

- 리뷰

SQL 종류

DQL	SELECT	조회
DDL	CREATE TABLE ALTER TABLE DROP TABLE	테이블 정의 수정 삭제
DML	INSERT UPDATE DELETE	테이블 내의 데이터 =레코드(여러 컬럼)저장 수정 삭제
DCL	GRANT REVOKE	DB 사용 권한 부여 회수 (단 DBA 만 사용 가능-system) hr 계정 명령어 사용불가
TCL	COMMIT ROLLBACK	트랜잭션 처리

SELECT 문법

SELECT	필수. 컬럼명,,, * 연산식 별칭 distinct 함수
FROM	필수. 테이블명
WHERE	생략가능. 조회 레코드 조건. 레코드 수 줄어든다
GROUP BY	생략가능. 집계함수 적용 그룹 컬럼명
HAVING	생략가능. 집계함수 조건식
ORDER BY	생략가능. 정렬 순서 컬럼명(index, 별칭) asc desc , , ,

join query

SELECT	컬럼명 - 2개이상 테이블 분리	
FROM	테이블명1, 테이블명2 --> 테이블1 레코드갯수*테이블2 레코드갯수	
WHERE	테이블명1.공통컬럼=테이블명2.공통컬럼 (컬럼 이름 동일 경우 대부분이다, 컬럼 성격 동일 = 내 사번, 상사 사번	
	oracle join 문법	ansi join 문법-나중. DB 표준
inner join	SELECT EMPLOYEE_ID, DEPARTMENT_NAME FROM EMPLOYEES E, DEPARTMENTS D WHERE E.DEPARTMENT_ID= D.DEPARTMENT_ID	SELECT EMPLOYEE_ID, DEPARTMENT_NAME FROM EMPLOYEES E INNER JOIN DEPARTMENTS D ON E.DEPARTMENT_ID= D.DEPARTMENT_ID WHERE
outer join	SELECT EMPLOYEE_ID, DEPARTMENT_NAME FROM EMPLOYEES E,	... LEFT(RIGHT) OUTER JOIN .. ON

	DEPARTMENTS D WHERE E.DEPARTMENT_ID(+)= D.DEPARTMENT_ID	
self join	SELECT E1.FIRST_NAME 사원이름 MANAGER_ID, E2.FIRST_NAME 상사이름 FROM EMPLOYEES E1, EMPLOYEES E2 WHERE E1.MANAGER_ID = E2.EMPLOYEE_ID	? INNER JOIN ? – ON OUTER JOIN - ON

E.DEPARTMENT_ID=D.DEPARTMENT_ID(+)
 사원의 부서코드 각 부서의 부서코드
 NULL 존재 NULL 존재하지 않는다
 ==> 사원 부서명 조회하되 부서 없는 사원 "포함"

E.DEPARTMENT_ID(+)=D.DEPARTMENT_ID
 사원의 부서코드 각 부서의 부서코드
 270 부서원 없다 270번 부서 존재
 ==> 우리 회사 모든 부서명, 부서원 사원 조회하되 부서원 없는 부서 "포함"

subquery – 조회 결과를 또다른 조회 필요

SELECT	(select...) SELECT SALARY, SUM(SALARY) FROM EMPLOYEES-->ERROR SELECT SALARY, (SELECT SUM(SALARY) FROM EMPLOYEES) FROM EMPLOYEES-->OK
FROM	(select..)
WHERE	단일행 subquery (select..) WHERE 컬럼명 = (SELECT.. 조회 결과 1개 리턴) = > >= < <= != 다중행 subquery (select..) WHERE 컬럼명 IN (SELECT.. 조회 결과 N개 리턴) > ANY , > ALL

	< ANY , < ALL, = ANY
--	-------------------------

6,7,8

6 장 SELECT / 6.5절 DML

7 장 데이터타입, 함수, JOIN, SUBQUERY

8 장 테이블 ...-DDL

4장 데이터베이스 모델링 111P

- db 테이블 설계

프로젝트 구현 순서

업무 분석

프로젝트 설계 - 자바, DB 저장 ,AI--> 오라클 , 계정, DB 저장 모델링, 테이블정의

회원+상품+구매테이블

1> 구매하지 않은 회원의 NULL 컬럼 많다

2> 모든 회원 = 회원테이블

3> 구매한 회원 = 상품구매테이블

4> NULL컬럼 많은가 , 중복 레코드 여러개인가

회원테이블

상품구매테이블

통합

테스트

배포

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(EMPLOYEES)

