


SQL 기본



배재대학교 컴퓨터공학과
김 창 수

Contents

 데이터 조회(SELECT 문)

 조건검색(WHERE 절)

 정렬(ORDER BY절)

 집합연산자

데이터 조회

❖ SELECT 명령문

- 테이블에 저장된 데이터를 검색하기 위한 명령문
- SELECT 명령문에서 SELECT절과 FROM절은 필수절

❖ 사용법

```
SELECT          [DISTINCT]{* | columnlist  }  
FROM            table_name;
```

데이터 조회

❖ SQL 명령문의 표준 형식

SELECT	[DISTINCT] { * column[alias]....}
FROM	table
[WHERE	condition]
[GROUP BY	group_by_expression]
[HAVING	group_condition]
[ORDER BY	column];

❖ SELECT 명령문 작성시 규칙

- SQL 명령문에서 대소문자를 구분하지 않음
 - 키워드는 주로 대문자로 사용하고 테이블명, 칼럼 이름은 소문자로 작성하는 것을 권장
- 절은 줄을 구분하여 작성하고, 들여쓰기를 사용하여 읽기 쉽게
- 테이블명, 칼럼 이름, 키워드(SELECT, FROM, WHERE 등)는 축약할 수 없음

데이터 조회

❖ 기본적인 SQL 질의

- SELECT절
 - 질의 결과를 포함 하려는 애트리뷰트들의 리스트 열거
 - DISTINCT 절을 사용해서 중복 제거
- FROM절
 - 질의에서 필요로 하는 릴레이션들의 리스트 열거
- WHERE절
 - 관계 대수의 선택(Select) 연산의 조건에 해당
- 중첩질의
 - 다른 질의의 WHERE절에 포함된 SELECT문
- GROUP BY절
 - 동일한 값을 갖는 튜플들을 한 그룹으로 묶는다
- HAVING절
 - 튜플들의 그룹이 만족해야 하는 조건
- ORDER BY절
 - 결과 튜플들의 정렬 순서

데이터 조회

❖ SQL 문장과 절

- 문장으로 구성, 문장의 끝은 세미콜론(;)

```
select *
from dept;
```

- 대소문자 구분하지 않음

```
SELECT *
FROM DEPT;
```

데이터 조회

❖ 특정 데이터만 보기

전체 컬럼 출력

```
select *
from EMP;
```

R2	EMPNO	R2	ENAME	R2	JOB	R2	MGR	R2	HIRED...	R2	SAL	R2	COMM	R2	DEPTNO
	1001		김사랑		사원		1013		07/03/01		300		(null)		20
	1002		한예슬		대리		1005		07/04/02		250		80		30
	1003		오지호		과장		1005		05/02/10		500		100		30
	1004		이병헌		부장		1008		03/09/02		600		(null)		20
	1005		신동협		과장		1005		05/04/07		450		200		30
	1006		장동건		부장		1008		03/10/09		480		(null)		30
	1007		이문세		부장		1008		04/01/08		520		(null)		10
	1008		감우성		차장		1003		04/03/08		500		0		30
	1009		안성기		사장		(null)		96/10/04		1000		(null)		20
	1010		이병헌		과장		1003		05/04/07		500		(null)		10
	1011		조항기		사원		1007		07/03/01		280		(null)		30
	1012		강혜정		사원		1006		07/08/09		300		(null)		20
	1013		박중훈		부장		1003		02/10/09		560		(null)		20
	1014		조인성		사원		1006		07/11/09		250		(null)		10

사원번호, 사원명만 출력

```
select empno,ename
from EMP;
```

R2	EMPNO	R2	ENAME
	1001		김사랑
	1002		한예슬
	1003		오지호
	1004		이병헌
	1005		신동협
	1006		장동건
	1007		이문세
	1008		감우성
	1009		안성기
	1010		이병헌
	1011		조항기
	1012		강혜정
	1013		박중훈
	1014		조인성

데이터 조회

❖ 산술연산자

- 칼럼 값에 산술 연산자를 적용하여 계산된 결과를 출력할 수 있는 기능을 제공
- 숫자 또는 날짜 타입에만 사용 가능
- 수학에서의 산술 연산자의 우선순위와 동일, 괄호를 사용하여 우선 순위 변경 가능

