

부속질의문과 중첩질의문

- SELECT 문의 결과가 테이블이기 때문에 검색문의 결과를 또다른 SQL의 입력으로 사용가능
- 부속질의문(sub-query): 다른 질의문에 포함되는 SQL 질의문
- 중첩질의문(nested query): 부속질의문을 하나 이상 포함하는 SQL 질의문
- FROM 절에 부속질의문 포함

```
SELECT *  
FROM (select * from student where dept = '컴퓨터') cs  
WHERE syear = 3
```

(해석) 컴퓨터과 학생 중 3학년인 학생들의 정보를 출력하라.

부속질의문과 중첩질의문

- WHERE 절에 부속질의문 포함

```
SELECT   sno
FROM     enrol
WHERE    final = (select max(final) from enrol where cno = 'C413')
```

(해석) 등록테이블에서 C413과목의 기말고사 점수 최대값하고 같은 기말고사 점수를 받은 학생의 학번을 검색하라.

- SELECT 절에 부속질의문 포함

```
SELECT   sno, (select count(cno) from enrol e where s.sno = e.sno) as course_count
FROM.    Student s
WHERE    syyear = 3
```

(해석) 3학년 학생들의 학번과 해당 학생이 신청한 과목의 수를 출력하라.

IN 을 이용한 중첩질의문

- WHERE 절에 IN 을 통해서 조건을 표시

```
SELECT      sname
FROM        STUDENT
WHERE        sno IN (SELECT sno FROM ENROL WHERE cno = 'C413');
```

(해석1) C413 과목을 수강하는 학생의 학번에 포함된 학생의 이름을 검색하라.

(해석2) C413 과목을 수강하는 학생의 이름을 검색하라.

- WHERE 절에 IN을 포함하는 중첩질의문은 조인질의문으로 변환이 가능함.

```
SELECT      sname
FROM        STUDENT s, ENROL e
WHERE        s.sno = e.sno and e.cno = 'C413';
```

NOT IN 을 이용한 중첩질의문

- WHERE 절에 NOT IN 을 통해서 조건을 표시

```
SELECT      sname
FROM        STUDENT
WHERE        sno NOT IN (SELECT sno FROM ENROL WHERE cno = 'C413');
```

(해석1) C413 과목을 수강하는 학생의 학번에 포함되지 않는 학생의 이름을 검색하라.

(해석2) C413 과목을 수강하지 않는 학생의 이름을 검색하라.

- WHERE 절에 NOT IN을 포함하는 중첩질의문은 조인질의문으로 변환이 어려움.

```
SELECT      sname
FROM        STUDENT s, ENROL e
WHERE        s.sno = e.sno and e.cno <> 'C413';
```

(해석) C413이 아닌 과목들을 수강한 학생의 이름을 검색하라.

즉, C413 과 다른 과목을 중복
수강한 학생은 table에 포함 된다.
왜 Not IN 에는 포함 X.

EXISTS 을 이용한 중첩질의문

- WHERE 절에 EXISTS 을 통해서 조건을 표시

```
SELECT      sname
FROM        STUDENT s
WHERE        EXISTS (SELECT *
                        FROM ENROL e
                        WHERE  e.sno = s.sno and e.cno = 'C413');
```

EXIST(subquery): 자체가 Boolean expression이고 subquery의 결과가 튜플이 존재하면 TRUE를, 튜플이 없으면 FALSE를 반환함.

- (해석1) C413 과목을 수강하는 ENROL 테이블 튜플이 있는 학생의 이름을 검색하라.
(해석2) C413 과목을 수강하는 학생의 이름을 검색하라.

- 조인질의문으로 변환이 가능함

NOT EXISTS 을 이용한 중첩질의문

- WHERE 절에 NOT EXISTS 을 통해서 조건을 표시

```
SELECT      sname
FROM        STUDENT s
WHERE       NOT EXISTS (SELECT *
                        FROM ENROL e
                        WHERE  e.sno = s.sno and e.cno = 'C413');
```

NOT EXISTS(subquery): 자체가 Boolean expression이고 subquery의 결과가 튜플이 존재하면 FALSE를, 튜플이 없으면 TRUE를 반환함.

- (해석1) C413 과목을 수강하는 ENROL 테이블 튜플이 없는 학생의 이름을 검색하라.
(해석2) C413 과목을 수강하지 않는 학생의 이름을 검색하라.

- 조인질의문으로 변환이 어려움.

UNION, INTERSECT를 이용한 중첩질의문

- 두개 이상의 SQL 검색문을 UNION, INTERSECT로 묶을 수 있음.

```
( SELECT  Sno
  FROM    STUDENT
  WHERE   Year = 1)
UNION
(SELECT  Sno
  FROM    ENROL
  WHERE   Cno = 'C324');
```

검색문의 결과들이 UNION-compatible 해야 함.

UNION: 합집합 테이블. 중복 튜플은 제거됨

UNION ALL: 중복을 제거하지 않는 합집합 테이블

INTERSECT: 교집합 테이블

*차집합은 NOT IN을 사용할 것.