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(DEPARTMENTS)

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

10시10분

- DDL

- 테이블 문법

이름 - 숫자시작불가능, 예약어사용불가능, _ 30문자길이제한

타입 - number(5,2) , varchar2 (byte) , date('rr/mm/dd')

create table 테이블명 (

컬럼명1 타입(길이) ,

컬럼명2 타입(길이) ,

컬럼명3 타입(길이) ,

..... ,

컬럼명n 타입(길이));

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(20) ,  
title varchar2(20),  
dept_id number(5),  
salary number(12, 2)  
);
```

emp 테이블 컬럼 1개 추가- 테이블 구조 변경(컬럼추가, 컬럼타입수정,컬럼삭제)

indate date	입사일
-------------	-----

```
alter table 테이블명 add 컬럼명1 타입(길이); ->컬럼추가  
alter table emp add indate date;
```

```
alter table 테이블명 drop column 컬럼명1 ;->컬럼삭제  
alter table emp drop column indate;
```

```
alter table 테이블명 modify 컬럼명1 타입(길이) ;->컬럼의 타입이나 길이 수정
```

title 컬럼 한글 10글자(한글 1글자 3바이트) 수정

```
alter table emp modify title varchar2(30);
```

(emp 테이블에 데이터가 삽입되지 않은 상태라면 수정 가능

단, emp 테이블에 데이터가 삽입된 상태라면 컬럼 길이 줄이는 건 불가능할 수도 있다.

타입 수정 불가능)

```
alter table emp add indate date;
```

indate 컬럼명을 hiredate 수정

```
alter table emp rename column indate to hiredate;
```

drop table 테이블명;
 drop table emp;--> 테이블 영구 삭제

sql developer – 메뉴
 sql 문법 입력

ddl – data definition language-테이블 대상
 dml- 테이블 레코드 대상

삽입	수정	삭제
insert into 테이블명 values(컬럼1값, 컬럼2값, ...); ==> 모든 컬럼 값 구성 insert into 테이블명(컬럼3, 컬럼5, ...) values(컬럼3값, 컬럼5값, ...); ==> 지정 컬럼 값 구성 (지정되지 않는 컬럼값 null)	update 테이블명 set 컬럼명1 = 변경값; --> 테이블 "모든" 레코드의 컬럼명1 변경 update 테이블명 set 컬럼명1 = 변경값 where 변경 조건식; --> 테이블 변경 조건 맞는 레코드의 컬럼명1 변경	delete 테이블명; --> 테이블 "모든" 레코드 삭제. 복구 가능 (drop table 테이블명 – 영구적 삭제-복구불가) delete 테이블명 where 삭제조건식; --> 테이블 삭제조건식 레코드 삭제. 복구 가능

emp 테이블 모든 컬럼 값 삽입

```

100 김사원 신입사원 20 1234567890.11 '21/08/16';
200 박대리 3년차대리 20 1234567990.11, '18/08/16';
101 최신입 신입사원 10 1234567890.11, sysdate;
201 이대리 1년차대리 null null '2020/12/12';
202 박과장 1년차과장 null null '01-01-2010';

insert into emp values( 100, '김사원', '신입사원', 20, 1234567890.11 , '21/08/16' ) ;
insert into emp values( 200, '박대리', '3년차대리', 20, 1234567990.11 , '18/08/16' ) ;
insert into emp values( 101, '최신입', '신입사원', 10, 1234567890.11 , sysdate) ;
insert into emp values( 201, '이대리', '1년차대리', NULL, null, '2020/12/12' );--> 명시적 null 삽입
==>
insert into emp(id, title, name, hiredate) values( 201, '1년차대리', '이대리', '2020/12/12' ) ;
-> 자동 null 삽입

insert into emp(id, title, name, hiredate) values( 202, '1년차과장', '박과장',
to_date('01-01-2010' , 'mm-dd-yyyy' ) ) ;
  
```

- emp 테이블에서 오늘 입사 직원 이름, 입사일 조회

```
select name, hiredate
from emp
where to_char(hiredate, 'rr/mm/dd') = to_char(sysdate, 'rr/mm/dd') ;

where hiredate = sysdate ;
```

date – rr/mm/dd.조회 기본 포맷. 년월일요일시분초 저장구성

0-49 : 2000

50-99 : 1900

2010/01/01 --> 날짜형식 인정(자동형변환)

to_date('01-01-2010' , 'mm-dd-yyyy') --> 명시적형변환

