

-

SELECT 조회컬럼 연산자 별칭 * distinct 함수 **그룹함수**
FROM 테이블명(subquery)
WHERE 컬럼명 연산자 값
GROUP BY 그룹함수컬럼명
HAVING 그룹함수명 연산자 값
ORDER BY 정렬기준컬럼명 인덱스 별칭 asc|desc

-함수

그룹	sum avg stdev varianc-숫자 count max min-숫자 문자 날짜 count(*) -> null값 포함 갯수
타입변환	cast (123 as date number varchar2) to_char to_date to_number to_char(sysdate, 'yyyy/mm/dd')
문자	upper lower initcap length instr substr--> 05년도 05월
숫자	round trunc mod
날짜	sysdate add_months months_between (? - ?) / 365
NULL	nvl(컬럼명, null대체값)
순위	rownum – subquery row_number() over (parttition by 순위정할그룹 order by 순위정할기준컬럼) rank dense_rank

268 – clob , blob – character | binary large object – 1tb

- 274p

join

예)

- 사번, 사원명, 부서코드 조회: employees 테이블
- 사번, 사원명, **부서명** 조회: employees + departments 테이블

hr

employees 테이블	departments 테이블
100 김사원 10	10 인사부
200 박대리 20	20 IT개발부
300 최과장 10	30 교육부
400 이부장 null	(+)

- inner join

```
select employee_id, first_name, employees.department_id, department_name
```

```
from employees, departments
```

```
where employees.department_id = departments.department_id;
```

```
select employee_id, first_name, e.department_id, department_name
```

```
from employees e, departments d
```

```
where e.department_id = d.department_id;
```

-outer join

각 사원의 사번 이름 부서명 조회되 부서 없는 사원도 포함 출력

```
select employee_id, first_name, e.department_id, department_name
```

```
from employees e, departments d
```

```
where e.department_id = d.department_id(+);
```

null 존재 조회해야 한다 = null x

-outer join

각 부서명, 부서직원명, 사번 조회하되 부서원 하나도 없는 부서명 같이 포함 출력

```
select employee_id, first_name, e.department_id, department_name
```

```
from employees e, departments d
```

```
where e.department_id(+) = d.department_id;
```

-

London 도시 근무하는 직원명, 부서명, 도시명 조회

(다국적 기업)

직원명 = employees

부서명 = departments

도시명 = locations

```
select first_name 직원명, department_name 부서명, city 도시명
```

```
from employees e, departments d, locations l
```

```
where e.department_id = d.department_id
```

```
and d.location_id = l.location_id
```

```
and UPPER(city)='LONDON';
```

- IT 관련 부서의 사원명, 부서명, 도시명 조회

(IT 관련 부서는 부서명에 IT 포함)

```
select first_name 사원명, department_name 부서명, city 도시명
from employees e, departments d, locations l
where e.department_id = d.department_id
and d.location_id = l.location_id
and department_name like '%IT%';
```

- 사원명 , 국가명 조회

EMPLOYEES – DEPARTMENT_ID

DEPARTMENTS – DEPARTMENT_ID, DEPARTMENT_NAME, ... , LOCATION_ID

LOCATIONS – LOCATION_ID, , CITY , COUNTRY_ID

COUNTRIES – COUNTRY_ID,..., COUNTRY_NAME

- EMPLOYEES 테이블 내사번, 내이름, 상사사번, 상사이름 조회

JOIN 테이블 자신 테이블일 때

self inner join

self outer join

EMPLOYEE_ID - 내사번

MANAGER_ID – 상사사번

SELECT me.employee_id 내사번, me.first_name 내이름, me.manager_id 상사사번,
man.employee_id 상사사번2, man.first_name 상사이름

FROM EMPLOYEES me, EMPLOYEES man

WHERE me.manager_id=man.employee_id(+);

상사사번null 존재 null존재 x

EMPLOYEES (내정보)				EMPLOYEES(상사정보)					
사번	이름	상사사번	급여	사번	이름	상사사번	급여
200	이자바		100		200	이자바		100	
300	김자바		200		300	김자바		200	
400	박자바		100		400	박자바		100	
100	최사장		NULL		100	최사장		NULL	

