  - \_
  - IS NULL
  - IS NOT NULL

## [실습과제 1] SQL문 작성하기

1. BOSTON 에 있는 부서의 이름은 무엇인가
2. 담당업무가 ANALYST 인 사원의 이름, 입사일자를 보이시오
3. 입사일자가 1981년 9월 1일 이후인 사원의 이름과 입사일자를 보이시오
4. 연봉이 3000 에서 4000 사이인 사원의 이름, 연봉을 보이시오
5. 사장의 이름을 보이시오 (힌트: 사장은 매니저가 없다)
6. 담당업무가 MANAGER 가 아닌 사원들의 이름, 담당업무를 보이시오
7. 담당업무가 CLERK, ANALYST, SALESMAN 인 사원들의 이름, 담당업무, 급여를 보이시오
8. 연봉이 3000 에서 4000 사이 이거나 담당업무가 MANAGER 인 사원의 이름, 연봉을 보이시오
9. 보너스를 받는 사원들의 이름과 보너스 금액을 보이시오
10. 부서번호가 20 인 사원 중에서 담당업무가 CLERK 이고 매니저의 사원번호가 7902 인 사람의 이름을 보이시오

## 4.3 기본 함수

❑ COUNT, MAX, MIN, AVG, SUM / IN, BETWEEN .. AND / LIKE

❑ 전체 사원은 모두 몇 명인가

```
select count(*)  
from emp ;
```

```
select count(*) as no_of_emp  
from emp ;
```

❑ 가장 높은 연봉의 금액은 얼마인가

```
select max(sal)  
from emp ;
```

## 4.3 기본 함수

- ❑ 담당업무가 CLERK 인 사원들의 연봉 합계를 보이시오

```
select sum(sal)
from emp
where job = 'CLERK' ;
```

- ❑ 모든 사원들의 현재 연봉 금액과, 연봉을 10% 올렸을 때 금액을 보이시오

```
select sal, sal * 1.1
from emp ;
```

```
select sal as before_sal,
       sal * 1.1 as after_sal
from emp ;
```



## 4.3 기본 함수

- 담당업무가 CLERK, ANALYST, SALESMAN 인 직원들의 이름, 담당업무, 급여를 보이시오

```
select ename, job, sal  
from emp  
where job in ('CLERK', 'ANALYST', 'SALESMAN');
```

- 연봉이 3000 에서 4000 사이인 직원의 이름, 연봉을 보이시오

```
select ename, sal  
from emp  
where sal between 3000 and 4000 ;
```

not in / not between .. and .. 도 사용 가능

## 4.3 기본 함수

- ❑ 사원 이름이 A 로 시작하는 사원들의 이름을 보이시오

```
select ename  
from emp  
where ename like 'A%' ;
```

- ❑ 이름에 IN 이 포함된 사원들의 이름을 보이시오

```
select ename  
from emp  
where ename like '%IN%' ;
```

- ❑ 이름의 세번째 글자가 A 인 사원들의 이름을 보이시오

```
select ename  
from emp  
where ename like '__A%' ;
```

## [실습과제 2] SQL문 작성하기

1. 담당 업무가 CLERK 인 사원은 모두 몇 명인가
2. 담당 업무가 CLERK 인 사원의 최고연봉 금액과 최저 연봉금액, 그리고 두 금액의 차이 금액을 보이시오.
3. 입사일이 1981년 9월 1일 이후인 사원의 평균 연봉을 보이시오
4. 매니저의 직원번호가 7698, 7639, 7782 이 아닌 사원의 이름, 담당업무를 보이시오
5. 1982년도와 1983년도에 입사한 사원의 이름, 입사일자를 보이시오
6. 담당업무 이름에 NA 를 포함하고 있는 사원의 이름, 담당업무를 보이시오
7. 보너스를 받는 사원의 이름과 보너스와 연봉을 합한 금액을 보이시오
8. 모든 사원의 연봉을 10% 올려주려면 현재보다 돈이 얼마나 더 있어야 하는지 보이시오

## 4.4 정렬, 그룹

- ❑ ORDER BY, GROUP BY .. HAVING
- ❑ 직원들의 이름을 알파벳 순 (오름차순)으로 보이시오

```
select ename  
from emp  
order by ename;
```