```
SELECT to_char(hiredate, 'rr/mm/dd HH24:MI:SS' ), to_char(sysdate, 'rr/mm/dd HH24:MI:SS')
FROM EMP;
```

- 박과장의 급여 900000

```
update emp
set salary = 900000
where name='박과장';
```

- 신입사원들의 급여 800000

```
update emp
set salary = 800000
where title='신입사원';
```

- 대리들의 급여 850000

```
update emp
set salary = 850000
where title like '%대리%';
```

```
where title = '1년차대리' or title='3년차대리';
```

```
where title in( '1년차대리' , '3년차대리' );
```

```
delete emp where name='이과장';
```

delete emp where title like '%과장';

- SUBQUERY – 다른 sql 문장 포함 select

select ... (select)

insert – subquery

- employees 테이블의 레코드 몇개를 (전부) 복사 emp 테이블

insert into emp values(.....)

employees	emp
employee_id number(5)	id number(5)
first_name varchar2(20)-성 제외	name varchar2(30) – 성 포함

<insert>

insert into emp values(.....) --> 1개 레코드 저장

insert into emp select * from employees; --> error

insert into emp(id, name, title, dept_id, salary, hiredate) select employee_id, first_name, job_id, department_id, salary, hire_date from employees where salary >= 15000; -->3

<update>

- Steven의 title 을 '사장' 변경

update emp set title='사장' where name='Steven';

- Steven의 title 을 박과장의 title로 변경

update emp set title=????? where name='Steven';

update emp

set title=(select title from emp where name='박과장')

where name='Steven';

select name, title from emp;

- Steven과 같은 title을 가진 사원의 급여를 2배 인상

update emp

set salary = salary * 2

where title = (select title from emp where name='Steven')

- 김사와 같은 title을 가진 사원의 급여를 이대리의 급여로 인상

update emp

set salary = (select salary from emp where name='이대리')

where title = (select title from emp where name='김사원')

<질문>

insert into emp employees

아까 쓴 명령어 두번 쳐서 지금 스티븐 니나 렉스가 두명씩 있는데 중복된 칼럼 삭제하는 방법이 있나요?

===>

delete emp where salary <= 30000; --> 삭제

insert ...

<delete>

니나 삭제

delete emp where name='Neena';

니나와 같은 급여 받는 사원 삭제

delete emp where salary=(select salary from emp where name='Neena');

<자동으로 증가하는 시퀀스>- db 다르다

insert 레코드 저장할 때 숫자, 1 증가(1부터 시작 - 1씩 증가 - 100까지 종료)

insert 레코드 저장할 때 숫자, 1 증가(100부터 시작 - 10씩 증가 - 1000까지 종료)

100 사번 저장

101 사번 그다음 레코드 저장

102 사번 그다음 레코드 저장

오라클 자동 증가숫자값 생성-sequence

1> 정의

테이블 정의 / 시퀀스 정의 / 사용자 정의 - create

create sequence 시퀀스명; 1부터 시작 - 1씩 증가 - 최대값

create sequence 시퀀스명

start with 시작값

incment by 증가치

maxvalue 종료값

```
create sequence emp_id_seq  
start with 300  
increment by 1  
maxvalue 999
```

```
select emp_id_seq.nextval from dual;  
select emp_id_seq.nextval from dual;  
select emp_id_seq.nextval from dual;  
select emp_id_seq.currval from dual;
```

2> 레코드 저장할 때 활용

```
insert into emp values(emp_id_seq.nextval, '서부장', '고참부장', 20, 1000000, '05/01/01'); 305  
insert into emp values(emp_id_seq.nextval, '박부장', '고참부장', 20, 1000000, '05/01/01'); 306  
insert into emp values(emp_id_seq.nextval, '김부장', '고참부장', 20, 1000000, '05/01/01'); 307
```

```
select emp_id_seq.nextval from dual; 308 생략
```

```
insert into emp values(emp_id_seq.nextval, '최부장', '고참부장', 20, 1000000, '05/01/01'); 309
```