- 내상사보다 급여를 많이 받는 사원의 이름 급여 조회

SELECT me.first_name 내이름, me.salary 내급여, me.manager_id 상사사번,
man.salary 상사급여

FROM EMPLOYEES me, EMPLOYEES man

WHERE me.manager_id=man.employee_id

and me.salary > man.salary;

cross join- 나오면 안된다(where 절 필수)

inner join

outer join

self join

표준sql / 오라클 미리 독자적 sql

표준join – ansi	오라클 join
<p>-inner join</p> <pre>select employee_id, first_name, e.department_id, department_name from employees e inner join departments d on e.department_id = d.department_id;</pre>	<p>-inner join</p> <pre>select employee_id, first_name, e.department_id, department_name from employees e, departments d where e.department_id = d.department_id;</pre>
<p>-outer join</p> <pre>select employee_id, first_name, e.department_id, department_name from employees e left outer join departments d on e.department_id = d.department_id;</pre> <pre>select employee_id, first_name, e.department_id, department_name from employees e right outer join departments d on e.department_id = d.department_id;</pre>	<p>-outer join</p> <pre>select employee_id, first_name, e.department_id, department_name from employees e, departments d where e.department_id = d.department_id(+);</pre> <pre>select employee_id, first_name, e.department_id, department_name from employees e, departments d where e.department_id(+) = d.department_id;</pre>
<pre>SELECT me.first_name 내이름, me.salary 내급 여, me.manager_id 상사사번, man.salary 상사급여 FROM EMPLOYEES me inner join EMPLOYEES man on me.manager_id=man.employee_id</pre>	<p>-self join</p> <pre>SELECT me.first_name 내이름, me.salary 내급 여, me.manager_id 상사사번, man.salary 상사급여 FROM EMPLOYEES me, EMPLOYEES man WHERE me.manager_id=man.employee_id</pre>

--	--

```
SELECT me.first_name 내이름, me.salary 내급여,  
me.manager_id 상사사번, man.salary 상사급여  
FROM EMPLOYEES me, EMPLOYEES man  
WHERE me.manager_id=man.employee_id  
and me.salary > man.salary
```

- 테이블 설계

개념적 모델링, 논리적 모델링, 물리적 모델링

고객 방문 기록

고객 이름	출생년도	주소	연락처	구매한 물건	단가(천 원)	수량
이승기	1987	서울	011-111-1111			
김범수	1979	경남	011-222-2222	운동화	30	2
김범수	1979	경남	011-222-2222	노트북	1000	1
김경호	1971	전남	019-333-3333			
조용필	1950	경기	011-444-4444	모니터	200	1
바비킴	1973	서울	010-000-0000	모니터	200	5
윤종신	1969	경남	안 남김			
김범수	1979	경남	011-222-2222	청바지	50	3
임재범	1963	서울	016-666-6666			
바비킴	1973	서울	010-000-0000	메모리	80	10
성시경	1979	경남	안 남김	책	15	5
은지원	1978	경북	011-888-8888	책	15	2
임재범	1963	서울	016-666-6666			
은지원	1978	경북	011-888-8888	청바지	50	1
바비킴	1973	서울	010-000-0000	운동화	30	2
은지원	1978	경북	011-888-8888			
은지원	1978	경북	011-888-8888	책	15	1
바비킴	1973	서울	010-000-0000	운동화	30	2
조관우	1965	경기	018-999-9999			

[그림 4-3] 데이터베이스 모델링 단계 1

기록된 내용에서 물건을 구매한 적이 없는 고객을 위쪽으로 다시 정렬해 보자.

고객 방문 기록

고객 이름	출생년도	주소	연락처	구매한 물건	단가(천 원)	수량
이승기	1987	서울	011-111-1111			
김경호	1971	전남	019-333-3333			
윤종신	1969	경남	안남김			
임재범	1963	서울	016-666-6666			
임재범	1963	서울	016-666-6666			
은지원	1978	경북	011-888-8888			
조관우	1965	경기	018-999-9999			
김범수	1979	경남	011-222-2222	운동화	30	2
김범수	1979	경남	011-222-2222	노트북	1000	1
조용필	1950	경기	011-444-4444	모니터	200	1
바비킴	1973	서울	010-000-0000	모니터	200	5
김범수	1979	경남	011-222-2222	청바지	50	3
바비킴	1973	서울	010-000-0000	메모리	80	10
성시경	1979	경남	안남김	책	15	5
은지원	1978	경북	011-888-8888	책	15	2
은지원	1978	경북	011-888-8888	청바지	50	1
바비킴	1973	서울	010-000-0000	운동화	30	2
은지원	1978	경북	011-888-8888	책	15	1
바비킴	1973	서울	010-000-0000	운동화	30	2

