

# 20220803 Database

🕒 작성일시	@2022년 8월 3일 오후 2:35
▼ 강의 번호	
📅 유형	
🔗 자료	
☑ 복습	<input type="checkbox"/>
☰ 속성	

## Chapter 5 다중 테이블 질의 (쿼리)

하나의 테이블에만 쿼리를 질의를 던지는 것이 적지 않아. 두개 세개 더 많은 테이블에 질의를 던질 수 있어야 한다.

1  
2

DESC employee;

Filter ...

- > 열
- > SQL 함수
- > SQL 키워드
- > 스니펫
- > 쿼리 내역
- > 쿼리 프
- > 매개 변

COLUMNS (9r × 6c)

Field	Type	Null	Key	Default	Extra
emp_id	smallint(5) unsigned	NO	PRI	(NULL)	auto_increment
fname	varchar(20)	NO		(NULL)	
lname	varchar(20)	NO		(NULL)	
start_date	date	NO		(NULL)	
end_date	date	YES		(NULL)	
superior_emp_id	smallint(5) unsigned	YES	MUL	(NULL)	
dept_id	smallint(5) unsigned	YES	MUL	(NULL)	
title	varchar(20)	YES		(NULL)	
assigned_branch_id	smallint(5) unsigned	YES	MUL	(NULL)	

1  
2

DESC department;

Fi

- >
- >
- >
- >
- >
- >
- >

COLUMNS (2r × 6c)

Field	Type	Null	Key	Default	Extra
dept_id	smallint(5) unsigned	NO	PRI	(NULL)	auto_increment
name	varchar(20)	NO		(NULL)	

---

### 5.1.1 데카르트 곱

```
mysql> SELECT e.fname, e.lname, d.name  
-> FROM employee e JOIN department d;
```

fname	lname	name
Michael	Smith	Operations
Michael	Smith	Loans
Michael	Smith	Administration
Susan	Barker	Operations
Susan	Barker	Loans
Susan	Barker	Administration
Robert	Tyler	Operations

이름을 줄여서 쓴다. JOIN

```

1 SELECT e.fname, e.lname, d.name
2 FROM employee e JOIN department d;

```

결과 #1 (54r x 3c)

fname	lname	name
Michael	Smith	Operations
Michael	Smith	Loans
Michael	Smith	Administration
Susan	Barker	Operations
Susan	Barker	Loans
Susan	Barker	Administration
Robert	Tyler	Operations
Robert	Tyler	Loans
Robert	Tyler	Administration
Susan	Hawthorne	Operations
Susan	Hawthorne	Loans

## 5.1.2 내부 조인

변경하기 위해서 테이블을 따로 만들어두는 것이 편하다.

```

mysql> SELECT e.fname, e.lname, d.name
-> FROM employee e JOIN department d
-> ON e.dept_id = d.dept_id;

```

```
SELECT e.fname, e.lname, d.name
FROM employee e JOIN department d
ON e.dept_id = d.dept_id;
```

## 5.2.2 테이블 재사용(같은 테이블을 두번 사용)

```
mysql> SELECT a.account_id, e.emp_id,
->    b_a.name open_branch, b_e.name emp_branch
-> FROM account a INNER JOIN branch b_a
->    ON a.open_branch_id = b_a.branch_id
->    INNER JOIN employee e
->    ON a.open_emp_id = e.emp_id
->    INNER JOIN branch b_e
->    ON e.assigned_branch_id = b_e.branch_id
-> WHERE a.product_cd = 'CHK';
```

account_id	emp_id	open_branch	emp_branch
10	1	Headquarters	Headquarters
14	1	Headquarters	Headquarters
21	1	Headquarters	Headquarters
1	10	Woburn Branch	Woburn Branch
4	10	Woburn Branch	Woburn Branch

```
SELECT a.account_id, e.emp_id,
b_a.name open_branch, b_e.name emp_branch
FROM account a INNER JOIN branch b_a
ON a.open_branch_id = b_a.branch_id
INNER JOIN employee e
ON a.open_emp_id = e.emp_id
INNER JOIN branch b_e
ON e.assigned_branch_id = b_e.branch_id
WHERE a.product_cd = 'CHK';
```

쉬운 예시

```
SELECT e.fname, a.lname
FROM employee AS e, employee AS a;
```

## Quiz 16

사원 테이블의 이름을 출력하는데 첫 번째 컬럼은 이름을 대문자로 출력하고 두 번째 컬럼은 이름을 소문자로 출력하고 세 번째 컬럼은 이름의 첫 번째 철자는 대문자로 하고 나머지는 소문자로 출력해 보겠습니다.

