고급 SQL

Database Laboratory

차례

- ▶ 예시 데이터
- ▶ 기본적인 질의 구조 및 연산자
- ▶ 집합연산자 & 조인 (JOIN)
- ▶ 집성 함수 (Aggregate Functions)
- ▶ 중첩 부질의 (Nested Subqueries)

예시 데이터

department

```
CREATE TABLE department (
dept_name VARCHAR(20),
building VARCHAR(15),
budget INT,
PRIMARY KEY(dept_name)
);
```

dept_name	building	budget
Comp.Sci.	Taylor	100000
Biology	Watson	90000
Elec.Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

예시 데이터

instructor

```
dept_na
CREATE TABLE instructor (
                                                                      id
                                                                                                     salary
                                                                               name
                                                                                           me
                                    VARCHAR(5),
            id
                                    VARCHAR(20) NOT NULL,
                                                                    22222
                                                                               Einstein
                                                                                          Physics
                                                                                                     95000
            name
                                    VARCHAR(20),
            dept_name
                                                                    12121
                                                                                Wu
                                                                                          Finance
                                                                                                     90000
            salary
                                    INT,
                                                                    32343
                                                                               El Said
                                                                                                     60000
                                                                                          History
            PRIMARY KEY(id)
);
                                                                    45565
                                                                                Katz
                                                                                        Comp.Sci.
                                                                                                     75000
                                                                    98345
                                                                                Kim
                                                                                         Elec.Eng.
                                                                                                     80000
                                                                    76766
                                                                                Crick
                                                                                                     72000
                                                                                         Biology
                                                                     10101
                                                                              Srinivasan
                                                                                                     65000
                                                                                         Comp.Sci.
                                                                    58583
                                                                               Clifieri
                                                                                         History
                                                                                                     62000
                                                                    83821
                                                                               Brandt
                                                                                         Comp.Sci.
                                                                                                     92000
                                                                    15151
                                                                               Mozart
                                                                                          Music
                                                                                                     40000
                                                                    33456
                                                                                Gold
                                                                                                     87000
                                                                                          Physics
```

76543

Finance

Singh

80000

예시 데이터

student

```
dept_na
CREATE TABLE student (
                                                                     id
                                                                                                   tot_cred
                                                                              name
                                                                                          me
                                    VARCHAR(5),
            id
                                    VARCHAR(20) NOT NULL,
                                                                   00128
                                                                                       Comp.Sci.
                                                                                                     102
                                                                              Zhang
            name
                                    VARCHAR(20),
            dept_name
                                                                    12345
                                                                             Shankar
                                                                                       Comp.Sci.
                                                                                                     32
            tot_cred
                                    INT,
                                                                    19991
                                                                              Brandt
                                                                                        History
                                                                                                     80
            PRIMARY KEY(id)
);
                                                                   23121
                                                                              Chavez
                                                                                        Finance
                                                                                                     110
                                                                   44553
                                                                              Peltier
                                                                                         Physics
                                                                                                     56
                                                                   45678
                                                                                         Physics
                                                                                                     46
                                                                               Levy
                                                                                                     54
                                                                    54321
                                                                             Williams
                                                                                       Comp.Sci.
                                                                    55739
                                                                             Sanchez
                                                                                         Music
                                                                                                     38
                                                                   70557
                                                                              Snow
                                                                                         Physics
                                                                                                      0
                                                                   76543
                                                                              Brown
                                                                                       Comp.Sci.
                                                                                                     58
                                                                   76653
                                                                               Aoi
                                                                                        Elec.Eng.
                                                                                                     60
                                                                   98765
                                                                             Bourikas
                                                                                        Elec.Eng.
                                                                                                     98
```

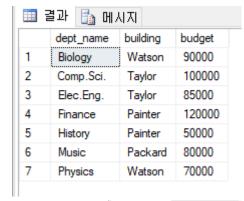
▶ 기본 구조

SELECT ··· FROM ··· WHERE ···

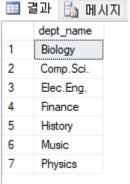
- ▶ SELECT 절
 - ▶ 질의의 결과로 바라는 컬럼명
- ▶ FROM 절
 - ▶ 질의 평가에 대한 모든 테이블
- ▶ WHERE 절
 - 질의 결과가 만족해야 하는 조건