데이터 조회

❖ 사용 예

- 사원 테이블에서 사원의 이름과 급여, 연봉을 출력

```
select ename, sal, sal*12
from emp;
```

EMPNO	ENAME	SAL	SAL*12
7369	김사랑	300	3600
7401	한예슬	250	3000
7520	오지호	500	6000
7566	이병헌	600	7200
7654	신동협	450	5400
7698	장동건	480	5760
7782	이문세	520	6240
7844	감우성	500	6000
7876	안성기	1000	12000
7902	이병헌	500	6000
7934	조향기	280	3360
7969	강혜정	300	3600
8000	박중훈	560	6720
8050	조인성	250	3000

데이터 조회

❖ NULL 개념

- NULL은 미확인 값이나 아직 적용되지 않은 값을 의미
- 연산, 할당, 비교가 불가능
- 예
 - 사원 테이블에서 커미션(comm)이 NULL인 경우 그 사원의 커미션이 0이라는 의미가 아니라 현재 시점에서 그 사원의 커미션을 모른다는 의미

EMPNO	ENAME	JOB	MGR	HIRED...	SAL	COMM	DEPTNO
1001	김사랑	사원	1013	07/03/01	300	(null)	20
1002	한예슬	대리	1005	07/04/02	250	80	30
1003	오지호	과장	1005	05/02/10	500	100	30
1004	이병헌	부장	1008	03/09/02	600	(null)	20
1005	신동협	과장	1005	05/04/07	450	200	30
1006	장동건	부장	1008	03/10/09	480	(null)	30
1007	이문세	부장	1008	04/01/08	520	(null)	10
1008	감우성	차장	1003	04/03/08	500	0	30
1009	안성기	사장	(null)	96/10/04	1000	(null)	20
1010	이병헌	과장	1003	05/04/07	500	(null)	10
1011	조항기	사원	1007	07/03/01	280	(null)	30
1012	강혜정	사원	1006	07/08/09	300	(null)	20
1013	박종훈	부장	1003	02/10/09	560	(null)	20
1014	조인성	사원	1006	07/11/09	250	(null)	10

데이터 조회

❖ 산술식과 함수에서 NULL 처리

- NULL에 산술식이나 함수를 적용하면 연산 결과도 NULL

❖ 사용 예

- 사원 테이블에서 연봉에 커미션을 더한 값을 출력하여라.

ENAME	SAL	JOB	SAL*12	SAL*12+COMM	COMM	DEPTNO
김사랑	300	사원	3600	(null)	(null)	20
한예슬	250	대리	3000	3080	80	30
오지호	500	과장	6000	6100	100	30
이병현	600	부장	7200	(null)	(null)	20
신동협	450	과장	5400	5600	200	30
장동건	480	부장	5760	(null)	(null)	30
이문세	520	부장	6240	(null)	(null)	10
감우성	500	차장	6000	6000	0	30
안성기	1000	사장	12000	(null)	(null)	20
이병현	500	과장	6000	(null)	(null)	10
조항기	280	사원	3360	(null)	(null)	30
강혜정	300	사원	3600	(null)	(null)	20
박중훈	560	부장	6720	(null)	(null)	20
조인성	250	사원	3000	(null)	(null)	10

데이터 조회

❖ NVL 변환 함수 : NVL

- NVL 함수는 NULL을 0 또는 다른 값으로 변환하기 위한 함수

❖ 사용법

```
NVL(expression1, expression2)
```

- expression1 : NULL을 포함하는 컬럼 또는 표현식
- expression2 : NULL을 대체하는 값
- 주의 : expression1과 expression2는 반드시 동일한 데이터 타입이어야한다.

데이터 조회

❖ 사용 예

- 커미션이 NULL인 경우에는 커미션을 0으로 계산한다.

R2	ENAME	R2	SAL	R2	COMM	R2	SAL*12+COMM	R2	NVL(COMM,0)	R2	SAL*12+NVL(COMM,0)
	김사랑		300		(null)		(null)		0		3600
	한예슬		250		80		3080		80		3080
	오지호		500		100		6100		100		6100
	이병현		600		(null)		(null)		0		7200
	신동협		450		200		5600		200		5600
	장동건		480		(null)		(null)		0		5760
	이문세		520		(null)		(null)		0		6240
	감우성		500		0		6000		0		6000
	안성기		1000		(null)		(null)		0		12000
	이병현		500		(null)		(null)		0		6000
	조항기		280		(null)		(null)		0		3360
	강혜정		300		(null)		(null)		0		3600
	박중훈		560		(null)		(null)		0		6720
	조인성		250		(null)		(null)		0		3000

