

# 220929 수업 -1

🕒 작성일시	@2022년 9월 29일 오전 9:00
≡ 키워드	서브쿼리,
▼ 유형	강의
📎 자료	
☑ 복습	<input type="checkbox"/>

## 서브 쿼리 SUBQUERY

- 메인 쿼리 안에 들어 있는 쿼리

```
select avg(sal) from emp; -- 2073
```

```
select empno, ename, sal
from emp
where 2073 < sal*12;
```

## 2개의 쿼리를 하나로 통합

```
select empno, ename, sal
from emp
where sal > (select avg(sal) from emp);
```

### 1. single row subquery

- 실행 결과가 단일 row 단일 column인 경우

ex) select avg(sal) from emp;

- 연산자 : ( = != < > ) 사용 가능

### 2. multi row subquery

- 실행 결과가 단일 column, row 여러 개인 경우

ex) select sal from emp

- 연산자 : IN, NOT IN, ANY, ALL
  - ALL : sal > 1000 and sal > 4000 and ...
  - ANY : sal > 1000 or sal > 4000 or ...

## 문법

1. 괄호 안에 있어야 한다  
(select max(sal) from emp)
2. 단일 컬럼으로 구성  
select max(sal), sum(sal) from emp → (x)
3. 단독으로 실행 가능

## 실행 순서

1. subquery 먼저 실행
2. 결과를 가지고 메인 query 실행

## Tip)

select ... (subquery) >> 한개의 행만 반환할 경우 (Scala subquery)  
from (subquery) >> inline view(가상테이블) ...  
where (subquery) >> 조건

```
-- 사원 테이블에서 jones의 급여보다 더 많은 급여를 받는 사원 사번, 이름, 급여 출력
select empno, ename, sal
from emp
where sal > (select sal from emp where ename = 'JONES');
```

```
select *
from emp
where sal = (select sal from emp where deptno = 30);
-- ORA-01427: single-row subquery returns more than one row
```

- 오류 발생 다수의 row를 반환

```
select *
from emp
where sal in (select sal from emp where deptno = 30);
```

- in 사용

```
select *
from emp
where sal not in (select sal from emp where deptno = 30);
-- sal != 1600 and sal != 1250 and sal != 2850 and sal != 1500 and sal != 950
```

## 예제

- 부하직원이 있는 사원의 사번과 이름을 출력  
→ 내 사번이 mgr 컬럼에 있으면

```
select empno, ename
from emp
where empno in (select mgr from emp);
```

- 부하직원이 없는 사원의 사번과 이름을 출력  
→ or 연산에서는 null이 문제가 되지 않으나 and 연산에서는 영향을 줌  
→ and 연산에 null이 존재하면 결과는 모두 null

```
select empno, ename
from emp
where empno not in (select nvl(mgr, 0) from emp);
```

- king에게 보고하는 즉 직속 상관이 king인 사원의 사번, 이름, 직종, 관리자사번 출력

```
select empno, ename, job, mgr
from emp
where mgr = (select empno from emp where ename = 'KING');
```

order by 에 숫자 → select 컬럼 순서

```
select empno, ename, job
from emp
order by 3;
```

- 자기 부서의 평균 월급보다 더 많은 월급을 받는 사원의 사번, 이름, 부서번호, 부서별 평균 월급을 출력하세요

```
select e.empno, e.ename, e.deptno, m.avgsal
from emp e join (select deptno, round(avg(nvl(sal,0))) as avgsal from emp group by deptno) m
on e.deptno = m.deptno
where e.sal > m.avgsal;
```

## DML

명 령 어	설 명
INSERT	테이블에 새로운 행 추가
UPDATE	테이블의 행 내용을 변경
DELETE	테이블의 행 삭제
COMMIT	저장되지 않은 모든 변경 사항을 Database 에 저장
SAVEPOINT	savepoint 설정
ROLLBACK	저장되지 않은 모든 변경 사항을 취소