- ▶ SELECT 절
 - 질의의 결과로 바라는 컬럼명

```
SELECT * FROM department
```



SQLQuery1.sql - ...IAC9H\u00fcUser (52))\u00e5 SELECT dept_name FROM department



- ▶ FROM 절
 - ▶ 질의 평가에 대한 모든 테이블

SQLQuery1.sql - ...IAC9H\User (52))*

SELECT * FROM student, instructor

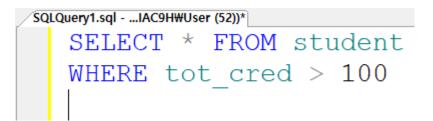
	Ⅲ 결과 🛅 메시지								
	id	name	dept_name	tot_cred	id	name	dept_name	salary	
1	00128	Zhang	Comp.Sci.	102	10101	Srinivasan	Comp.Sci.	65000	
2	12345	Shankar	Comp.Sci.	32	10101	Srinivasan	Comp.Sci.	65000	
3	19991	Brandt	History	80	10101	Srinivasan	Comp.Sci.	65000	
4	23121	Chavez	Finance	110	10101	Srinivasan	Comp.Sci.	65000	
5	44553	Peltier	Physics	56	10101	Srinivasan	Comp.Sci.	65000	
6	45678	Levy	Physics	46	10101	Srinivasan	Comp.Sci.	65000	
7	54321	Williams	Comp.Sci.	54	10101	Srinivasan	Comp.Sci.	65000	
8	55739	Sanchez	Music	38	10101	Srinivasan	Comp.Sci.	65000	
9	70557	Snow	Physics	0	10101	Srinivasan	Comp.Sci.	65000	
10	76543	Brown	Comp.Sci.	58	10101	Srinivasan	Comp.Sci.	65000	
11	76653	Aoi	Elec.Eng.	60	10101	Srinivasan	Comp.Sci.	65000	
12	98765	Bourikas	Elec.Eng.	98	10101	Srinivasan	Comp.Sci.	65000	
13	00128	Zhang	Comp.Sci.	102	12121	Wu	Finance	90000	
14	12345	Shankar	Comp.Sci.	32	12121	Wu	Finance	90000	
15	19991	Brandt	History	80	12121	Wu	Finance	90000	
16	23121	Chavez	Finance	110	12121	Wu	Finance	90000	
17	44553	Peltier	Physics	56	12121	Wu	Finance	90000	
18	45678	Levy	Physics	46	12121	Wu	Finance	90000	

▶ WHERE 절

▶ 질의 결과가 만족해야 하는 조건

```
SELECT * FROM student
WHERE tot_cred < 100
```

	id	name	dept_name	tot_cred
1	12345	Shankar	Comp.Sci.	32
2	19991	Brandt	History	80
3	44553	Peltier	Physics	56
4	45678	Levy	Physics	46
5	54321	Williams	Comp.Sci.	54
6	55739	Sanchez	Music	38
7	70557	Snow	Physics	0
8	76543	Brown	Comp.Sci.	58
9	76653	Aoi	Elec.Eng.	60
10	98765	Bourikas	Elec.Eng.	98





- > 재명명 연산
 - ▶ 테이블과 컬럼명을 재명명하는 연산

```
SQLQuery1.sql-...IAC9H\u00fcUser (52))\u00e9

SELECT i.name, s.name

FROM instructor AS i, student AS s
```

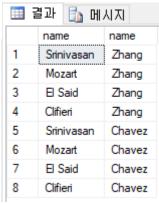
```
SQLQuery1.sql-...IAC9H\wvvser(52))*

SELECT i.name, s.name

FROM instructor AS i, student AS s

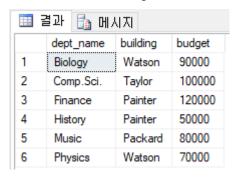
WHERE i.salary < 70000 AND s.tot_cred > 100
```

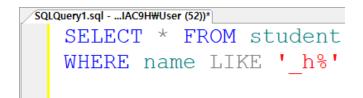


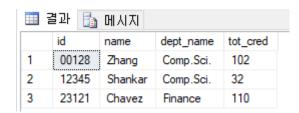


- ▶ 스트링 연산
 - ▶ 문자열 비교를 위한 문자열 매칭 연산자 (LIKE)
 - percent (%): The % character matches any substring.
 - ▶ underscore (_): The _ character matches any character.