UPPER(ENAME)	LOWER(ENAME)	INITCAP(ENAME)
KING	king	King
BLAKE	blake	Blake
:	:	:
ADAMS	adams	Adams
MILLER	miller	Miller

```

1 SELECT UPPER(ename), LOWER(ename), CONCAT(SUBSTR(UPPER(ename), 1,1), SUBSTR(LOWER(ename), 2))
2 FROM emp;

```

emp (14r x 3c)

UPPER(ename)	LOWER(ename)	CONCAT(SUBSTR(UPPER(ename), 1,1), SUBSTR(LOWER(ename), 2))
KING	king	King
BLAKE	blake	Blake
CLARK	clark	Clark
JONES	jones	Jones
MARTIN	martin	Martin
ALLEN	allen	Allen
TURNER	turner	Turner
JAMES	james	James
WARD	ward	Ward
FORD	ford	Ford
SMITH	smith	Smith
SCOTT	scott	Scott
ADAMS	adams	Adams
MILLER	miller	Miller

```

SELECT UPPER(ename), LOWER(ename), CONCAT(SUBSTR(UPPER(ename), 1,1), SUBSTR(LOWER(ename), 2))
FROM emp;

```

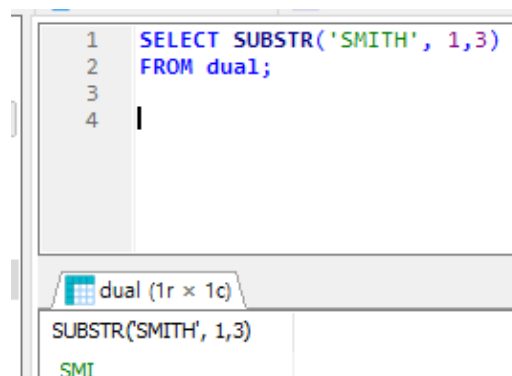
- INITCAP(ename)은 오라클에서 쓰여서 하이드에서 안 됨.

SQL에서의 함수는 크게 2가지로 나눌 수 있다.

1. 단일행 함수 - 하나의 행에만 적용되는 함수
2. 다중행 함수 - 여러 행의 값을 입력받아 하나의 행으로 출력하는 함수

## Quiz 17

영어 단어 SMITH에서 SMI만 잘라내서 출력해 보겠습니다.



```
SELECT SUBSTR('SMITH', 1,3)
FROM dual;
```

dual 테이블은 더미 테이블로 특정 테이블에 종속되지 않는 내용을 확인할 때 사용되는 일종의 가상의 연습용 테이블이다.

SQL은 다른 대부분의 컴퓨터 언어와는 달리 1에서 시작한다.

## Quiz 18

이름을 출력하고 그 옆에 이름의 철자 개수를 출력해 보겠습니다.

ENAME	LENGTH(ENAME)
KING	4
BLAKE	5
:	:
ADAMS	5
MILLER	6

1	SELECT	ename,	LENGTH(ename)
2	FROM	emp;	

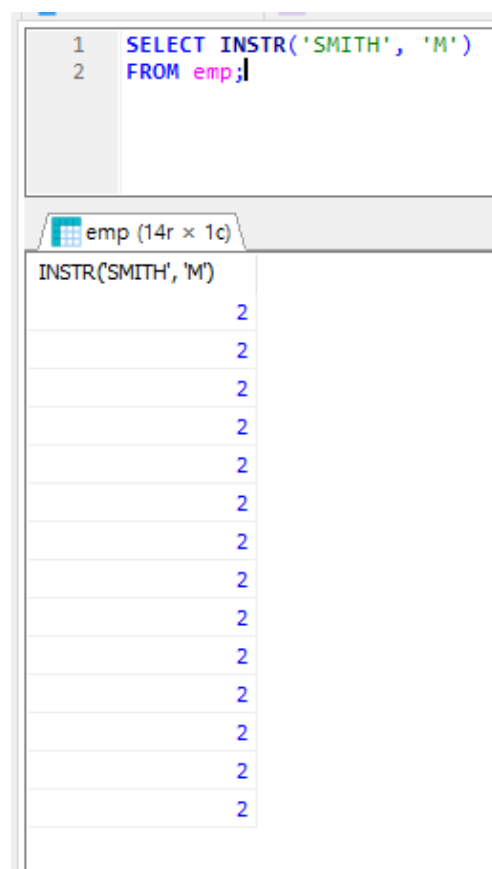
emp (14r × 2c)	
ename	LENGTH(ename)
KING	4
BLAKE	5
CLARK	5
JONES	5
MARTIN	6
ALLEN	5
TURNER	6
JAMES	5
WARD	4
FORD	4
SMITH	5
SCOTT	5
ADAMS	5
MILLER	6

```
SELECT ename, LENGTH(ename)
FROM emp;
```