DDL(데이터 정의어) : create, alter, drop, truncate, rename, modify

DML(데이터 조작어) : insert.update, delete

DQL(데이터 질의어) : select

DCL(데이터 제어어) : 관리자 : grant, revoke

TCL(트랜잭션) : commit, rollback, savepoint

transaction : 하나의 논리적 작업 단위

---

## INSERT

```
create table temp(  
    id number primary key, --null(x) , 중복 데이터(x)  
    name nvarchar2(20)  
);
```

### 1. 가장 일반적인 insert

```
insert into temp(id,name)  
values(100, '홍길동');
```

### 2. 컬럼 목록 생략 (쓰지 말기)

```
insert into temp values(200, '김유신');  
  
select * from temp;  
rollback;
```

### 3. 문제

```
insert into temp(name)  
values('아무개');      --id pk 제약
```

```
insert into temp(id, name)
values(100, '개똥이'); -- 중복값x id pk 제약
```

### Tip

-- 일반적 SQL은 프로그램적 요소 없음  
--PL-SQL (변수, 제어문)

```
create table temp2(id varchar2(50));
/*
BEGIN
    for I IN 1..100 LOOP
        insert into temp2(id) values('A' || to_char(i));
    END LOOP;
END;
*/
select * from temp2;
```

```
create table temp3(
    memberid number(3) not null, --3자리 정수
    name varchar(10), --null 허용
    regdate date default sysdate -- 초기값 설정 (기본값)
);
```

- default : 기본값

```
select * from col where tname = 'TEMP3';
```

	⚡ TNAME	⚡ COLNO	⚡ CNAME	⚡ COLTYPE	⚡ WIDTH	⚡ SCALE	⚡ PRECISION	⚡ NULLS	DEFAULT
1	TEMP3	1	MEMBERID	NUMBER	22	0	3	NOT NULL	(null)
2	TEMP3	2	NAME	VARCHAR2	10	(null)	(null)	NULL	(null)
3	TEMP3	3	REGDATE	DATE	7	(null)	(null)	NULL	sysdate

## 팁

### 1. 대량 데이터 삽입

```
create table temp4(id number);
create table temp5(num number);

insert into temp4(id) values(1);
insert into temp4(id) values(2);
insert into temp4(id) values(3);
insert into temp4(id) values(4);
insert into temp4(id) values(5);
insert into temp4(id) values(6);
insert into temp4(id) values(7);
insert into temp4(id) values(8);
insert into temp4(id) values(9);
insert into temp4(id) values(10);
commit;
```

- 요구사항 temp4에 있는 모든 데이터를 temp5 넣고 싶다
  - insert into 테이블명(컬럼리스트) values
  - insert into 테이블명(컬럼리스트) select 절

```
insert into temp5(num)
select id from temp4; -- 대량 데이터 삽입
```

### 2. 대량 데이터 삽입

- 담을 테이블이 없는 경우 → 테이블 복제(구조 : 스키마)
  - 단, 제약 정보는 복제하지 않는다 (PK, FK)
- 순수한 데이터 구조 복제 + 데이터 복사

```
create table copyemp
as
select * from emp;
```

```
create table copyemp2
as
select empno, ename, sal
```

```
from emp
where deptno = 30;
```

- 틀(구조)만 복제하고 데이터는 복사하고 싶지 않아요

```
create table copyemp3
as
  select * from emp where 1=2;
```

## UPDATE



update 테이블명  
set 컬럼명 = 값, 컬럼명2 = 값2 .....  
where 조건절



update 테이블명  
set 컬럼 = (subquery)  
where 컬럼명 = (subquery)

```
update copyemp
set sal = 1111
where deptno = 20;
```

```
update copyemp
set sal = (select sum(sal) from emp);
```



```
update copyemp  
set ename='AAA', job='BBB', hiredate=sysdate, sal=(select sum(sal) from emp)  
where empno = 7369;
```

---

## Delete

```
delete from copyemp;
```

```
delete from copyemp where deptno=10;
```

---

하나의 테이블에 대해서 작업 JAVA 에서 EMP 테이블에서 CURD 작업

App(java)입장에서

1. 전체조회 (함수)
2. 조건조회 (함수)
3. 삽입 (함수)
4. 수정 (함수)
5. 삭제 (함수)

개발 >> JAVA 코드 > DAO

---

## DDL

- create, alter, drop, rename
- 테이블 생성, 수정, 삭제

---

## drop

```
drop table board;
```

---

## 테이블 정보 보기

```
select * from user_tables where lower(table_name) = 'board';
select * from col where lower(tname) = 'board'; --개발자
--제약정보
select * from user_constraints where lower(table_name) = 'board';
```