데이터 조회

❖ 컬럼에 별명 부여 방법

- 컬럼 이름과 별명 사이에 공백을 추가하는 방법
- 컬럼 이름과 별명 사이에 AS 키워드를 추가하는 방법
- 큰따옴표를 사용하는 방법
 - 컬럼 이름과 별명 사이에 공백을 추가하는 경우
 - 특수문자를 추가하거나 대소문자를 구분하는 경우

데이터조회

❖ 사용 예

- 사원 테이블에서 사원 이름 컬럼의 별명은 **사원명**, 연봉계산 컬럼의 별명은 **연봉**으로 부여하여 출력하여라

```
SELECT ename as 사원명, sal*12+nvl(comm, 0) as 연봉
FROM emp;
```

사원명	연봉
김사향	3600
한예슬	3080
오지호	6100
이병헌	7200
신동협	5600
장동건	5760
이문세	6240
감우성	6000
안성기	12000
이병헌	6000
조항기	3360
강혜정	3600
박종훈	6720
조인성	3000

데이터조회

❖ 사용 예

- 사원 테이블에서 사원 이름 칼럼의 별명은 **사원 명**, 연봉계산 칼럼의 별명은 **연_봉**으로 부여하여 출력하여라

```
SELECT ename as "사원 명", sal*12+nvl(comm, 0) as "연_봉"
FROM emp;
```

사원 명	연_봉
김사향	3600
한예슬	3080
오지호	6100
이병헌	7200
신동협	5600
장동건	5760
이문세	6240
감우성	6000
안성기	12000
이병헌	6000
조항기	3360
강혜정	3600
박종훈	6720
조인성	3000

별명에 대소문자를 구분하거나
공백 및 특수문자를 사용하는
경우에는 “ ”를 사용한다

데이터조회

❖ 합성(concatenation)연산자 (||)

- 하나의 칼럼과 다른 칼럼, 산술 표현식 또는 상수 값과 연결하여 하나의 칼럼처럼 출력할 경우에 사용

❖ 사용 예

```
SQL> SELECT ename || '의 직급은 ' || job || '입니다.' as 직급
2 FROM emp;
```

직급

김사	랑의	직	급	은	사	원	입	니	다.	
한예	슬의	직	급	은	대	리	입	니	다.	
오지	호의	직	급	은	과	장	입	니	다.	
이병	현의	직	급	은	부	장	입	니	다.	
신동	현의	직	급	은	과	장	입	니	다.	
장동	건의	직	급	은	부	장	입	니	다.	
이무	세의	직	급	은	부	장	입	니	다.	
감우	성의	직	급	은	차	장	입	니	다.	
안성	기의	직	급	은	사	장	입	니	다.	
이병	현의	직	급	은	과	장	입	니	다.	
조향	기의	직	급	은	사	원	입	니	다.	
강혜	정의	직	급	은	사	원	입	니	다.	
박중	훈의	직	급	은	사	부	장	입	니	다.
조인	성의	직	급	은	사	원	입	니	다.	

데이터조회

❖ DISTINCT

■ 중복데이터 제거

```
SELECT deptno
FROM emp;
```

```
SQL> select deptno
      2 from emp;
```

DEPTNO
20
30
30
20
30
30
10
30
20
10
30
20
20
10

```
SELECT distinct deptno
FROM emp;
```

```
SQL> select distinct deptno
      2 from emp;
```

DEPTNO
30
20
10

WHERE 절을 이용한 조건 검색

❖ WHERE절

- 테이블에 저장된 데이터중에서 원하는 데이터만 선택적으로 검색하는 기능
- WHERE 절의 조건문은 칼럼 이름, 연산자, 상수, 산술 표현식을 결합하여 다양한 형태로 표현 가능
- WHERE 절에서 사용하는 데이터 타입은 문자, 숫자, 날짜 타입 사용 가능
- 문자와 날짜 타입의 상수 값은 작은 따옴표(')로 묶어서 표현하고 숫자는 그대로 사용
- 상수 값에서 영문자는 대소문자를 구별

❖ 사용법

```
SELECT          [DISTINCT]{* | column[alias]...}  
FROM            table  
[WHERE          condition];
```

WHERE 절을 이용한 조건 검색

❖ 사용 예

- 사원 테이블에서 급여가 500이상인 사원만 검색하여 사번, 이름, 직급, 급여, 부서 번호를 출력하여라

```
SQL> select empno, ename, job, sal, deptno
2  from emp
3  where sal >= 500;
```

EMPNO	ENAME	JOB	SAL	DEPTNO
1003	오지 호	과장	500	30
1004	이병헌	부장	600	20
1007	이문세	부장	520	10
1008	감우성	차장	500	30
1009	안성기	사장	1000	20
1010	이병헌	과장	500	10
1013	박중훈	부장	560	20

7 개의 행이 선택되었습니다.