## Quiz 19

사원 이름 SMITH에서 알파벳 철자 M이 몇 번째 자리에 있는지 출력해 보겠습니다.



```
1 SELECT INSTR('SMITH', 'M')
2 FROM emp;
```

emp (14r x 1c)
INSTR('SMITH', 'M')
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2

```
SELECT INSTR('SMITH', 'M')
FROM emp;
```

abcdefgh@naver.com 이메일에서 naver.com만 추출하고 싶다!

```

1 SELECT substr('abcdefgh@naver.com', 10,18)
2 FROM dual;

```

dual (1r x 1c)

substr('abcdefgh@naver.com', 10,18)
naver.com

```

1 SELECT SUBSTR('abcdefgh@naver.com', INSTR('abcdefgh@naver.com', '@') + 1 )
2 FROM dual;

```

dual (1r x 1c)

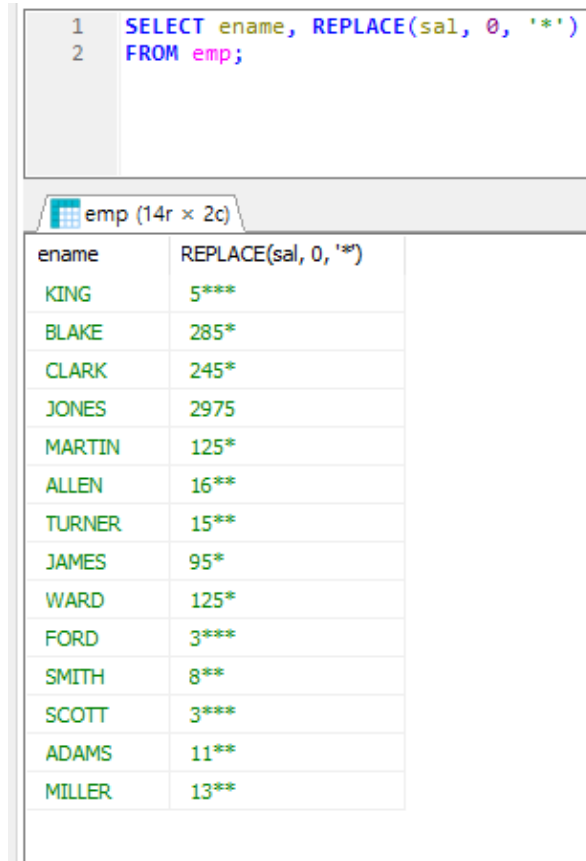
SUBSTR('abcdefgh@naver.com', INSTR('abcdefgh@naver.com', '@') + 1 )
naver.com

## Quiz 20

이름과 월급을 출력하는데, 월급을 출력할 때 숫자 0을 \*(별표)로 출력해 보겠습니다.

KING	5***
BLAKE	285*
:	:
ADAMS	11**
MILLER	13**

```
SELECT ename, REPLACE(sal, 0, '*')  
FROM emp;
```



ename	REPLACE(sal, 0, '*')
KING	5***
BLAKE	285*
CLARK	245*
JONES	2975
MARTIN	125*
ALLEN	16**
TURNER	15**
JAMES	95*
WARD	125*
FORD	3***
SMITH	8**
SCOTT	3***
ADAMS	11**
MILLER	13**

## Quiz 21

이름과 월급을 출력하는데 월급 컬럼의 자릿수를 10자리로 하고, 월급을 출력하고 남은 나머지 자리에 별표(\*)를 채워서 출력해 보겠습니다.