---

## 가상 컬럼 (조합 컬럼)

```
create table vtable(
    no1 number,
    no2 number,
    no3 number GENERATED ALWAYS as (no1 + no2) VIRTUAL
);
```

- 테이블 설계 시 미리 지정

```
--insert into vtable(no1, no2, no3) values(10, 20, 30); (x)
```

## 실예제

```
create table vtable2(
    no number, --순번
```

```

p_code char(4), --제품 코드(A001, B002)
p_date char(8), --입고일(20220922)
p_qty number, --입고수량
p_bungi number(1) GENERATED ALWAYS as (
    CASE WHEN substr(p_date, 5, 2) in ('01', '02', '03') then 1
         WHEN substr(p_date, 5, 2) in ('04', '05', '06') then 1
         WHEN substr(p_date, 5, 2) in ('07', '08', '09') then 1
         ELSE 4
    END ) VIRTUAL
);

```

- 실무에서 활용되는 코드
- 제품정보(입고) : 분기별 데이터 추출(4분기)
- 입고일 (2022-09-10)

```

insert into vtable2(p_date) values('20190101');
insert into vtable2(p_date) values('20190501');
insert into vtable2(p_date) values('20190601');
insert into vtable2(p_date) values('20191101');
insert into vtable2(p_date) values('20191201');
commit;

```

## Alter 테이블 수정

```

alter table temp6
add ename varchar2(20);

```

## rename 테이블 컬럼 이름 수정

```

alter table temp6
rename column ename to username;

```

## 기존 컬럼의 타입 크기 수정 → modify

```

alter table temp6
modify(username varchar2(2000));

```

## 기존 테이블 컬럼 삭제

```
alter table temp6
drop column username;
```

## 테이블 전체 삭제

- delete : 데이터만 삭제
  - 데이터 크기만큼 테이블 크기도 증가
  - 데이터만 삭제하면 테이블 크기 줄지 않음
- truncate
  - 공간 크기까지 삭제
  - where 불가

```
truncate table emp;
```

- drop : 테이블 삭제

```
drop table temp6;
```

## 데이터 무결성 제약 조건

- insert , update 제약 설정

제 약 조 건	설 명
PRIMARY KEY(PK)	유일하게 테이블의 각행을 식별 (NOT NULL 과 UNIQUE 조건을 만족)
FOREIGN KEY(FK)	열과 참조된 열 사이의 외래키 관계를 적용하고 설정합니다.
UNIQUE key(UK)	테이블의 모든 행을 유일하게 하는 값을 가진 열 (NULL 을 허용)
NOT NULL (NN)	열은 NULL 값을 포함할 수 없습니다.
CHECK(CK)	참이어야 하는 조건을 지정함 (대부분 업무 규칙을 설정)

---

## primary key

- not null
- unique
- 데이터 1건 보장 → 조회 성능 → index 자동 생성
- PK 테이블 당 1개만 사용 (컬럼 1개, 여러개 컬럼 묶어서 1개(복합키))

### 제약 정보 확인

```
select * from user_constraints where table_name='EMP';
```

### 테이블 생성

```
create table temp7(  
    --id number primary key -- 권장하지 않아요 (제약 이름이 자동 생성되서 찾기힘듦) >> SYS_C006997  
    -- 제약 삭제, 수정시 이름 필요함. 찾기 힘들다.  
    id number constraint pk_temp7_id primary key, -- pk_temp7_id 관용적인 표현  
    name varchar2(20) not null, --constraint 표현 쓰지 마세요. 그냥 써도 문제 x  
    addr varchar2(50)  
);
```

---

## UNIQUE key

- 테이블의 모든 행을 유일하게 하는 값을 가진 열(NULL 을 허용)
- 컬럼의 수만큼 생성 가능

```
create table temp8(  
    id number constraint pk_temp8_id primary key,  
    name nvarchar2(20) not null,  
    jumin nchar(6) constraint uk_temp8_jumin unique, --null가능  
    addr nvarchar2(1000)
```

```
);  
  
--일반적으로  
--jumin nchar(6) not null constraint uk_temp8_jumin unique,
```

---

## 기존 테이블에 제약 추가

- 제약 추가 시 테이블에 있는 데이터 검사

```
alter table temp9  
add constraint pk_temp9_id primary key(id);
```

→ add constraint pk\_temp9\_primary key(id, jumin); 복합키