[그림 4-4] 데이터베이스 모델링 단계 2

L자형 테이블을 빈칸이 있는 곳과 없는 곳으로 분리해 보자. 그러면, 다음과 같이 고객 방문 기록이 고객 테이블과 구매 테이블로 분리된다.

고객 테이블

고객 이름	출생년도	주소	연락처
이승기	1987	서울	011-111-1111
김경호	1971	전남	019-333-3333
윤종신	1969	경남	안 남김
임재범	1963	서울	016-666-6666
임재범	1963	서울	016-666-6666
은지원	1978	경북	011-888-8888
조관우	1965	경기	018-999-9999
김범수	1979	경남	011-222-2222
김범수	1979	경남	011-222-2222
조용필	1950	경기	011-444-4444
바비킴	1973	서울	010-000-0000
김범수	1979	경남	011-222-2222
바비킴	1973	서울	010-000-0000
성시경	1979	경남	안 남김
은지원	1978	경북	011-888-8888
은지원	1978	경북	011-888-8888
바비킴	1973	서울	010-000-0000
은지원	1978	경북	011-888-8888
바비킴	1973	서울	010-000-0000

구매 테이블

구매한 물건	단가(천 원)	수량
운동화	30	2
노트북	1000	1
모니터	200	1
모니터	200	5
청바지	50	3
메모리	80	10
책	15	5
책	15	2
청바지	50	1
운동화	30	2
책	15	1
운동화	30	2

[그림 4-5] 데이터베이스 모델링 단계 3

예: 사원테이블

사번	이름	급여	입사일	상사사번	부서이름	근무도시	근무국가	부서장이름
100	김사원	1000	20/05/28	99	인사부	인천	한국	김부장
110	박사원	1200	20/01/28	98	전산부	런던	영국	박부장
120	이사원	1100	20/03/28	99	인사부	인천	한국	김부장

사원테이블

사번	이름	급여	입사일	상사사번	부서코드
100	김사원	1000	20/05/28	99	1
110	박사원	1200	20/01/28	98	2
120	이사원	1100	20/03/28	99	1

부서테이블

부서코드	부서이름	근무도시	근무국가	부서장이름
1	인사부	인천	한국	김부장
2	전산부	런던	영국	박부장
3	총무부	서울	한국	김부장

join – 열 합침-

테이블1.행 + 테이블2.행--> 1개행 열 결합

291p

union – 행 합침

select 사번, 이름 from 사원 ;->100

select 아이디, 이름 from 회원 ;-.200

select 사번, 이름 from 사원

union

select 아이디, 이름 from 회원 ;

사원- 100행

사번(문자) 이름(문자)

회원-200행

아이디(문자) 이름(문자)

회원

아이디(문자) 이름(문자)

-재난지원금 50번 부서나 급여 5000 이하 급여 받는 사원에게 지급(중복자 제외)

```
select first_name, department_id, salary
```

```
from employees
```

```
where department_id=50 or salary <= 5000;
```

-->54

```
select first_name, department_id, salary
```

```
from employees
```

```
where department_id=50
```

```
union
```

```
select first_name, department_id, salary
```

```
from employees
```

```
where salary <= 5000;
```

--> 54(50번 부서이고 급여 5000 이하 중복 1번 조회)

-재난지원금 50번 부서나 급여 5000 이하 급여 받는 사원에게 지급

(중복자 2번)

```
select first_name, department_id, salary
```

```
from employees
```

```
where department_id=50
```

```
union all
```

```
select first_name, department_id, salary
```

```
from employees
```

```
where salary <= 5000;
```

-재난지원금 50번 부서나 급여 5000 이하 급여 받는 사원에게 지급

(50부서 조건 만족 사원만)

```
select first_name, department_id, salary
from employees
where department_id=50
minus
select first_name, department_id, salary
from employees
where salary <= 5000;
```