ENAME	SALARY1	SALARY2
KING	*****5000	5000*****
BLAKE	*****2850	2850*****
CLARK	*****2450	2450*****
:	:	:
SCOTT	*****3000	3000*****
ADAMS	*****1100	1100*****
MILLER	*****1300	1300*****

```
SELECT ENAME, LPAD(sal, 10, '*') AS salary1, RPAD(sal, 10, '*') AS salary2
FROM emp;
```

```
1 SELECT ENAME, LPAD(sal, 10, '*') AS salary1, RPAD(sal, 10, '*') AS salary2
2 FROM emp;
```

emp (14r x 3c)

ENAME	salary1	salary2
KING	*****5000	5000*****
BLAKE	*****2850	2850*****
CLARK	*****2450	2450*****
JONES	*****2975	2975*****
MARTIN	*****1250	1250*****
ALLEN	*****1600	1600*****
TURNER	*****1500	1500*****
JAMES	*****950	950*****
WARD	*****1250	1250*****
FORD	*****3000	3000*****
SMITH	*****800	800*****
SCOTT	*****3000	3000*****
ADAMS	*****1100	1100*****
MILLER	*****1300	1300*****

## Quiz 22

첫 번째 컬럼은 영어 단어 smith 철자를 출력하고, 두 번째 컬럼은 영어 단어 smith에서 s를 잘라서 출력하고, 세 번째 컬럼은 영어 단어 smith에서 h를 잘라서 출력하고, 네 번째 컬럼은 영어 단어 smiths의 양쪽에 s를 잘라서 출력해 보겠습니다.

smith	mith	smit	mith
-------	------	------	------

```
SELECT 'smith', LTRIM('smith','s'), RTRIM('smith','h'), TRIM('s' from
      'smiths')
FROM dual;
```

“austin “ ← 공백 없애기

TRIM(삭제옵션(선택), 삭제할문자(선택), 원본문자열 데이터 (필수))	특정 문자를 지움	SELECT trim( ENAME), trim('A' FROM ENAME), trim(LEADING 'A' FROM ENAME), trim(TRAILING 'S' FROM ENAME) FROM EMP;	O
LTRIM(원본문자열, 삭제할문자(선택)) RTRIM(원본문자열, 삭제할문자(선택))	왼쪽, 오른쪽 지정문자를 삭제	SELECT Ltrim( ENAME), Rtrim( ENAME) FROM EMP;	선택문자열 없음 공백만 삭제

1	SELECT LTRIM(' smith '), RTRIM(' smith '),
2	TRIM(' ' FROM ' smith ')
3	FROM DUAL;

smith (1r × 3c)		
LTRIM(' smith ')	RTRIM(' smith ')	TRIM(' ' FROM ' smith ')
smith	smith	smith

```
SELECT LTRIM(' smith '), RTRIM(' smith '),
      TRIM(' ' FROM ' smith ')
FROM DUAL;
```

함수명	설명	예제	MYSQL 확인
UPPER(문자열)	대문자로 변환하여 반환	select upper(ENAME), lower(ENAME), INITCAP(ENAME) FROM EMP;	O
LOWER(문자열)	소문자로 변환하여 반환		O
INITCAP(문자열)	괄호안 문자데이터중 첫글자는 대문자로, 나머지는 소문자로 변환후 반환		함수 없음!
SUBSTR(문자열데이터, 시작위치, 추출길이(선택))	문자열 데이터를 시작 위치부터 추출길이만큼 추출합니다. 추출 길이 없을경우 마지막까지 추출	select JOB, SUBSTR(JOB,2) , SUBSTR(JOB,1,2) FROM EMP;	O
INSTR(문자열, 위치를 찾으려는 부분문자, 시작위치(선택), 몇번째인지지정(선택))	문자열이 어디에 포함되어있는지 확인시 사용	SELECT INSTR('HELLO WORLD','L'), INSTR('HELLO WORLD','L',5), INSTR('HELLO WORLD','L',2,2) FROM DUAL;	SELECT INSTR('FOOBAR','BAR');
REPLACE(문자열, 찾는 문자열, 대체할문자(선택))	다른문자로 바꾸는 함수!	select JOB , replace(JOB, 'A' , 'Q') FROM EMP;	대체할 문자 생략 불가.
LPAD(문자열,자리수,채울문자(선택)) RPAD(문자열,자리수,채울문자(선택))	데이터의 빈 공간을 설정한 자리수 만큼 설정한 문자로 채운다.	SELECT lpad(ENAME,10,'@'), RPAD(ENAME,10,'@') FROM EMP;	채울문자 필수입력
CONCAT(문자열,문자열)	두 문자열 데이터를 합친다.	SELECT concat(ENAME, JOB) FROM EMP;	O
TRIM(삭제옵션(선택), 삭제할문자(선택), 원본문자열 데이터 (필수))	특정 문자를 지움	SELECT trim( ENAME), trim('A' FROM ENAME), trim(LEADING 'A' FROM ENAME), trim(TRAILING 'S' FROM ENAME) FROM EMP;	O