비교 연산자를 사용한 조건 검색

❖ 비교 연산자

- WHERE 절에서 숫자, 문자, 날짜의 크기나 순서를 비교하는 연산자

연산자	의미
=	같다
!=, <>	같지 않다
>	크다
>=	크거나 같다
<	작다
<=	작거나 같다

비교 연산자를 사용한 조건 검색 예

❖ 사용 예

- 사원 테이블에서 10번 부서에 소속된 사원만 검색하여 사번, 이름, 직급, 급여, 부서번호를 출력하여라.

```
SQL> select empno, ename, job, sal, deptno  
2  from emp  
3  where deptno = 10;
```

EMPNO	ENAME	JOB	SAL	DEPTNO
1007	이문세	부장	520	10
1010	이병헌	과장	500	10
1014	조인성	사원	250	10

논리 연산자를 이용한 조건 검색

❖ 논리 연산자

- WHERE 절에서 여러 개의 조건을 결합할 경우
- AND, OR, NOT과 같은 논리 연산자를 사용

연산자	의미
AND	모든 조건이 참일 때, 참 값을 반환
OR	모든 조건 중에서 하나가 참일 때, 참 값을 반환
NOT	조건과 반대되는 결과를 반환

AND 논리 연산자를 이용한 조건 검색 예

❖ 사용 예

- 사원 테이블에서 30번 부서 소속 사원 중에서 직급이 과장인 사원만 검색하여 사번, 이름, 직급, 급여, 부서번호를 출력하여라.

```
SQL> select empno, ename, job, sal, deptno  
2  from emp  
3  where deptno = 30 and job='과장';
```

EMPNO	ENAME	JOB	SAL	DEPTNO
1003	오지호	과장	500	30
1005	신동협	과장	450	30

OR 논리 연산자를 이용한 조건 검색 예

❖ 사용 예

- 사원 테이블에서 30번 부서 소속 사원 또는 직급이 과장인 사원만 검색하여 사번, 이름, 직급, 급여, 부서번호를 출력하여라.

```
SQL> select empno, ename, job, sal, deptno  
2  from emp  
3  where deptno = 30 or job='과장';
```

EMPNO	ENAME	JOB	SAL	DEPTNO
1002	한예슬	대리	250	30
1003	오지호	과장	500	30
1005	신동협	과장	450	30
1006	장동건	부장	480	30
1008	감우성	차장	500	30
1010	이병헌	과장	500	10
1011	조항기	사원	280	30

NOT 논리 연산자를 이용한 조건 검색 예

❖ 사용 예

- NOT 연산자를 이용하여 사원 테이블에서 30번 부서 소속 사원을 제외한 사원만 검색하여 사번, 이름, 직급, 급여, 부서번호를 출력하라.

```
SQL> select empno, ename, job, sal, deptno  
2  from emp  
3  where not deptno = 30;
```

EMPNO	ENAME	JOB	SAL	DEPTNO
1001	김 사랑	사원	300	20
1004	이 병헌	부장	600	20
1007	이 문세	부장	520	10
1009	안 성기	사장	1000	20
1010	이 병헌	과장	500	10
1012	강 혜정	사원	300	20
1013	박 중훈	부장	560	20
1014	조 인성	사원	250	10

SQL 연산자를 이용한 조건 검색

❖ SQL 연산자

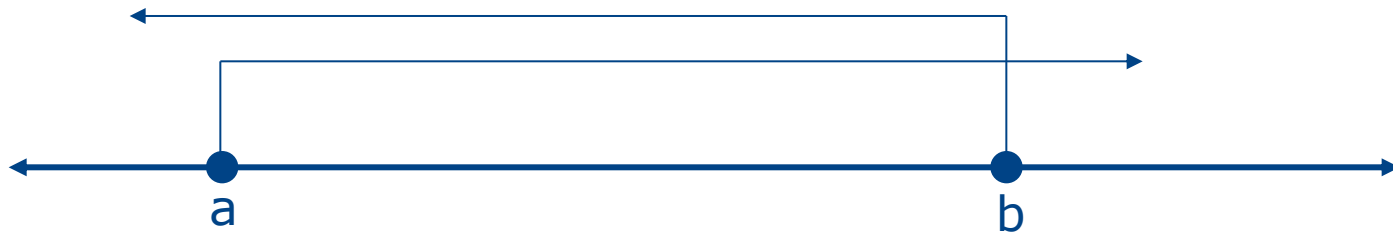
- SQL 연산자는 SQL 언어에만 제공
- SQL 연산자는 모든 데이터 타입에 대해 사용 가능

