

데이터베이스의 활용-SQL 기초

데이터와 친해지기

정렬

- 데이터를 원하는 순서로 정렬
- 사용
 - DESC : 내림차순 (Descendant)
 - 한글 : ㅎ - ㄱ
 - 영어 : z-a
 - 숫자 : 큰수 - 작은수
 - ASC : 오름차순 (Ascendant)
 - 한글 : ㄱ - ㅎ
 - 영어 : a-z
 - 숫자 : 작은수 - 큰수

```
SELECT * FROM player ORDER BY age DESC;
```

	123seq	ABCname	123age	ABCcompany	ABCcountry
1	2	날강두	39	맨유	포르투갈
2	1	메시	37	바르셀로나	아르헨티나
3	4	심교훈	36	토트넘	한국
4	5	도준혁	33	[NULL]	한국
5	3	박수호	29	바르셀로나	한국

- 정렬에 중복 데이터가 있을 때, 다음 정렬 지정 가능

```
SELECT * FROM player ORDER BY country DESC, age ASC;
```

	123seq	ABCname	123age	ABCcompany	ABCcountry
1	3	박수호	29	바르셀로나	한국
2	5	도준혁	33	[NULL]	한국
3	4	심교훈	36	토트넘	한국
4	2	날강두	39	맨유	포르투갈
5	1	메시	37	바르셀로나	아르헨티나

Quiz 사전준비

```
CREATE TABLE student (
  email VARCHAR(100) NOT NULL,
  name VARCHAR(20) NOT NULL,
  age INT NOT NULL,
  school VARCHAR(20),
  birth DATE,
  PRIMARY KEY(email)
);
```

```
/*본인 포함하여 조원들 정보 데이터 추가 (총 4개의 데이터)*/
/*e.g.*/)
INSERT INTO student (email, name, age, school, birth)
VALUES ('oomoojoo@naver.com', '도준혁', 33, '동서대학교', '1991-05-29');
```

Quiz4. 정렬

4-1. student 테이블에서 나이를 기준으로 내림차순 정렬하여 출력 (단, 같은 데이터가 있을 시, 이름을 기준으로 내림차순으로 정렬한다)

	ABC email	ABC name	123 age	ABC school	🕒 birth
1	ejoon99@korea.ac.kr	이준수	25	고려대	1999-10-10
2	gyuheon0522@naver.com	이규현	24	용인대	2000-05-22
3	wnsah0203@naver.com	성준모	22	인천대	2002-03-19
4	daisyvoo33@gmail.com	유예린	21	한국외대	2003-04-22

집계 함수

- 종류
 - COUNT
 - SUM
 - AVG
 - MIN
 - MAX

COUNT

```
SELECT COUNT(*) FROM player;
```

- COUNT(*): '모든 열 = 테이블 전체'가 조회되는 갯수

```
SELECT COUNT(name) FROM player;
```

- 해당 열(name) 기준으로 조회되는 갯수

123	COUNT(*)	▼
1	5	

123	COUNT(name)	▼
1	5	

```
SELECT COUNT(company) FROM player;
```

- 집계 함수는 집합 안에 NULL 값이 있을 경우 무시

123	COUNT(company)	▼
1	4	

SUM

```
SELECT SUM(age) FROM player;
```

- 문자나 날짜는 합계를 구할 수 없음
- 집계 함수는 집합 안에 NULL 값이 있을 경우 무시

123	SUM(age)	▼
1	174	

활용

```
SELECT SUM(age)/COUNT(age) FROM player;
```

123	SUM(age)/COUNT(age)	▼
1	34.8	

AVG

```
/*결과 동일 (SELECT SUM(age)/COUNT(age) FROM player;)*  
SELECT AVG(age) FROM player;
```

123	AVG(age)	▼
1	34.8	

MIN

```
SELECT MIN(age) FROM player;
```

	123 MIN(age) ▼
1	29

MAX

```
SELECT MAX(age) FROM player;
```

	123 MAX(age) ▼
1	39

MIN / MAX

- 문자나 날짜에도 사용 가능

```
SELECT MIN(name), MAX(name) FROM player;
```

	ABC MIN(name) ▼	ABC MAX(name) ▼
1	날강두	심교훈

Quiz5. 집계함수

5-1. student 테이블의 총 데이터 개수를 출력

	123 count(*) ▼
1	4

5-2. student 테이블의 나이 합계, 평균 나이, 가장 많은 나이, 가장 적은 나이 출력

	123 SUM(age) ▼	123 AVG(age) ▼	123 MAX(age) ▼	123 MIN(age) ▼
1	92	23	25	21

GROUP BY

- 특정 컬럼을 그룹화

```
SELECT country, COUNT(country)
FROM player
GROUP BY country
```

	ABC country ▼	123 COUNT(country) ▼
1	아르헨티나	1
2	포르투갈	1
3	한국	3

```
SELECT country, COUNT(country)
FROM player
GROUP BY country
HAVING COUNT(country) = 1;

/*실행 불가*/
/*
SELECT country, COUNT(country)
FROM player
WHERE COUNT(country) = 1
GROUP BY country;
*/
```

	ABC country ▼	123 COUNT(country) ▼
1	아르헨티나	1
2	포르투갈	1

- 내부처리 순서
 - FROM -> WHERE -> GROUP BY -> HAVING -> SELECT -> ORDER BY
- GROUP BY에서 지정한 컬럼 이외의 컬럼은 집계함수를 사용하지 않은 채 SELECT 구에 지정하는 것은 권장하지 않음 (지정하지 않은 컬럼은 어떤 값으로 나와야 할지 알기 어렵기 때문)

Quiz6. 그룹화 - GROUP BY / HAVING

6-1. student 테이블에서 나이별 학생 수를 출력 (출력 : 나이, 학생 수)

	123 age ▼	123 count(email) ▼
1	21	1
2	22	1
3	24	1
4	25	1

6-1-1. 출력된 데이터 중 동값이 있는 데이터(나이별 학생 수가 2이상)들만 출력 (출력 : 나이, 학생 수)

123 age	123 count(email)

6-2. student 테이블에서 나이가 23세 이상인 학생들 중 학교별 학생 수를 출력하라. (출력 : 학교, 학생 수)

	ABC school	123 count(email)
1	고려대	1
2	용인대	1

서브쿼리

- 서브쿼리는 SELECT 명령에 의한 데이터 질의로, 상부가 아닌 하부의 부수적인 질의를 의미
 - SELECT문 안에 SELECT문을 의미
 - 사용
 - (SELECT 명령)

```
SELECT *
FROM player
WHERE age = (SELECT MIN(age) FROM player);
```

	123 seq	ABC name	123 age	ABC company	ABC country
1	3	박수호	29	바르셀로나	한국

Quiz7. 서브쿼리

7-1. student 테이블에서 평균나이보다 많은 학생들을 출력 (출력 : 모든 컬럼)

	ABC email	ABC name	123 age	ABC school	🕒 birth
1	ejoon99@korea.ac.kr	이준수	25	고려대	1999-10-10
2	gyuheon0522@naver.com	이규현	24	용인대	2000-05-22