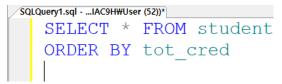
```
SQLQuery1.sql-...IAC9H\u00fcUser (52))\u00e4 SELECT \u00e9 FROM department \u00c4 WHERE dept_name LIKE \u00e4\u00e4i\u00b8'\u00e4
```

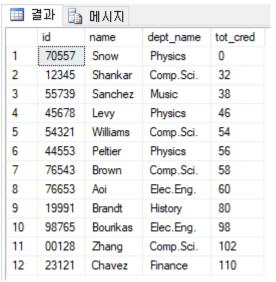


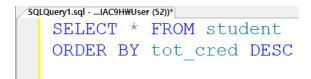




- ▶ 데이터 출력의 순서화
 - 각 컬럼에 대해 내림차순(desc) 또는 오름차순(asc)으로 정렬하여 출력
 - ▶ 오름차순이 기본 값







ⅲ 결과 🔒 메시지							
	id	name	dept_name	tot_cred			
1	23121	Chavez	Finance	110			
2	00128	Zhang	Comp.Sci.	102			
3	98765	Bourikas	Elec.Eng.	98			
4	19991	Brandt	History	80			
5	76653	Aoi	Elec.Eng.	60			
6	76543	Brown	Comp.Sci.	58			
7	44553	Peltier	Physics	56			
8	54321	Williams	Comp.Sci.	54			
9	45678	Levy	Physics	46			
10	55739	Sanchez	Music	38			
11	12345	Shankar	Comp.Sci.	32			
12	70557	Snow	Physics	0			

▶ 기본 구조

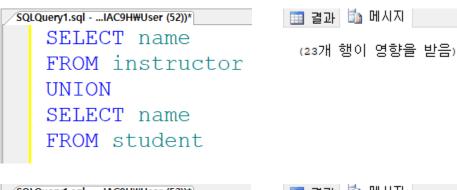
```
SELECT ...
[UNION | UNION ALL | INTERSECT | EXCEPT]

SELECT ...
```

- ▶ 두 SELECT문의 컬럼 개수와 데이터 타입은 일치해야 함.
- ▶ 검색 결과의 헤더는 앞쪽 SELECT문에 의해 결정된다.

연산자	의미	결과
UNION	하지하	중복을 제거한 결과의 합을 검색
UNION ALL	합집합	중복을 포함한 결과의 합을 검색
INTERSECT	교집합	양쪽 모두에서 포함된 행을 검색
EXCEPT	차집합	첫번째 검색 결과에서 두번째 검색 결과를 제외한 나머지를 검색

- ▶ UNION, UNION ALL (합집합)
 - ▶ 결과의 합을 검색

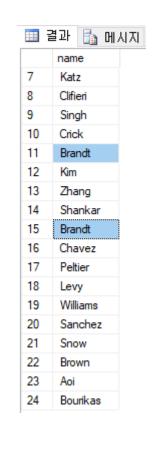


SQLQuery1.sql - ...IAC9H\u00fcUser (52))\u00e4
SELECT name
FROM instructor
UNION ALL
SELECT name
FROM student



(24개 행이 영향을 받음)





- ▶ INTERSECT (교집합)
 - ▶ 양쪽 모두에서 포함된 데이터를 검색

```
SQLQuery1.sql - ...IAC9HWUser (52))*

SELECT name
FROM instructor
INTERSECT
SELECT name
FROM student
```



▶ EXCEPT (차집합)

▶ 첫번째 검색 결과에서 두번째 검색 결과를

제외한 나머지를 검색

SQLQuery1.sql - ...IAC9H\User (52))*

SELECT name
FROM instructor
EXCEPT
SELECT name
FROM student



- ▶ 카티전 곱 (Cartesian Product)
- ▶ INNER JOIN
- OUTER JOIN(LEFT, RIGHT)

- ▶ 카티전 곱 (Cartesian Product)
 - ▶ 두 테이블에 대해서 가능한 모든 경우의 수를 계산

SQLQuery1.sql - ...IAC9H\u00fcUser (52))\u00e9 SELECT \u00e9 FROM student, department

- ▶ INNER JOIN (WHERE 절을 이용)
 - > 공통 컬럼의 값이 같은 데이터만을 매치하여, 결과로 공통 컬럼을 포함하는 데이터의 집합의 결과

```
SQLQuery1.sql-...|AC9H\u00fcUser (52))\u00e4

SELECT \u00e4

FROM student AS s, department AS d

WHERE s.dept_name = d.dept_name
```