- ❑ 직원들의 이름을 알파벳 역순 (내림차순)으로 보이시오

```
select ename  
from emp  
order by ename desc;
```

asc : 오름차순 (기본), desc : 내림차순

## 4.4 정렬, 그룹

- ❑ 정렬 기준이 2개 이상인 경우
- ❑ 직원들의 부서번호와 담당업무를 보이되 먼저 부서번호로 정렬 (오름차순)하고 부서내에서는 담당업무명을 내림차순으로 정렬하시오

```
select deptno, job  
from emp  
order by deptno, job desc;
```

## 4.4 정렬, 그룹

- ❑ 직원들의 담당업무 목록을 보이시오 (담당업무에는 어떤 것들이 있는지)

```
select job  
from emp  
group by job;
```

- ❑ 각 담당업무별로 몇 명의 직원이 있는지 보이시오

```
select job, count(*)  
from emp  
group by job;
```

명령어 순서 : where .. group by .. order by

## 4.4 정렬, 그룹

- ❑ 담당업무와 담당업무별 인원수, 급여합계 를 보이되 PRESIDENT 는 제외하시오.( 담당업무별 인원수가 많은 순으로 출력하시오)

```
select job, count(*), sum(sal)
from emp
where job <> 'PRESIDENT'
group by job
order by count(*) desc;
```

group by 를 사용하게 되면 select 다음에 올 수 있는 것은

- 1) group by 에 서술된 컬럼이름
- 2) count, max,min, avg 와 같은 집계함수

## 4.4 정렬, 그룹

❑ 다음 SQL 문에서 잘못된 것은 ?

```
select job, ename, count(*), sum(sal)
from emp
group by job;
```

```
select job, count(*), sum(sal)
from emp
group by job
order by hiredate, sal;
```

order by 를 group by 와 함께 사용하는 경우 order by 다음에 올 수 있는 것은 select 에 있는 컬럼이름



## 4.4 정렬, 그룹

- ❑ 담당업무와 담당업무별 인원수, 연봉합계 를 보이되 PRESIDENT 는 제외하시오. 그리고 급여합계가 4000 이상인 경우만 보이시오 ( 담당업무별 인원수가 많은 순으로 출력하시오)

```
select job, count(*), sum(sal)
from emp
where job <> 'PRESIDENT'
group by job
having sum(sal) >= 4000
order by count(*) desc;
```

having 은 만들어진 그룹에 적용되는 조건을 지정할때 사용  
where 는 각각의 행(row)에 적용되는 조건을 지정할때 사용

## [실습과제 3] SQL문 작성하기

1. 모든 사원들의 이름을 알파벳 역순으로 보이시오
2. 모든 담당업무명을 알파벳 순으로 보이시오
3. 모든 사원의 이름과 업무, 급여를 급여순으로 보이시오
4. 모든 사원의 이름과 부서번호를 입사일자순으로 보이시오
5. 모든 사원의 사원번호, 이름, 급여를 부서번호순으로 보이시오
6. 담당 업무별 평균 연봉을 보이시오
7. 부서별 평균 연봉을 보이되 많은 연봉 우선으로 보이시오
8. 각 부서별 최고 연봉을 보이시오



## [실습과제 3] SQL문 작성하기

9. 모든 사원의 정보를 담당업무에 대해 오름차순으로, 그리고 같은 업무 내에서는 연봉 내림차순으로 정렬하여 보이시오
10. 각 매니저별 담당 사원의 인원수, 평균 연봉을 보이시오
11. 각 담당업무별로 최고연봉과 최저 연봉을 보이시오
12. 담당업무별 인원수, 평균연봉 을 보이되 업무별 인원수가 3명 이상인 경우에 대해서만 보이시오
13. 각 부서별 부서번호, 사원수, 연봉합계를 보이시오
14. 각 부서별 부서번호, 사원수, 연봉합계를 보이되 사원수가 3명 이상인 부서의 정보만 보이시오.
15. 부서번호, 부서별 사원수, 연봉합계 를 보이되 급여합계가 4000~5000 사이인 경우만 보이시오 . 단, 입사일자가 1981-01-01 이전 사원은 제외하고, 연봉합계가 많은 순으로 출력하시오