LTRIM(원본문자열, 삭제할문자(선택)) RTRIM(원본문자열, 삭제할문자(선택))	왼쪽, 오른쪽 지정문자를 삭제	SELECT Ltrim( ENAME), Rtrim( ENAME) FROM EMP;	선택문자열 없음 공백만 삭제
ROUND(숫자, 반올림위치(선택))	특정위치에서반올림	select ROUND(SAL, 1), ROUND(SAL) FROM EMP;	O
TRUNC(숫자, 버림위치(선택))	특정위치에서 버림		TRUNCATE(숫자, 버림위치(필수))
CEIL(숫자)	지정한 숫자와 가까운 큰 정수를 찾음	SELECT CEIL(3.141592), FLOOR(3.141592);	O
FLOOR(숫자)	지정한 숫자와 가까운 작은 정수		
MOD(나눗셈될숫자, 나눌숫자)	숫자를 나눈 나머지 값을 구함	SELECT MOD(3.141592,3);	O
날짜관련			
SYSDATE	날짜 데이터	SELECT SYSDATE AS NOW, SYSDATE+1 AS TOMORROW,SYSDATE -1 AS YESTERDAY FROM DUAL;	SELECT SYSDATE(), NOW(), NOW()+1, NOW()-1;
ADD_MONTHS((날짜데이터), 더할 개월수)	몇개월 이후의 날짜를 구함	SELECT sysdate, ADD_MONTHS(SYSDATE,3) FROM DUAL;	없음 DATE_ADD사용 DATEADD(NOW(), INTERVAL 1 MONTH)
*mysql 함수 DATE_ADD(date, INTERVAL expr unit)	시간 더하기	select date_add(now(), interval 1 day), date_add(now(), interval 1 hour), date_add('20211018', interval 1 month);	*MYSQL
*mysql 함수 DATE_SUB(date, INTERVAL expr unit)	시간빼기	select date_sub(now(), interval 1 day), date_sub(now(), interval 1 hour), date_sub('20211018', interval 1 month);	*MYSQL

MONTHS_BETWEEN(날짜 데이터1, 날짜 데이터 2)	두 날짜 간의 개월수 차이를 구함.	select months_between(hiredate,sysdate) from emp;	없음
*mysql 함수 DATEDIFF(날짜데이터 1, 날짜데이터2)	두 날짜 사이의 차이를 구함	*MYSQL SELECT datediff(NOW(), '20220620')	
NEXT_DAY(날짜데이터, 요일문자)	다음날요일,달 마지막 날짜를 구함	SELECT SYSDATE, NEXT_DAY(SYSDATE, '월요일'), LAST_DAY(SYSDATE) FROM DUAL	없음
LAST_DAY(날짜데이터)	속한 달의 마지막 날짜를 출력		
ROUND(날짜, 포맷)	날짜 반올림	SELECT round(NOW(), 'CC'), round(NOW(), 'YYYY'), round(NOW(), 'Q'), round(NOW(), 'DDD'), round(NOW(), 'HH'), TRUNC(NOW(), 'CC'), TRUNC(NOW(), 'YYYY'), TRUNC(NOW(), 'Q'), TRUNC(NOW(), 'DDD'), TRUNC(NOW(), 'HH');	SELECT round(NOW(), 'CC'), round(NOW(), 'YYYY'), round(NOW(), 'Q'), round(NOW(), 'DDD'), round(NOW(), 'HH'), TRUNCATE(NOW(), 'CC'), TRUNCATE(NOW(), 'YYYY'), TRUNCATE(NOW(), 'Q'), TRUNCATE(NOW(), 'DDD'), TRUNCATE(NOW(), 'HH');
TURUNC(날짜, 포맷)	날짜 버림		
TO_CHAR		SELECT TO_CHAR(SYSDATE(), 'YYYY/MM/DD HH24:MI:SS')	없음
*MY SQL DATE_FORMAT(날짜, 포맷)	날짜, 숫자데이터를 문자 데이터로 변환	*MYSQL SELECT DATE_FORMAT(SYSDATE(), '%Y-%M-%D');	
TO_NUMBER		SELECT TO_NUMBER('1') FROM DUAL;	CAST 함수써야함

