

# DataBase

1. employees 테이블에서 모든 내용을 출력

```
select * from employees;
```

2. employees 테이블에서 employee *id* , first\_name , last\_name 만 출력

```
select employee_id, first_name, last_name from employees;
```

3. employees 테이블에서 employee\_id, first\_name, last\_name 만 출력하되 내림차순

```
select employee_id, first_name, last_name from employees order by employee_id desc;
```

4. employees테이블에서 job의 종류를 알고자하면

```
select distinct job_id from employees;
```

5. 사원번호, 전체이름, 그리고 각각의 이메일

```
select employee_id 사원번호, concat(last_name, ' ', first_name) 전체이름, concat(email, '@company') e_mail from employees;
```

6. employees테이블에서 employee\_id, salary,salary에 500을 더한값,100을 뺀값,10%추가해서 2로 나눈값을 출력

```
select employee_id 사원번호, salary 급여, salary+500 추가급여, salary-100 인하급여, (salary*1.1)/2 조정급여from employees;
```

7. employees 테이블에서 employee\_id 가 100인 사원 정보를 출력

```
select * from employees where employee_id=100;
```

8. employees 테이블에서 first\_name 이 david인직원정보를 출력

```
select * from employees where first_name='david';
```

9. employees 테이블에서 employee\_id가 105 이상인 직원정보출력

```
select * from employees where employee_id>=105;
```

10. employees 테이블에서 salary가 10,000이상이고 20,000이하인 직원정보출력

```
select * from employees where salary between 10000 and 20000;
```

11. employees 테이블에서 salary가 10000, 17000, 24000 인직원정보

```
select * from employees where salary in(10000,17000,24000);
```

12. employees 테이블에서 job\_id 의 값이 AD로 시작하는 모든 데이터 출력

와일드 카드 문자 : \* 여러글자를 대신 ?한글자를 대신

DB에서는 : %여러글자를대신 , - 한글자를대신

```
select * from employees where job_id like 'ad%';
```

13. employees 테이블에서 job\_id 의 값이 AD로 시작하면서 뒤에따라오는 문자열이 3자리인 데이터의 값

```
select * from employees where job_id like 'ad____';
```

14. employees 테이블에서 manager\_id가 null인 값인 직원정보

주의 null은 0과 공백문자와는 다름.

```
select * from employees where manager_id is null;
```

15. employees 테이블에서 salary가 4000을 초과하면서

job\_id가 IT\_PROG인 값을 조회

```
select * from employees where salary>4000 and job_id like 'IT_PROG';
```

16. employees 테이블에서 salary가 4000을 초과하면서

job\_id가 IT\_PROG이거나 account인 경우를 조회

```
select * from employees where salary>4000 and (job_id like 'IT_PROG' or job_id like '%ACC%');
```

17. employees 테이블에서 employee\_id가 105가 아닌직원을 조회

```
select * from employees where employee_id!=105;
```

===

1. employees 테이블에서 last\_name을 소문자와 대문자로 각각 출력하고, email의 첫번째 문자는 대문자로 출력

```
select last_name,  
lower(last_name) lower적용,  
upper(last_name) upper적용,  
email  
from employees;
```

2. employees 테이블에서 job\_id 데이터 값의 첫째 자리부터 시작해서 두개의 문자를 출력

```
select job_id,  
substr(job_id, 1, 2)  
from employees;
```

3. employees 테이블에서 job\_id 문자열 값이 ACCOUNT면 ACCNT로 출력

```
select job_id,  
replace(job_id, 'ACCOUNT', 'ACCNT')  
from employees;
```

4. employees 테이블에서 first\_name에 대해 12자리의 문자열 자리를 만들되 first\_name의 데이터 값이 12자리보다 작으면 왼쪽부터 \*를 채워서 출력

```
select first_name,  
lpad(first_name, 12, '*')  
from employees;
```

5. employees 테이블에서 job\_id의 데이터 값에 대해 왼쪽 방향으로 'F'문자를 만나면 삭제하고 또 오른쪽 방향으로 'T'문자를 만나면 삭제

```
SELECT  
  TRIM(LEADING 'F' FROM TRIM(TRAILING 'T' FROM job_id)) AS modified_job_id  
FROM employees;
```

6. employees 테이블에서 salary를 30일로 나눈 후 나눈값의 소수점 첫째자리, 소수점 둘째 자리, 정수 첫째 자리에서 반올림한 값을 출력  
\*round.5이상이면 반올림  
trunc는 무조건 절삭

```
select salary,  
salary/30 일급,  
round(salary/30, 0),  
round(salary/30, 1),  
round(salary/30, -1)  
from employees;
```

7. employees 테이블에서 first\_name, last\_name, department\_id, salary를 출력하되 department\_id가 60인 경우에는 급여를 10%인상한 값을 계산하여 출력하고 나머지 경우에는 원래의 값을 출력 그리고 department\_id가 60인 경우에는 10% 인상을 출력하고 미인상을 출력

```
select  
  first_name,  
  last_name,  
  salary 원래급여,  
  if(department_id= '60', salary*1.1, salary) 조정된급여,  
  if(department_id='60', '10%인상', '미인상') 인상여부  
from employees;
```

8. employees 테이블에서 job\_id가 IT\_PROG라면 employee\_id, first\_name, last\_name, salary를 출력하되 salary가 9000이상이면 상위급여 6000과 8999사이이면 중위급여 그외는 하위급여로 출력

```

select
employee_id, first_name, last_name, salary,
case
when salary >=9000 then '상위급여'
when SALARY between 6000 and 8999 then '중위급여'
else '하위급여'
end as 급여등급
from employees
where job_id = 'it_prog';

```

9. Rank,Dense\_rank, row\_number 함수를 각각 이용해 employees 테이블의 salary값이 높은 순서대로 순위를 매겨 출력

```

SELECT
employee_id,
salary,
RANK() OVER(ORDER BY salary DESC) rank_급여,
DENSE_RANK() OVER(ORDER BY salary DESC) dense_rank_급여,
ROW_NUMBER() OVER(ORDER BY salary DESC) row_number_급여
FROM employees
ORDER BY salary DESC;

```

10. employees 테이블에서 salary의 행 수가 몇개인지 세어서 출력

```

select count(salary) salary_행수
from employees;

```