연산자	의미
BETWEEN a AND b	a와 b의 사이의 값, a, b 도 포함
IN(a, b, c,..., n)	a, b, c, ..., n 중의 하나와 일치하면 참
LIKE	문자 패턴과 부분적으로 일치(% , _)하면 참
IS NULL	NULL 이면 참

BETWEEN 연산자를 이용한 조건 검색

❖ BETWEEN 연산자

- 특정 칼럼의 데이터 값이 하한 값 a와 상한 값 b사이에 포함되는 행을 검색하기 위한 연산자
- AND 논리 연산자를 사용한 것과 같은 결과 집합 출력



BETWEEN 연산자를 이용한 조건 검색 예

❖ 사용 예

- BETWEEN 연산자를 사용하여 급여가 400에서 500사이인 사원의 사번, 이름, 급여를 출력하여라.

```
SQL> select empno, ename, sal  
2  from emp  
3  where sal between 400 and 500;
```

EMPNO	ENAME	SAL
-----	-----	-----
1003	오지 호	500
1005	신동협	450
1006	장동건	480
1008	감우성	500
1010	이병현	500

IN 연산자를 이용한 조건 검색

❖ $IN(a, b, c, \dots)$ 연산자

- $IN(a, b, c, \dots)$ 연산자는 특정 칼럼의 데이터 값이 a, b, c, \dots 값 중에 하나라도 일치하면 참이 되는 연산자
- OR 논리 연산자를 사용한 것과 같은 결과 집합 출력



IN 연산자를 이용한 조건 검색 예

❖ 사용 예

- IN 연산자를 사용하여 커미션이 80이거나 100이거나 200인 사원의 사번, 이름, 급여, 커미션을 출력하여라

```
SQL> SELECT empno, ename, sal, comm  
2 FROM emp  
3 WHERE comm in(80, 100, 200);
```

EMPNO	ENAME	SAL	COMM
1002	한예슬	250	80
1003	오지호	500	100
1005	신동협	450	200

LIKE연산자를 이용한 조건 검색

❖ LIKE 연산자

- 컬럼에 저장된 문자열중에서 LIKE 연산자에서 지정한 문자 패턴과 부분적으로 일치하면 참이 되는 연산자
- LIKE 다음에는 pattern을 기술해야 하는데 pattern에 다음과 같이 두 가지 와일드카드가 사용됩니다.

와일드카드	의미
%	문자가 없거나, 하나 이상의 문자가 어떤 값이 와도 상관없다
_	하나의 문자가 어떤 값이 와도 상관없다

LIKE연산자를 이용한 조건 검색

❖ 사용 예

- 사원 테이블에서 성이 '이' 씨인 사원을 출력하여라.

```
SQL> SELECT *  
2 FROM emp  
3 WHERE ename = '이';
```

선택된 레코드가 없습니다.

```
SQL> SELECT empno, ename, sal, comm  
2 FROM emp  
3 WHERE ename like '이%';
```

EMPNO	ENAME	SAL	COMM
1004	이병헌	600	
1007	이문세	520	
1010	이병헌	500	

ESCAPE 옵션

❖ ESCAPE 옵션

- ESCAPE 옵션은 LIKE 연산자에서 사용하는 와일드 문자(% , _) 자체를 포함하는 문자열을 검색할 때, 와일드 문자를 일반 문자처럼 취급하기 위한 옵션

ESCAPE 옵션

❖ 사용 예

- 이름이 세 번째 글자가 "성"이고, 그 뒤는 무엇이 오든 관계없는 사원의 사번, 이름, 급여를 출력하여라

```
SQL> SELECT empno, ename, sal, comm  
2 FROM emp  
3 WHERE ename like '__성%';
```