TO_NUMBER		SELECT TO_NUMBER('1') FROM DUAL;	CAST 함수써야함
*MYSQL CAST(문자열 AS INTEGER)	문자데이터를 숫자데이터로 변환	*MYSQL SELECT CAST('1' AS INTEGER) AS NUM;	
TO_DATE	문자데이터를 날짜 데이터로 변환	TO_DATE	X DATE_FORMAT 함수 쓸것
NVL()	열 또는 데이터를 입력하여 해당데이터가 NULL이 아닐 경우데이터를 그대로 반환하고, NULL인 경우지정한 데이터를 반환.	SELECT NVL(COMM,0) FROM EMP;	X IFNULL 함수 쓸것
NVL2()	데이터가 NULL이 아닐때 반환데이터를 추가로 지정가능	SELECT NVL2(COMM, 'O', 'X') FROM EMP	X
*MYSQL IFNULL(비교군, 리턴내용)	NULL 일경우 사용자 지정내용 리턴	*MYSQL SELECT IFNULL(NAME, '값이없습니다') FROM TEST	

DECODE	<p>DECODE([검사대상이 될 열 또는 데이터, 연산이나 함수의 결과],[조건1],[데이터가 조건1과 일치할때 반환결과], [조건2],[데이터가 조건2와 일치할때 반환할 결과],</p> <p>.....</p> <p>[조건N],[데이터가 조건N와 일치할때 반환할 결과],</p> <p>],</p> <p>[위조건과 일치한 경우가 없을때 반환할 결과])</p> <p>IF나 /SWITCH-CASE 조건문과 같은 이 데이터를 조건에부합하는지 비교하고 부합하는 경우 설정한 결과를 반환한다.</p>	<pre>SELECT EMPNO, ENAME, JOB, SAL, DECODE( JOB ,'MANAGER', SAL*1.1, 'SALESMAN', SAL*1.1, 'ANALYST', SAL ,SAL* 1.03) FROM EMP;</pre>	<p>없음 IF나 CASE문 쓸것</p> <p>ENCODE(문자열,키)</p> <p>암호화 함수</p> <p>DECODE(문자열, 키)</p> <p>복호화 함수</p> <p>사용 용도가 다름</p>
--------	---	--	--

CASE	<p>조건을 비교하되 각 조건에 사용하는 데이터가 서로 상관없어도 비교가능, 기존데이터 값이 같은 데이터외에 다양한 조건을 사용할 수있다.</p> <p>CASE [비교할 데이터]          WHEN [조건1] THEN [조건1 True 반환할 결과]          WHEN [조건2] THEN [조건2 True 반환할 결과]          WHEN [조건3] THEN [조건3 True 반환할 결과]          ...          WHEN [조건n] THEN [조건n True 반환할 결과]          ELSE [위조건에 일치하는경우가 없을때 바나할 결과]          END</p>	<p>SELECT EMPNO,          CASE JOB          WHEN 'MANAGER' THEN          SAL*1.1          WHEN 'SALESMAN' THEN          SAL*1.05          ELSE SAL*1.03          END          FROM EMP;</p>	O
* MYSQL IF문	<p>오라클 함수의 decode함수는 조건과 같을 겨우만 비교하지만, 조건문을 적어 줄수있다.</p> <p>IF (조건, 조건이 일치할경우 VALUE, 조건이 일치하지 않을 경우 VALUE)</p>	<p>*MY SQL          SELECT deptno,          IF(deptno &lt;= 20,'RESEARCH','OPERATIONS') deptno as "          Dept Name"          FROM dept;</p>	
다중행함수			

SUM(해당열)	합계	<pre>SELECT COUNT(SAL), COUNT(ALL SAL),COUNT(distinct SAL ), SUM(SAL), MAX(SAL), MIN(SAL), AVG(SAL) FROM EMP;</pre>
COUNT(해당열)	데이터의 갯수를 구해줌	
MAX(),MIN()	최대값, 최소값	
AVG()	평균값	

가장 오른쪽이 하이드가 쓰는 언어이고 맨 왼쪽이 오라클 언어.



