

Three-Valued Logic

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

AND의 경우 FALSE 와의 조합은 모두 FALSE

TRUE 와의 조합은 모두 자기 자신

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	FALSE

OR의 경우 TRUE 와의 조합은 모두 TRUE

FALSE 와의 조합은 모두 자기 자신

NOT	TRUE	FALSE	UNKNOWN
NOT	FALSE	TRUE	UNKNOWN

Nested Queries

- WHERE절 내에 (select-from-where) 블록을 사용할 수 있다. In은 블록안에 해당element가 있으면 TRUE로 평가된다.
- (select-from-where) 블록과 비교할 요소에 tuple(e.g., (Pno, Hours))도 사용 가능.
- ANY : IN과 의미는 동일하나, 비교연산자(>,>=,=,<=,<) 사용가능.
- ALL : nested query전체 값을 비교해 모두 있어야 TRUE, 비교연산자 사용가능.
- AS : 잠재적 에러나 모호함을 피하기 위해 별칭 사용 가능.
- EXISTS : nested query에 존재여부를 TRUE,FALSE로 반환.

JOIN

- INNER JOIN : 두 table에서 join condition을 만족하는 tuple들로 result table을 만드는 JOIN
- OUTER JOIN : 두 table에서 join condition을 만족하지 않는 tuple들도 result table에 포함하는 JOIN
 - LEFT, RIGHT, FULL 존재.

Aggregate Function in SQL

- COUNT,SUM,MAX,MIN,AVG 등 Built-in function존재.

- GROUP : tuple의 subgroup을 만들어준다. 이때 조건은 WHERE이 아니라 HAVING을 사용해야 함.
- GROUP은 반드시 SELECT안에 존재해야 함.
- 그룹화 속성에 가능한 값이 NULL인 경우 해당 null 값에 대해 별도의 그룹이 생성.

Combining the WHERE and the HAVING Clause

- we want to count the total number of employees whose salaries exceed \$40,000 in each department, but only for departments where more than five employees work

%% 틀린 코드 %%

```
SELECT Dno, COUNT(*) FROM EMPLOYEE
WHERE Salary > 40000
GROUP BY Dno
HAVING COUNT(*) > 5;
```

%% 올바른 코드 %%

```
SELECT Dno, COUNT(*) FROM EMPLOYEE
WHERE Salary > 40000 AND Dno IN {
    (SELECT Dno FROM EMPLOYEE
     GROUP BY Dno
     HAVING COUNT(*) > 5)
}
GROUP BY Dno;
```

위 틀린코드와 같이 작성하면 Salary가 40000 이상인 테이블이 먼저 생기고, 이후 Dno으로 그룹하게 되면 그룹되는 직원들은 모두 40000 이상의 Salary를 가지는 직원들이라 문제의 요구사항과 달라진다.

ASSERTION

- 제약조건이 추가로 필요한 경우 CREATE ASSERTION으로 생성할 수 있다.

TRIGGER

- 데이터베이스에서 특정 이벤트가 발생할 때 자동으로 실행되는 특수한 유형의 저장 프로시저.

Concept of a view in SQL

- 다른 테이블에서 파생된 Single Table로 defining tables이라고도 불림.
- 반드시 채워질 필요는 없는 가상테이블.
- 가능한 업데이트를 제한한다.
- 쿼리에는 제한이 없다.

Schema evolution commands

- DBA는 DB가 동작하는 동안 바꾸고 싶을 수 있는데, 이때 DB schema를 recompilation할 필요 없다.
- DROP : CASCADE, RESTRICT 옵션을 선택해 schema를 제거할 수 있다.
- ALTER table action
 - Adding or dropping a column
 - Changing a column definition
 - Adding or dropping table constraints