EMPNO	ENAME	SAL	COMM
1008	감우성	500	0
1014	조인성	250	

NULL 연산자를 이용한 조건 검색

❖ IS NULL

- IS NULL 연산자는 컬럼 값 중에서 NULL을 포함하는 행을 검색하기 위해 사용하는 연산자

❖ IS NOT NULL 연산자

- IS NOT NULL 연산자는 NULL이 아닌 행을 검색 경우에 사용하는 연산자

NULL 연산자를 이용한 조건 검색 예

❖ 사용 예

- 사원 테이블에서 커미션이 없는 사원의 이름, 직급, 커미션을 출력 하여라.

```
SQL> select ename, job, comm
2  from emp
3  where comm=null;
```

comm칼럼에 'NULL' 문자열이 있는지
비교 검색

선택된 레코드가 없습니다.

```
SQL> select ename, job, comm
2  from emp
3  where comm is null;
```

comm칼럼 값이 NULL인 행 검색

ENAME	JOB	COMM
김 사랑	사원	
이병헌	부장	
장동건	부장	
이문세	부장	
안성기	사장	
이병헌	과장	
조항기	사원	
강혜정	사원	
박중훈	부장	
조인성	사원	

연산자 우선순위

우선순위	연산자
1	비교 연산자(=, !=, ^=, <>, <, <=, >, >=) SQL 연산자(BETWEEN, IN, LIKE, IS NULL)
2	NOT
3	AND
4	OR

연산자 우선순위 예

❖ 사용 예

- 30번 부서의 사원 중에서 직급이 과장 또는 사원인 사원의 이름, 직급, 부서 번호를 출력하여라.

```
SQL> select ename, job, deptno  
2  from emp  
3  where deptno=30  
4  and (job='과장'  
5  or job='사원');
```

ENAME	JOB	DEPTNO
오지호	과장	30
신동협	과장	30
조향기	사원	30

연산자 우선 순위 예

❖ 사용 예

- 30번 부서의 직원 중에서 과장이거나 부서에 상관없이 직원인 직원의 이름, 직급, 부서 번호를 출력하여라.

```
SQL> select ename, job, deptno  
2   from emp  
3   where deptno=30  
4   and job='과장'  
5   or job='직원';
```

ENAME	JOB	DEPTNO
김사랑	직원	20
오지호	과장	30
신동협	과장	30
조향기	직원	30
강혜정	직원	20
조인성	직원	10

정렬

❖ 정렬(sorting)

- SQL 명령문에서 검색된 결과는 테이블에 데이터가 입력된 순서대로 출력
- 하지만, 데이터의 출력 순서를 특정 컬럼을 기준으로 오름차순 또는 내림차순으로 정렬하는 경우가 자주 발생
- 여러 개의 칼럼에 대해 정렬 순서를 정하는 경우도 발생
- 예
 - 사원 테이블에서 사번순이나 급여순으로 정렬하는 경우

ORDER BY 절

❖ 기본적인 정렬 방법

- 문자 값은 알파벳순으로 출력되고, 한글은 가나다순으로 출력된다
- 숫자 값은 가장 작은 값으로 먼저 출력된다
- 날짜 값은 과거의 날짜순으로 출력된다

❖ 사용법

```
SELECT          [DISTINCT]{* | column[alias]...}  
FROM            table  
[WHERE          condition]  
[ORDER BY      {column, expression}[ASC|DESC]]
```

- ORDER BY : 컬럼이나 표현식을 기준으로 출력 결과를 정렬할 때 사용
- ASC : 오름차순으로 정렬, 기본 값
- DESC : 내림차순으로 정렬하는 경우에 사용, 생략 불가능

집합 연산자

❖ 집합 연산자

- 테이블을 구성하는 행집합에 대해 테이블의 부분 집합을 결과로 반환하는 연산자
- 합병 가능 : 집합 연산의 대상이 되는 두 테이블의 칼럼수가 같고, 대응되는 칼럼끼리 데이터 타입이 동일