- 반올림 ROUND

```

1  /* 반올림 ROUND
2
3  ROUND(대상, 반올림 위치)
4  ..... 반올림 위치가 소수점 아래로 양수로 그 자리까지로 반올림
5  ..... 반올림 위치가 소수점 위에서는 (즉 정수자리에서는) 오히려 음수로
6  ..... 그 자리에서 반올림
7  */
8  /* 5까지는 올리고 4까지는 버리는 게 반올림 ROUND */
9  SELECT '1234.5678', ROUND(1234.5678, 1), ROUND(1234.5678, -1)
10 FROM DUAL;

```

DUAL (1r × 3c)		
1234.5678	ROUND(1234.5678, 1)	ROUND(1234.5678, -1)
1234.5678	1,234.6	1,230

876.567 숫자를 출력하는데 소수점 두 번째 자리인 6에서 반올림해서 출력해 보겠습니다.

```

SELECT ROUND(876.567, 1)
FROM DUAL;

```

```

1  SELECT ROUND(876.567, 1)
2  FROM DUAL;

```

DUAL (1r × 1c)	
ROUND(876.567, 1)	876.6

- Truncate 버리기

3	*/
4	
5	SELECT '1234.5678' AS 숫자 , Truncate(1234.5678,-2)
6	FROM DUAL;

DUAL (1r × 2c)	
숫자	Truncate(1234.5678,-2)
1234.5678	1,200

/\* 버림 TRUNCATE

TRUNCATE(대상, 버림 위치)

버림 위치가 소수점 아래에서는 양수로 그 자리까지 남기고 버림  
 버림 위치가 소수점 위에는 (즉 정수 자리에서는) 음수로 그 자리에서 버림  
 \*/

- 나머지 연산자

예) 5/ 2 나머지 1

1	/* 나머지 함수 mod()
2	SELECT MOD(대상 숫자, 나눌 숫자)
3	*/
4	
5	SELECT MOD(5, 2)
6	FROM DUAL;
7	
8	

DUAL (1r × 1c)	
MOD(5, 2)	
	1

사원 번호와 사원 번호가 홀수이면 1, 짝수이면 0을 출력

1	/* 나머지 함수 mod()
2	SELECT MOD(대상 숫자, 나눌 숫자)
3	*/
4	
5	SELECT empno, (MOD(empno, 2))
6	FROM emp;
7	
8	

emp (14r × 2c)	
empno	(MOD(empno, 2))
7,839	1
7,698	0
7,782	0
7,566	0
7,654	0
7,499	1
7,844	0
7,900	0
7,521	1
7,902	0
7,369	1
7,788	0
7,876	0
7,934	0

```
SELECT empno, (MOD(empno, 2))
FROM emp;
```

사원 번호가 짝수인 사원들의 사원 번호와 이름을 출력하는 쿼리

EMPNO	ENAME
7698	BLAKE
7782	CLARK
7566	JONES
7654	MARTIN
:	:

```

SELECT EMPNO, ENAME
FROM emp
WHERE MOD(ename, 2) = 0;

```

1	SELECT EMPNO, ENAME
2	FROM emp
3	WHERE MOD(ename, 2) = 0;

emp (14r × 2c)	
EMPNO	ENAME
7,839	KING
7,698	BLAKE
7,782	CLARK
7,566	JONES
7,654	MARTIN
7,499	ALLEN
7,844	TURNER
7,900	JAMES
7,521	WARD
7,902	FORD
7,369	SMITH
7,788	SCOTT
7,876	ADAMS
7,934	MILLER

다음의 쿼리는 10을 3으로 나눈 몫을 출력하는 쿼리입니다.