-재난지원금 50번 부서나 급여 5000 이하 급여 받는 사원에게 지급

(50부서와 급여 5000 이하 두 조건 모두 만족)

```
select first_name, department_id, salary
from employees
where department_id=50
intersect
select first_name, department_id, salary
from employees
where salary <= 5000;
```

- 집합연산자

집합 = select 조회결과 테이블

합집합 = union / union all

교집합

차집합

union--> 2조건 만족하거나 1조건만 만족하는 사원들 모두 조회(1회)

union all -->2조건 만족하거나 1조건만 만족하는 사원들 모두 조회(2회)

minus --> 조건1 만족하지만 조건2 만족하지 못하는 사원 조회

intersect --> 2조건 만족 사원"만" 조회(1회)

```
select *  
  
from employees  
  
where department_id not in (William부서코드 조회 2명 - 10, 50 )
```

8장

```
select
```

1>DDL - 테이블 정의 / 구조 변경 /삭제

테이블 정의

```
CREATE TABLE 테이블이름 (  
  
    컬럼명1 타입(길이) 제약조건 ,  
  
    컬럼명2 타입(길이) 제약조건 ,  
  
    .....  
  
    컬럼명N 타입(길이) 제약조건  
  
);
```

```
ALTER TABLE 테이블이름 (ADD 컬럼명10 타입(길이) 제약조건 );
```

```
ALTER TABLE 테이블이름 (MODIFY 컬럼명1 ???(???) ????? );
```

```
ALTER TABLE 테이블이름 (DROP COLUMN 컬럼명1);
```

```
DROP TABLE 테이블이름;
```

```
CONN hr/hr
```

- db 테이블 소유주=schema = 사용자 = 계정

- 사용자 생성 권한 - systeml 계정

```
conn system/system;
```

```
create user jdbc identified by jdbc;
```

```
grant resource, connect to jdbc;--> data control lang(system)
```

```
(revoke resource, connect from jdbc)
```

```
conn jdbc/jdbc;
```

테이블/컬럼- 숫자시작불가능, 오라클키워드불가능, 길이제한, _

-jdbc

emp 테이블

id 정수 5자리	사번
name 문자열 20자리	이름
title 문자열 20자리	직급
dept_id 정수 5자리	부서코드
salary 실수 정수 10자리.소수점 2자리	급여

```
create table emp (
```

```
id number(5) ,
```

```
name varchar2(4000),
```

```
title varchar2(20),
```

```
dept_id number(5),
```

```
salary number(12 , 2)
```

```
);
```

- emp 테이블에 입사일 저장 컬럼 추가

```
alter table emp add indat date;
```

dml 데이터 저장 – 수정 – 삭제

emp

(100, '이사원', '사원', 10, 99000.5)

```
insert into emp values(100, '이사원', '사원', 10, 99000.5);
```

```
insert into emp values(200, '김대리', null, null, null);
```

```
insert into emp(id,name) values(300, '박과장');
```

```
insert into emp values(400, '최부장', '부장', 20, 99000.5);
```

```
insert into emp values(500, '박대리', '대리', 20, 99000.5);
```

- emp 테이블에서 급여 못받는 사원의 급여를 1000 수정

```
update emp
```

```
set salary = 1000
```

```
where salary is null;
```

- 이름이 박대리 사원의 부서를 이사원의 부서로 이동 .

```
update emp
```

```
set dept_id = (select dept_id from emp where name='이사원' and rownum=1)
```

```
where name='박대리';
```

```
commit;
```

```
select dept_id from emp where name='이사원' and rownum=1;
```

```
delete emp where id=100;
```


run sql command line

dml

insert 수행 후 ---> db 영구 저장 sql | 취소 sql

insert 수행 후 --> (메모리 임시 저장 - 다른 세션 결과 못본다)

---> commit---> db 반영 - 저장 ---> 다른 session insert 결과 반영

insert 수행 후 --> (메모리 임시 저장 - 다른 세션 결과 못본다)

---> rollback---> 메모리 삭제 --> 다른 session insert 결과 미반영

(자바 sql 연동 - 가장 오류)

TCL – TRANSACTION CONTROL LANG.