```
select max(id) + 1 from emp
```

3> 시퀀스 없이 효과

```
insert into emp values( (select max(id) + 1 from emp), '이부장', '고참부장', 20, 1000000, '05/01/01');
```

309

merge -> 1 개 sql (저장 또는 수정 모두 가능)

dml -

INSERT / DELETE / UPDATE

1> 데이터 조작

2> SUBQUERY

3> 영구 저장 수정 삭제 이후 영구 반영 / 취소 복구 선택

4> TRANSACTION CONTROL LANGUAGE - TCL

5> commit / rollback

6> 트랜잭션 – 논리적 1개의 업무이지만 단일 업무 여러개 구성.

여러개 단일 업무 모두다 완성 수행 / 성공 1개 업무 취소

여러개 sql (dml) 결합되어 1개 일 처리.

ALL OR NOTHING

commit / rollback

A-B계좌이체 1개 업무 – (A 계좌 출금 + B 계좌 입금 - 성공 - 계좌이체 성공 . commit)

A-B계좌이체 1개 업무 – (A 계좌 출금 + 은행전산망 / 전원 문제 중단+ B 계좌 입금 시도 못하고
상태 –계좌 이체 실패. 원상태 복구 . rollback)

1> 계좌 테이블 정의 – 2개 신규 계좌 가입

create table account(

code number(5),

balance number(8, 2));

==> (ddl - 오라클 자동 commit)

insert into account values(12345, 10000);

insert into account values(54321, 5000);

2> 계좌 이체

12345 계좌주인이 54321 계좌로 이체- 5000

--2-1> 12345 계좌에서 5000 출금

update account

set balance = balance – 5000

where code = 12345;

--2-2> 54321 계좌로 5000 입금

update account

set balance = balance + 5000

where code = 54321;

==> insert 2, update 2 sql (?????)

==> commit (완전 db 저장)

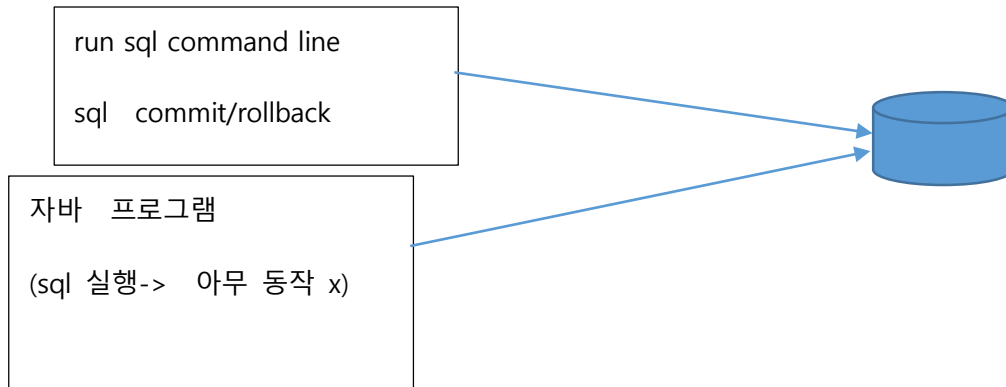
```
update account
set balance = balance - 1000
where code = 12345;
-- ok
```

```
update account
set balance = balance + '1000
where code = 54321;
-- error
==rollback
```

-트랜잭션 진행중

- 1> sql 실행 시작 - 자동 트랜잭션1 시작
- 2> ddl-create ... 실행 - 자동 commit - 완료 - 트랜잭션1 종료
- 3> 3-1. dml - insert delete update 실행 - commit - 완료 - 트랜잭션2 종료
 - 3-2. dml - insert delete update 실행 - rollback - 취소 원상복구 - 트랜잭션2 종료
- 4> dml - insert delete update 실행 - 트랜잭션3 시작
- 5> ...

conn hr/hr	conn hr/hr
1.delete emp where id=305;	
	2.select * from emp;--> delete 반영 이전 (트랜잭션 진행중)
3.commit;	
	4.select * from emp;--> delete 반영 (트랜잭 션 종료)
	5.delete emp where id=205;
	6.rollback;
7.select * from emp;--> delete 미반영 (트랜잭션 종료)	
8. update emp set salary = 700000 where title = '신입사원';	
	9. update emp set dept_id=30 where name = '김사원'; ==> 8번 sql 동일 레코드 9번 존재 8 sql 트랜잭션종료할 때까지 대기중(lock)



```
connect system/system;  
create user test identified by test;  
grant resource, connect to test;  
revoke resource, connection from test;
```