	결과 🚡	메시지					
	id	name	dept_name	tot_cred	dept_name	building	budget
1	00128	Zhang	Comp.Sci.	102	Comp.Sci.	Taylor	100000
2	12345	Shankar	Comp.Sci.	32	Comp.Sci.	Taylor	100000
3	19991	Brandt	History	80	History	Painter	50000
4	23121	Chavez	Finance	110	Finance	Painter	120000
5	44553	Peltier	Physics	56	Physics	Watson	70000
6	45678	Levy	Physics	46	Physics	Watson	70000
7	54321	Williams	Comp.Sci.	54	Comp.Sci.	Taylor	100000
8	55739	Sanchez	Music	38	Music	Packard	80000
9	70557	Snow	Physics	0	Physics	Watson	70000
10	76543	Brown	Comp.Sci.	58	Comp.Sci.	Taylor	100000
11	76653	Aoi	Elec.Eng.	60	Elec.Eng.	Taylor	85000
12	98765	Bourikas	Elec.Eng.	98	Elec.Eng.	Taylor	85000

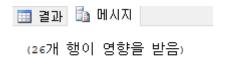
- ▶ INNER JOIN (INNER JOIN과 ON을 이용)
 - > 공통 컬럼의 값이 같은 데이터만을 매치하여, 결과로 공통 컬럼을 포함하는 데이터의 집합의 결과

```
SELECT *
FROM student AS s INNER JOIN department AS d
ON s.dept_name = d.dept_name
```

	id	name	dept_name	tot_cred	dept_name	building	budget
1	00128	Zhang	Comp.Sci.	102	Comp.Sci.	Taylor	100000
2	12345	Shankar	Comp.Sci.	32	Comp.Sci.	Taylor	100000
3	19991	Brandt	History	80	History	Painter	50000
4	23121	Chavez	Finance	110	Finance	Painter	120000
5	44553	Peltier	Physics	56	Physics	Watson	70000
6	45678	Levy	Physics	46	Physics	Watson	70000
7	54321	Williams	Comp.Sci.	54	Comp.Sci.	Taylor	100000
8	55739	Sanchez	Music	38	Music	Packard	80000
9	70557	Snow	Physics	0	Physics	Watson	70000
10	76543	Brown	Comp.Sci.	58	Comp.Sci.	Taylor	100000
11	76653	Aoi	Elec.Eng.	60	Elec.Eng.	Taylor	85000
12	98765	Bourikas	Elec.Eng.	98	Elec.Eng.	Taylor	85000

- OUTER JOIN(LEFT)
 - ▶ 왼쪽의 테이블이 기준

SQLQuery1.sql-...IAC9H\u00fcUser (52))*
SELECT i.name AS 교수, s.name AS 학생
FROM instructor AS i LEFT JOIN student AS s
ON i.dept name = s.dept name



17	Katz	Brown
18	Clifieri	Brandt
19	Singh	Chavez
20	Crick	NULL
21	Brandt	Zhang
22	Brandt	Shankar

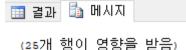
- OUTER JOIN(RIGHT)
 - ▶ 오른쪽의 테이블이 기준

```
SQLQuery1.sql ...IAC9H\u00faUser (52))*

SELECT i.name AS 교수, s.name AS 학생

FROM instructor AS i RIGHT JOIN student AS s
ON i.dept_name = s.dept_name
```





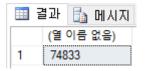
(25개 행이 영향을 받음

- AVG
- MIN
- MAX
- SUM
- COUNT
- ▶ GROUP BY, HAVING 절

- AVG
 - ▶ 평균값

```
SQLQuery1.sql - ...IAC9H\u00fcUser (52))*

SELECT AVG (salary) FROM instructor
```



- MIN
 - ▶ 최소값

```
SELECT MIN (salary) FROM instructor
```



- MAX
 - ▶ 최대값

```
SQLQuery1.sql - ...IAC9H\u00fcUser (52))*

SELECT MAX (salary) FROM instructor
```



- SUM
 - ▶ 모두 더한 값

```
SQLQuery1.sql - ...IAC9H\u00fcUser (52))\u00e5
SELECT SUM (salary) FROM instructor
```

```
    글과
    등
    메시지

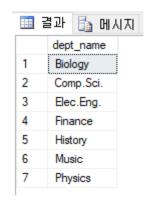
    (열 이름 없음)
    1
    898000
```