INSERT --> COMMIT

INSERT --> ROLLBACK

UPDATE --> COMMIT -> UPDATE

commit / rollback

ddl- 자동 commit drop table emp;	create alter drop
dml – commit/rollback 결정 transaction 처리 언어 1> commit 하지 않은 상태 / rollback 상태이면 다른 세션이 처리 결과 미반영 2> commit하면 다른 세션의 처리 결과 반영 3>	insert update delete

오라클 run sql...- 오라클연결-update 실행(where id=100)--> commit

자바프로그램—오라클연결—update 실행(where id=100) 대기중

```
insert into 테이블명[(컬럼명,,,,)] values (값, ' ' , ' ' , );
```

```
insert into 테이블명[(컬럼명,,,,)] values (값, ' ' , ' ' , );
```

```
commit;
```

```
update 테이블명 set 변경컬럼명=변경값 where 변경조건식;
```

```
update 테이블명 set 변경컬럼명=변경값;--> 테이블 모든 행 변경
```

--삭제

```
delete 테이블명 where 삭제조건식; -> 조건 맞는 데이터행 삭제. 테이블 구조 남김
```

```
delete 테이블명; --> 테이블 모든 데이터행 삭제. 테이블 구조 남김
```

```
===> rollback
```

```
drop table 테이블명 --> 테이블 모든 데이터행 삭제 + 테이블 구조 삭제
```

```
select ~ where
```

```
update ~ where
```

```
delete ~ where
```

- emp 테이블에 title 컬럼 길이 20->10자리 변경

(길이 축소 – 항상 실행 x)

name --> 20자리 넘는 데이터를 저장 상태

title --> 10자리 넘는 데이터를 저장 상태

```
alter table emp modify title varchar2(10);
```

alter table emp modify name varchar2(20);

- emp 테이블에서 입사일 컬럼 삭제

alter table emp drop column indate;

- emp 테이블 삭제

drop table emp;==> sql 문장 복구 불가

- update 시 subquery

update 테이블명

set 컬럼이름= (subquery)

where 컬럼이름 연산자 (subquery)

- delete subquery

delete 테이블명

where 컬럼이름 연산자 (subquery)

- insert 시 subquery

insert into emp values(.....)--> 1행 삽입

insert into emp(id,name,title, dept_id, salary)

select employee_id, first_name, job_id, department_id, salary from employees;

->107개 삽입

(hr – grant select on employees to jdbc;)

insert into emp values()

select ~

select employee_id, first_name, job_id, department_id, salary

from hr.employees;

emp =- 컬럼갯수,컬럼타입

employees -컬럼갯수,컬럼타입

- insert update delete –subquery

- create table emp(컬럼1 타입(길이) , , , ,)

- employees 테이블처럼 11개컬럼 이름,타입, 자리 그대로 복사 테이블 생성+데이터복사

create table emp_copy

as

select * from hr.employees;

- 자동으로 증가하는 시퀀스 = 객체

create table

create user

create sequence

100

101

102

insert into emp values(????, '이사원', '사원', 10, 99000.5);

insert into emp values(????, '김대리', null, null, null);

숫자데이터값을 자동 증가 역할 객체=시퀀스

1. 시퀀스 생성

create sequence 시퀀스이름;

--> 1부터 시작 1씩 증가 ?????

create sequence 시퀀스이름(

start with 10,

increment by 5,

maxvalue 100

);

--> 10부터 시작 5씩 증가 100

2. 시퀀스 활용

시퀀스명.currval --> 10 10 10

시퀀스명.nextval -> 10 15 20

dual

insert into emp values(emp_seq.nextval, '이자바', '사원', 30, 45000.55);

6 8

insert into dept values(emp_seq.nextval, '부서명','서울');

