/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT 문 - 행 추가

구문

- 한행추가 :

- INSERT INTO 테이블명 (컬럼 [, 컬럼]) VALUES (값 [, 값[])

- 모든 컬럼에 값을 넣을 경우 컬럼 지정구문은 생략 할 수 있다.

- 조회결과를 INSERT 하기 (subquery 이용)

- INSERT INTO 테이블명 (컬럼 [, 컬럼]) SELECT 구문

- INSERT할 컬럼과 조회한(subquery) 컬럼의 개수와 타입이 맞아야 한다.

- 모든 컬럼에 다 넣을 경우 컬럼 설정은 생략할 수 있다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

insert into emp (emp\_id, emp\_name, job\_id, mgr\_id, hire\_date, salary, comm\_pct, dept\_id)

values (1000, '홍길동', 'IT\_PROG', 120, '2019-07-15', 5000, 0.1, 60);

--모든 컬럼에 값을 넣을 경우 컬럼선택은 생략 가능, 값의 순서: 테이블 생성시 지정한 컬럼순

--Null : null값

--DATE : '년/월/일' 이외의 조합은 to\_date() 변환, sysdate : 실행시점의 일시를 반환하는 함수.

insert into emp values (1100, '박철수', null, 120, to\_date('2015/03','yyyy/mm'), 5000, 0.1, null);

insert into emp values (1200, '박철수', null, 120, sysdate, 5000, 0.1, null);

select \* from emp order by emp\_id desc;

--salary : NOT NULL 제약조건 => 반드시 값이 들어가야 한다.

insert into emp (emp\_id, emp\_name, hire\_date) values(1300,'이순신','2010/10/05');

--salary 정수부: 5자리 => 7자리(데이터 크기가 컬럼의 크기보다 크면 에러)

insert into emp (emp\_id, emp\_name, hire\_date, salary) values(1300,'이순신','2010/10/05', 1000000);

--제약조건: primary key(기본키) 컬럼에 같은 값을 insert 못함.

-- foreign key(외래키) 컬럼에는 반드시 부모테이블의 primary key 컬럼에 있는 값들만 넣을 수 있다.

insert into emp (emp\_id, emp\_name, hire\_date, salary, dept\_id) values(1400,'이순신이순신','2010/10/05', 10000, 500);

create table emp2( emp\_id number(6), emp\_name varchar2(20), salary number(7,2));

--emp에서 조회한 값을 emp2에 insert

insert into emp2 (emp\_id, emp\_name, salary)

select emp\_id, emp\_name, salary

from emp

where dept\_id = 10;

select \* from emp2;

--TODO: 부서별 직원의 급여에 대한 통계 테이블 생성.

-- 조회결과를 insert. 집계: 합계, 평균, 최대, 최소, 분산, 표준편차

drop table salary\_stat;

create table salary\_stat(

dept\_id number(6),

salary\_sum number(15,2),

salary\_avg number(10,2),

salary\_max number(7,2),

salary\_min number(7,2),

salary\_var number(20,2),

salary\_stddev number(7,2)

);

insert into salary\_stat

select dept\_id, sum(salary), round(avg(salary), 2), max(salary), min(salary), round(variance(salary),2), round(stddev(salary),2)

from emp

group by dept\_id

order by 1;

select \* from salary\_stat;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UPDATE : 테이블의 컬럼의 값을 수정

UPDATE 테이블명

SET 변경할 컬럼 = 변경할 값 [, 변경할 컬럼 = 변경할 값]

[WHERE 제약조건]

- UPDATE: 변경할 테이블 지정

- SET: 변경할 컬럼과 값을 지정

- WHERE: 변경할 행을 선택.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

-- 직원 ID가 200인 직원의 급여를 5000으로 변경

select emp\_id, salary from emp where emp\_id = 200;

update emp

set salary = 5000

where emp\_id = 200;

--rollback; (원상복구)

--commit;(현재 상태 완전 저장(커밋시 롤백이 안됨))

-- 직원 ID가 200인 직원의 급여를 10% 인상한 값으로 변경.

update emp

set salary = salary \* 1.1

where emp\_id = 200;

-- 부서 ID가 100인 직원의 커미션 비율을 0.2로 salary는 3000을 더한 값으로 변경.

select \* from emp where dept\_id = 100;

update emp

set comm\_pct = 0.2, salary = salary + 3000

where dept\_id = 100;

-- TODO: 부서 ID가 100인 직원들의 급여를 100% 인상

update emp

set salary = salary \* 2

where dept\_id = 100;

-- TODO: IT 부서의 직원들의 급여를 3배 인상

select \* from emp e, dept d where e.dept\_id = d.dept\_id and d.dept\_name = 'IT';

update emp

set salary = salary \* 3

where dept\_id = (select dept\_id from dept where dept\_name = 'IT');

-- TODO: EMP 테이블의 모든 데이터를 MGR\_ID는 NULL로 HIRE\_DATE 는 현재일시로 COMM\_PCT는 0.5로 수정.

update emp

set mgr\_id = null, hire\_date = sysdate, comm\_pct = 0.5;

rollback;

select \* from emp;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELETE : 테이블의 행을 삭제

구문

- DELETE FROM 테이블명 [WHERE 제약조건]

- WHERE: 삭제할 행을 선택

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

select \* from dept where dept\_id = 100;

delete from dept where dept\_id = 200;

delete from dept where dept\_id = 100; --emp테이블(자식테이블)에서 100번 부서를 참조하고있으므로 삭제가 불가능하다.

--자식테이블의 참조 값을 변경하거나 그 행을 삭제 한 뒤 처리한다.

delete from emp where dept\_id = 100;

rollback;

delete from emp;

select \* from emp;

-- TODO: 부서 ID가 없는 직원들을 삭제

delete from emp where dept\_id is null;

-- TODO: 담당 업무(emp.job\_id)가 'SA\_MAN'이고 급여(emp.salary) 가 12000 미만인 직원들을 삭제.

update emp

set mgr\_id = null

where emp\_id in (select emp\_id from emp where mgr\_id in (148,149));

delete from emp where job\_id = 'SA\_MAN' and salary <12000;

-- TODO: comm\_pct 가 null이고 job\_id 가 IT\_PROG인 직원들을 삭제

delete from emp where comm\_pct is null and job\_id = 'IT\_PROG';

create table emp2

as

select \* from emp;

select \* from emp2;

delete from emp2;

rollback;

/\*

truncate table 테이블명; =>DDL문. 자동 커밋

->전체 데이터를 삭제 (delete from 테이블명)

->rollback을 이용해 복구가 안된다.

\*/

/\*rollback의 중간지점 savepoint 이름

=>rollback 이름

ddl은 commit이 바로 됨

client tool을 종료시 바로 commit됨

\*/