- COUNT
 - ▶ 데이터의 개수

```
SQLQuery1.sql - ...IAC9H\User (52))*
    SELECT COUNT (salary) FROM instructor
🎹 결과 🚹 메시지
  (열 이름 없음)
SQLQuery1.sql - ...IAC9H\User (52))*
    SELECT COUNT (salary) FROM instructor
    WHERE salary > 70000
🎹 결과 🚹 메시지
  (열 이름 없음)
```

- ▶ GROUP BY, HAVING 절
 - ▶ GROUP BY : 특정 컬럼으로 그룹 짓는 연산자
 - ▶ HAVING : 그룹 지어진 데이터에 대한 조건을 부여

```
SQLQuery1.sql - ...IAC9H\u00fcUser (52))\u00e5
SELECT dept_name FROM instructor
GROUP BY dept_name
```



SELECT dept_name FROM instructor
GROUP BY dept_name
HAVING dept_name LIKE '_i%'



- > 중첩 부질의란?
 - ▶ 다른 질의 내에 내포되는 SELECT-FROM-WHERE 표현 식
 - ▶ 부질의의 공통적인 사용은 집합 멤버쉽, 집합 비교 및 집합 수의 테스트를 수행하는 것
 - ▶ IN
 - SOME & ALL
 - EXISTS & NOT EXISTS
 - ▶ FROM 절의 부질의

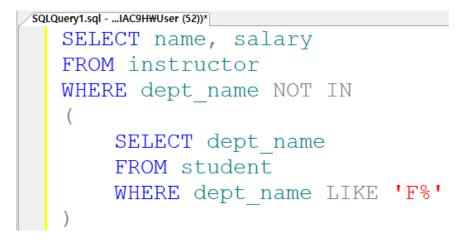
IN & NOT IN

▶ 집합 멤버쉽

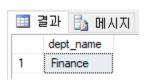
```
SQLQuery1.sql-...IAC9HWUser (52))*

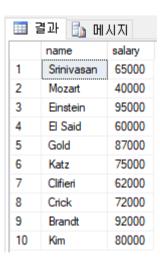
SELECT name, salary
FROM instructor
WHERE dept_name IN
(

SELECT dept_name
FROM student
WHERE dept_name LIKE 'F%'
)
```









SOME & ALL

▶ 집합 비교

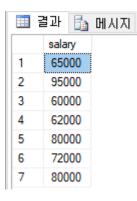
```
SQLQuery1.sql-...IAC9HWUser (52))*

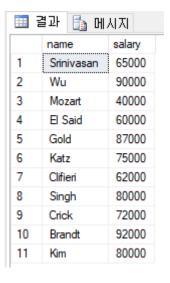
SELECT name, salary
FROM instructor
WHERE salary < SOME
(
SELECT salary
FROM instructor
WHERE name LIKE '%i%'
)
```

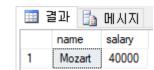
```
SQLQuery1.sql-...IAC9HWUser (52))*

SELECT name, salary
FROM instructor
WHERE salary < ALL
(

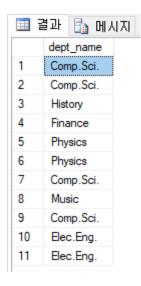
SELECT salary
FROM instructor
WHERE name LIKE '%i%'
)
```







- EXISTS & NOT EXISTS
 - ▶ 빈 테이블 검사



	⊞ 결과 🛅 메시지							
	id	name	dept_name	salary				
1	10101	Srinivasan	Comp.Sci.	65000				
2	12121	Wu	Finance	90000				
3	15151	Mozart	Music	40000				
4	22222	Einstein	Physics	95000				
5	32343	El Said	History	60000				
6	33456	Gold	Physics	87000				
7	45565	Katz	Comp.Sci.	75000				
8	58583	Clifieri	History	62000				
9	76543	Singh	Finance	80000				
10	76766	Crick	Biology	72000				
11	83821	Brandt	Comp.Sci.	92000				
12	98345	Kîm	Elec.Eng.	80000				

▶ FROM 절의 부질의

```
SELECT *
FROM (

SELECT DISTINCT dept_name
FROM student
) AS tmp
WHERE tmp.dept_name LIKE '_i%'
```