❖ 사용법

```
SELECT          명령문1  
[UNION | UNION ALL | INTERSECT | MINUS]  
SELECT          명령문2;
```

집합 연산자

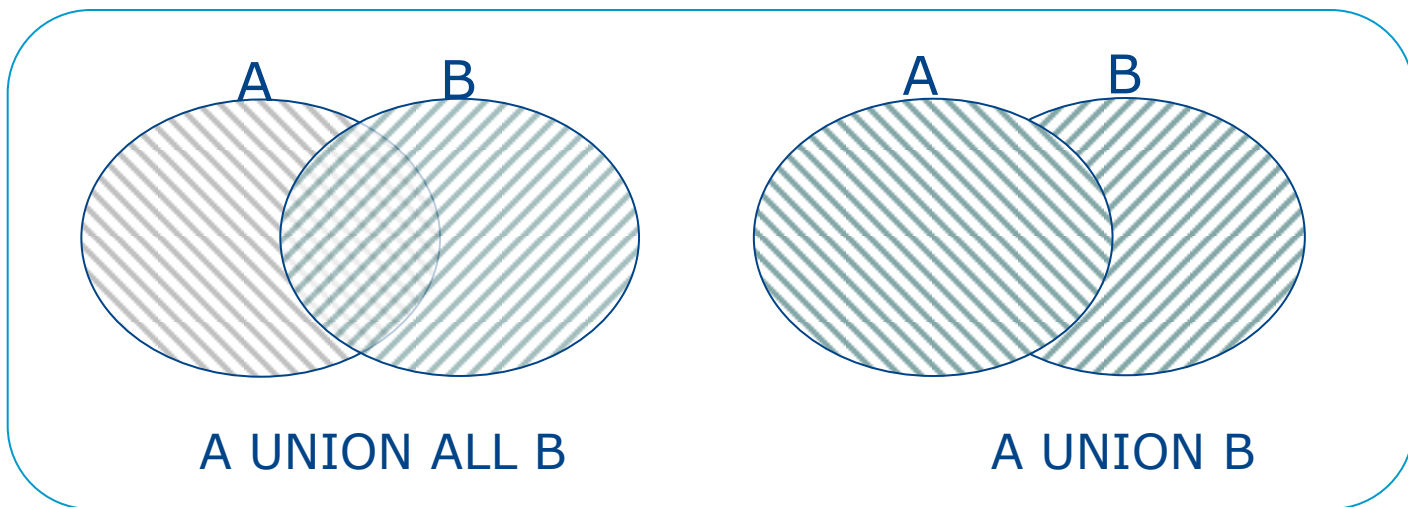
❖ 집합 연산자 종류

집합 연산자	의미
UNION	두 집합에 대해 중복되는 행을 제외한 합집합
UNION ALL	두 집합에 대해 중복되는 행을 포함한 합집합
MINUS	두 집합간의 차집합
INTERSECT	두 집합간의 교집합

UNION, UNION ALL 연산

❖ UNION, UNION ALL 연산

- UNION 연산은 수학의 집합 연산에서 합집합을 의미
- UNION 연산은 두 개의 테이블에 속하면서 중복되는 행을 제외한 행 집합을 결과 집합으로 반환
- UNION ALL 연산은 중복되는 행을 제외하지 않고 모두 출력



집합 연산을 위한 테이블 생성

❖ 사용 예

- 30번 부서이면서 급여가 500 이상인 사원의 집합(emp_sal)과 30번부서이면서 직급이 과장인 사원의 집합(emp_job)으로 구성된 두 개의 테이블 생성

```
SQL> create table emp_sal
  2  as select *
  3  from emp
  4  where deptno = 30 and sal >= 500;
```

테이블이 생성되었습니다.

```
SQL> create table emp_job
  2  as select *
  3  from emp
  4  where deptno = 30 and job = '과장';
```

테이블이 생성되었습니다.

emp_sal

오지호, 감우성

emp_job

오지호, 신동협

합병 불가능한 경우

- ❖ UNION 연산을 실행하는 두 테이블이 합병 불가능한 경우,
 - 두 질의의 칼럼수가 다르므로 오류 발생

```
SQL> select empno, ename
2   from emp_sal
3   union
4   select empno, ename, job
5   from emp_job;
select empno, ename
*
```

1행에 오류:
ORA-01789: query block has incorrect number of result columns

UNION, UNION ALL 연산의 비교

❖ 사용 예

- emp_sal과 emp_job테이블에 대해 UNION, UNION ALL 연산을 각각 수행한 결과를 출력하여라

```
SQL> select ename, job, sal
2   from emp_sal
3   union
4   select ename, job, sal
5   from emp_job;
```

ENAME	JOB	SAL
-----	-----	-----
감우성	차장	500
신동협	과장	450
오지호	과장	500

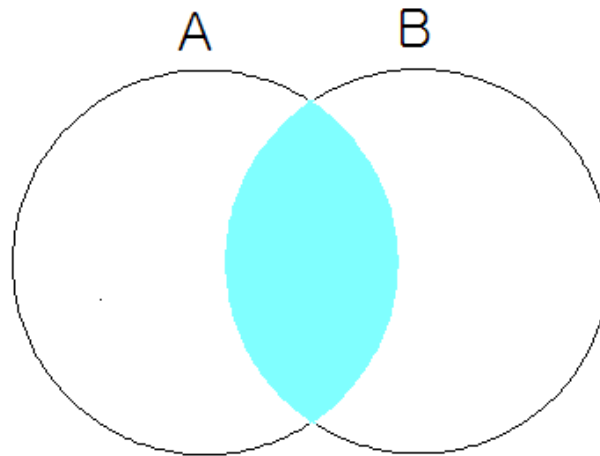
```
SQL> select ename, job, sal
2   from emp_sal
3   union all
4   select ename, job, sal
5   from emp_job;
```