```
conn hr/hr;  
grant select on emp to test;
```

```
connect test/test;  
select * from hr.emp;
```

```
select  
create / alter / drop  
insert / update / delete  
commit / rollback  
grant / revoke
```

제약조건

- 1> 데이터 저장 / 수정 / 삭제 dml 사용시 데이터 상태 모순 방지
- 2> 데이터 제약조건 – integrity constraints

제약조건 종류 – 5개

not null	컬럼값 비어있으면 안된다 조건 지정
unique	컬럼값 중복되면 안된다 조건 지정
primary key	not null + unique 조건 지정(레코드 식별자)
check	사용자 정의 조건 지정
foreign key	다른 테이블의 컬럼에 포함 값들만 사용 조건 지정

create table 테이블명 (컬럼명1 타입(길이) ??????, , ,)

제약조건 문법

not null	id number(5) constraint emp_id_nn not null
unique	컬럼값 중복되면 안된다 조건 지정 id number(5) constraint emp_id_uk unique
primary key	not null + unique 조건 지정(레코드 식별자) id number(5) constraint emp_id_pk primary key
check	사용자 정의 조건 지정 id number(5) constraint emp_id_ck check (사용자 지정) id number(5) constraint emp_id_ck check (id >= 1000 and id <=2000)
foreign key	다른 테이블의 컬럼에 포함 값들만 사용 조건 지정 id number(5) constraint emp_id_fk references dept(dept_id)

1>

create table 테이블명 (컬럼명1 타입(길이) constraint emp_id_pk primary key , , ,)

2>

create table 테이블명 (컬럼명1 타입(길이), 컬럼명2 타입(길이) , ...,
constraint emp_id_pk primary key(컬럼명1) ,
constraint /...
)

3> alter table add 컬럼 1개 추가

alter table add constraint emp_id_pk primary key(컬럼명1)

c_emp 테이블	c_dept 테이블
emp_id 5자리정수 중복x, null x	dept_id 5자리정수 중복x, null x
name 30자리문자열 null x	dept_name 30자리 문자열 '부서' 끝난다
salary 소수점이하 2(전체10) 1000 이상	city 20자리 문자열
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(30) constraint c_dept_name_ck check (dept_name like '%부서'),
```

```
city varchar2(20)
```

```
);
```

```
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)
```

```
);
```

- 계정의 모든 테이블 이름 조회

```
select tname from tab;
```

```
select name from user_tables;
```

- 제약조건 지정 조회

```
select * from user_constraints;
```

```
desc user_constraints;
```

```
select constraint_name, constraint_type, search_condition
```

```
from user_constraints
```

```
where constraint_name='C_DEPT_ID_PK'; ---> (테이블명 대문자)
```

```
where table_name='C_EMP'; ---> (테이블명 대문자)
```

```
DESC C_EMP;
```

- 제약조건 정의 테이블 생성 DDL 정의

- C_DEPT, C_EMP 테이블에 DML 적용효과

```
INSERT INTO C_DEPT VALUES(10, '인사부서', '제주시');
```

```
INSERT INTO C_DEPT VALUES(20, '교육부서', '부산시');
```

```
INSERT INTO C_DEPT VALUES(30, '기획부서', '서울시');
```

```
INSERT INTO C_DEPT VALUES(NULL, '전산부서', '서울시');
```

```
==>INSERT INTO C_DEPT(DEPT_NAME, CITY) VALUES('전산부서', '서울시');
```

```
INSERT INTO C_DEPT VALUES(30, '기획부', '서울시');
```

```
insert into c_emp values(100, '박사원', 2000, sysdate, '사원', 10);
```

```
insert into c_emp values(200, null, 2000, sysdate, '사원', 10);
```

```
insert into c_emp values(200, '김대리', 5000, sysdate, '1년차대리', 10);
```

```
insert into c_emp values(200, '김대리', 5000, sysdate, '대리', 1000);
```

```
==>
```

```
insert into c_dept values(1000, '신생부서', '서울시');
```

```
insert into c_emp values(200, '김대리', 5000, sysdate, '대리', 1000);
```


- update – 제약조건 효력

update c_emp

set salary = salary – 500

where ?

- delete – 제약조건 효력

(foreign key !!!)

c_emp 테이블의 dept_id 컬럼이 c_dept 테이블의 dept_id 컬럼값 참조

자식 테이블

부모 테이블

1> 부모 테이블은 자식 테이블보다 먼저 정의된다.

2> 부모 테이블에서 값 삭제시 - delete c_dept where dept_id = 30;

자식 테이블에서 dept_id=30 값 참조중이라면 삭제 불가능

update c_emp set dept_id = 20 where dept_id=10;

or

delete c_emp where dept_id=10;

+

delete c_dept where dept_id=10;

5시 30분