7 9

==> 사번 중복 x

3> 수정 삭제

alter sequence 시퀀스이름 start with 10

alter sequence 시퀀스이름 increment by 5

drop sequence 시퀀스이름;

create / alter / drop

insert (시퀀스) / delete / update

subquery

329p

- 제약조건 = 현실세계 모델링 = 테이블 데이터 모순 없어야 한다

constraint

1. 중복 x --> unique
2. null 값 허용 x --> not null
3. 중복 x + null x -> primary key
4. 다른 테이블 포함 값 사용 가능 --> foreign key
5. 사용자 조건 --> check

c_dept-모든 부서 정보	c_emp- 모든 사원 정보
dept_id 10 20	emp_id
dept_name 인재개발부 교육부	emp_name
city 제주 서울	title
	salary
	dept_id

create table c_dept(

dept_id number(5) constraint c_dept_id_pk primary key,

dept_name varchar2(20) constraint c_dept_name_uk unique,

city varchar2(20) constraint c_dept_city_nn not null

);

```
create table c_emp(  
emp_id number(5) constraint c_emp_emp_id_pk primary key,  
emp_name varchar2(20) constraint c_emp_name_nn not null,  
title varchar2(10) constraint c_emp_title_ck check (title in ('사원', '대리', '과장','부장','임원')),  
salary number(12, 2) constraint c_emp_salary_ck check(salary >= 1000),  
dept_id number(5) constraint c_emp_dept_id_fk references c_dept(dept_id)  
);
```

---> create table ddl 제약조건 정의

insert update delete dml 제약조건 효력 발생

```
insert into c_dept values(10, '인재개발부', '제주');
```

```
insert into c_dept values(20, '교육부', '서울');
```

```
insert into c_dept values(30, '전산개발부', '대전');
```

```
commit;
```

```
select * from c_dept;
```

```
insert into c_emp values(100, '김사원','사원',1000,10);
```

```
commit;
```

```
insert into c_emp values(200, '박대리','대리',1999,10);
```

```
insert into c_emp values(300, '안대리','대리',1999,20);
```

```
insert into c_emp values(400, '박과장','과장',3000,30);
```

```
insert into c_emp values(500, '박부장','부장',5000,10);
```

```
commit;
```

```
1>update c_emp set dept_id=20 where dept_id=10;
```

2>delete c_dept where dept_id=10;

drop table c_dept;

==> 실행오류(c_emp 테이블이 참조중(자식 테이블 c_emp)

drop table c_dept cascade constraints;

==> ok

alter table c_emp rename 이전컬럼명 to 새컬럼명;

rename 이전테이블명 to 새로운테이블명;

아이디입력 - xxxx (자바 + db 연동) --> db c_emp 저장(sql)--> db 전송-->오라클

dto dao

2>DML - 데이터저장 / 수정 / 삭제

3> TCL - 트랜잭션처리

- 테이블 생성

- SUBQUERY

- 제약조건

- 제약조건

- c_emp 테이블

emp_id 5자리정수 중복x, null x

name 30자리문자열 null x

salary 소수점이하 2(전체10) 1000 이상

indate 날짜

title 30자리문자열 사원 대리 과장 부장 임원 중 1개 가능

dept_id 5자리 정수 c_dept 테이블에 존재하는 부서만 가능

```
create table c_dept(  
dept_id number(5) constraint c_dept_id_pk primary key,  
dept_name varchar2(100) constraint c_dept_name_uk unique,  
city varchar2(100) constraint c_dept_name_nn not null  
);
```

```
insert into c_dept values (10, '인재개발부', '제주');
```

```
insert into c_dept values (20, '교육부', '서울');
```

```
insert into c_dept values (30, '전산개발부', '부산');
```

```
commit;
```

```
create table c_emp
```

```
(emp_id number(5) constraint c_emp_id_pk primary key,
```

```
name varchar2(30) constraint c_emp_name_nn not null,
```

```
salary number(10,2) constraint c_emp_sal_ck check (salary >= 1000 ),
```

```
indate date ,
```

```
title varchar2(30) constraint c_emp_title_ck check
```

```
(title in('사원', '대리', '과장', '부장', '임원') ),
```

```
dept_id number(5) constraint c_emp_dept_id_fk references c_dept(dept_id)
```

```
);
```

```
alter table c_emp drop constraint c_emp_dept_id_fk;
```

```
alter table c_emp add constraint c_emp_dept_id_fk foreign key (dept_id) references c_dept(dept_id);
```

drop table c_dept;-->참조중이므로 오류

drop table c_dept cascade constraints;

desc user_constraints;

select CONSTRAINT_NAME, CONSTRAINT_TYPE, SEARCH_CONDITION from user_constraints

where owner='HR'

and table_name='EMPLOYEES';