

모든 문제는 변경한 후 select문으로 조회로 확인한 후 commit합니다

### 문제 1.

DEPTS테이블의 다음을 추가하세요

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
280	개발	null	1800
290	회계부	null	1800
300	재정	301	1800
310	인사	302	1800
320	영업	303	1700

### 문제 2.

DEPTS테이블의 데이터를 수정합니다

1. department\_name 이 IT Support 인 데이터의 department\_name을 IT bank로 변경
2. department\_id가 290인 데이터의 manager\_id를 301로 변경
3. department\_name이 IT Helpdesk인 데이터의 부서명을 IT Help로 , 매니저아이디를 303으로, 지역아이디를 1800으로 변경하세요
4. 이사, 부장, 과장, 대리 의 매니저아이디를 301로 한번에 변경하세요.

### 문제 3.

삭제의 조건은 항상 primary key로 합니다, 여기서 primary key는 department\_id라고 가정합니다.

1. 부서명 영업부를 삭제 하세요
2. 부서명 NOC를 삭제하세요

## 문제4

1. Depts 사본테이블에서 department\_id 가 200보다 큰 데이터를 삭제하세요.
2. Depts 사본테이블의 manager\_id가 null이 아닌 데이터의 manager\_id를 전부 100으로 변경하세요.
3. Depts 테이블은 타겟 테이블 입니다.
4. Departments테이블은 매번 수정이 일어나는 테이블이라고 가정하고 Depts와 비교하여 일치하는 경우 Depts의 부서명, 매니저ID, 지역ID를 업데이트 하고 새로유입된 데이터는 그대로 추가해주는 merge문을 작성하세요.

## 문제 5

1. jobs\_it 사본 테이블을 생성하세요 (조건은 min\_salary가 6000보다 큰 데이터만 복사합니다)
2. jobs\_it 테이블에 다음 데이터를 추가하세요

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
IT_DEV	아이티개발팀	6000	20000
NET_DEV	네트워크개발팀	5000	20000
SEC_DEV	보안개발팀	6000	19000

3. jobs\_it은 타겟 테이블 입니다
4. jobs테이블은 매번 수정이 일어나는 테이블이라고 가정하고 jobs\_it과 비교하여 min\_salary컬럼이 0보다 큰 경우 기존의 데이터는 min\_salary, max\_salary를 업데이트 하고 새로 유입된 데이터는 그대로 추가해주는 merge문을 작성하세요