ENAME	JOB	SAL
-----	-----	-----
오지호	과장	500
감우성	차장	500
오지호	과장	500
신동협	과장	450

INTERSECT 연산

❖ INTERSECT 연산

- 수학의 집합 연산에서 교집합을 의미
- INTERSECT 연산은 두 개의 테이블에 모두 속하는 행 집합을 결과 집합으로 반환



A INTERSECT B

INTERSECT 연산

❖ 사용 예

```
SQL> select ename, job, sal
2  from emp_sal
3  intersect
4  select ename, job, sal
5  from emp_job;
```

ENAME	JOB	SAL
오지호	과장	500

emp_sal

오지호, 감우성

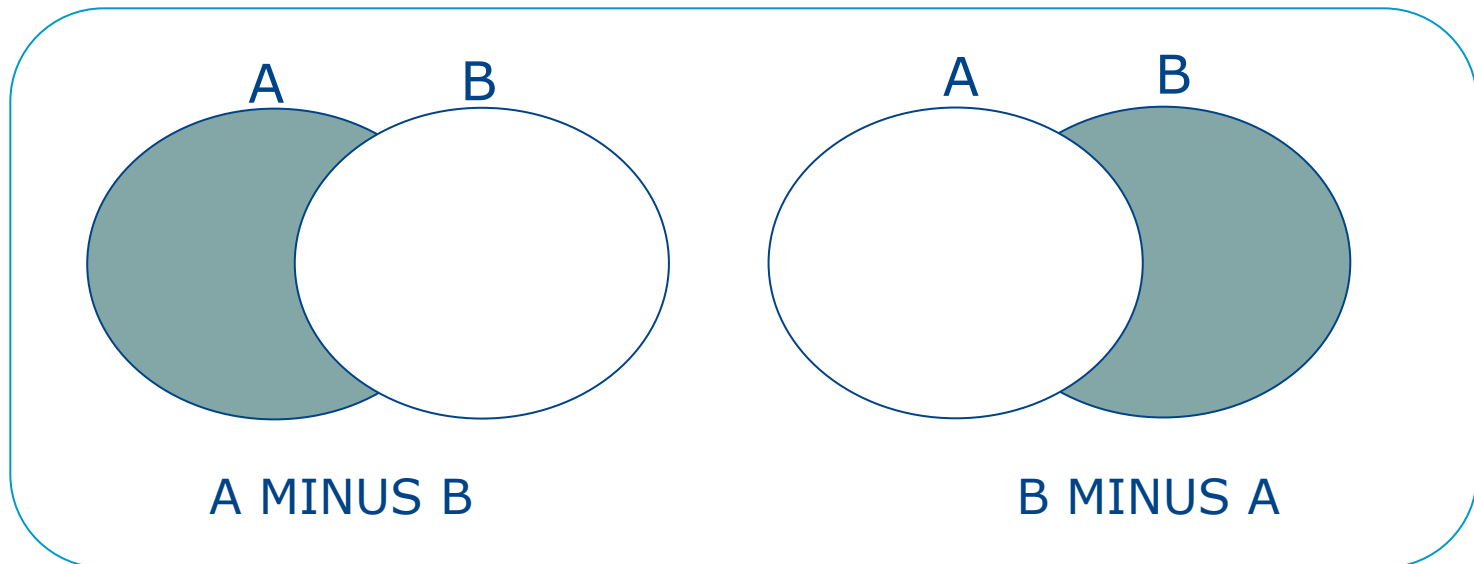
emp_job

오지호, 신동협

MINUS 연산

❖ MINUS 연산

- 수학의 집합 연산에서 차집합을 의미
- MINUS 연산은 테이블 A, B에서 A테이블에 속하지만 B 테이블에는 속하지 않는 행 집합을 결과 집합으로 반환



MINUS 연산의 예

```
SQL> select ename, job, sal
2  from emp_sal
3  minus
4  select ename, job, sal
5  from emp_job;
```

$emp_sal - emp_job$

ENAME	JOB	SAL
-----	-----	-----
감우성	차장	500

```
SQL> select ename, job, sal
2  from emp_job
3  minus
4  select ename, job, sal
5  from emp_sal;
```

$emp_job - emp_sal$

ENAME	JOB	SAL
-----	-----	-----
신동협	과장	450