FLOOR(10/3)
3

```
SELECT FLOOR(10/3)
FROM emp;
```

```
1 SELECT FLOOR(10/3)
2 FROM emp;
3
```

emp (14r x 1c)

FLOOR(10/3)

3

3

3

3

3

3

3

3

3

3

3

3

3

3

/\* CASE

```
CASE
  WHEN sal >= 3000 Then 500
  WHEN sal >= 2000 Then 300
  WHEN sal >= 1000 Then 200
  ELSE 0
END AS Bonus
```

• /

```
SELECT ename, job, sal,
CASE
  WHEN sal >= 3000 Then 500
  WHEN sal >= 2000 Then 300
  WHEN sal >= 1000 Then 200
  ELSE 0
END AS Bonus
FROM emp;
```

```
1  SELECT ename, job, sal,
2  CASE
3  WHEN sal >= 3000 Then 500
4  WHEN sal >= 2000 Then 300
5  WHEN sal >= 1000 Then 200
6  ELSE 0
7  END AS Bonus
8  FROM emp;
```

emp (14r x 4c)

ename	job	sal	Bonus
KING	PRESIDENT	5,000	500
BLAKE	MANAGER	2,850	300
CLARK	MANAGER	2,450	300
JONES	MANAGER	2,975	300
MARTIN	SALESMAN	1,250	200
ALLEN	SALESMAN	1,600	200
TURNER	SALESMAN	1,500	200
JAMES	CLERK	950	0
WARD	SALESMAN	1,250	200
FORD	ANALYST	3,000	500
SMITH	CLERK	800	0
SCOTT	ANALYST	3,000	500
ADAMS	CLERK	1,100	200
MILLER	CLERK	1,300	200

응용

1	SELECT	ename, job, sal,
2	CASE	
3	WHEN	sal >= 3000 Then sal + 500
4	WHEN	sal >= 2000 Then sal + 300
5	WHEN	sal >= 1000 Then sal + 200
6	ELSE	sal
7	END AS	Bonus
8	FROM	emp;

emp (14r × 4c)			
ename	job	sal	Bonus
KING	PRESIDENT	5,000	5,500
BLAKE	MANAGER	2,850	3,150
CLARK	MANAGER	2,450	2,750
JONES	MANAGER	2,975	3,275
MARTIN	SALESMAN	1,250	1,450
ALLEN	SALESMAN	1,600	1,800
TURNER	SALESMAN	1,500	1,700
JAMES	CLERK	950	950
WARD	SALESMAN	1,250	1,450
FORD	ANALYST	3,000	3,500
SMITH	CLERK	800	800
SCOTT	ANALYST	3,000	3,500
ADAMS	CLERK	1,100	1,300
MILLER	CLERK	1,300	1,500

다음의 쿼리는 보너스를 출력할 때 직업이 SALESMAN, ANALYST이면 500을 출력하고, 직업이 CLERK, MANAGER이면 400을 출력하고, 나머지 직업은 0을 출력하는 쿼리입니다.

ENAME	JOB	보너스
KING	PRESIDENT	0
BLAKE	MANAGER	400
CLARK	MANAGER	400
JONES	MANAGER	400
MARTIN	SALESMAN	500
ALLEN	SALESMAN	500
TURNER	SALESMAN	500
JAMES	CLERK	400
WARD	SALESMAN	500
FORD	ANALYST	500
SMITH	CLERK	400
SCOTT	ANALYST	500
ADAMS	CLERK	400
MILLER	CLERK	400

```

SELECT ename, job, CASE WHEN job in ('SALESMAN','ANALYST') THEN 500
                        WHEN job in ('CLERK','MANAGER') THEN 400
                        ELSE 0 END as 보너스
FROM emp;

```



1	SELECT	ENAME,	JOB,	CASE	WHEN	job	IN	(	'SALESMAN',	'ANALYST')	THEN	500	
2					WHEN	job	IN	(	'CLERK',	'MANAGER')	THEN	400	
3					ELSE	0	END	AS	보	너	스		
4	FROM	emp;											
5													

ENAME	JOB	보너스
KING	PRESIDENT	0
BLAKE	MANAGER	400
CLARK	MANAGER	400
JONES	MANAGER	400
MARTIN	SALESMAN	500
ALLEN	SALESMAN	500
TURNER	SALESMAN	500
JAMES	CLERK	400
WARD	SALESMAN	500
FORD	ANALYST	500
SMITH	CLERK	400
SCOTT	ANALYST	500
ADAMS	CLERK	400
MILLER	CLERK	400

```

SELECT ENAME, JOB, CASE WHEN job IN ('SALESMAN', 'ANALYST') THEN 500
WHEN job IN ('CLERK', 'MANAGER') THEN 400
ELSE 0 END AS 보너스
FROM emp;

```