DATABASE

SQL(DML - SELECT)

SQL(DML - SELECT)

| SQL의 분류

- SQL문은 DBMS에 줄 수 있는 명령의 종류에 따라 3가지로 분류됨
- SQL문 대부분은 DML이며 실무에 사용하는 SQL문 또한 대부분 DML을 사용함

DDL	**데이터 정의 언어 (Data Definition Language)**	- 데이터베이스 개체(스키마 or 테이블)를 생성/삭제/변경하는 역할을 담당 - 예) CREATE, DROP, ALTER, USE 등
DML	**데이터 조작 언어 (Data Manipulation Language)**	- 테이블의 데이터를 검색/수정/삭제하는 역할을 담당 - 예) SELECT, INSERT, UPDATE, DELETE 등
DCL	**데이터 제어 언어 (Data Control Language)**	- 데이터베이스에서 실행한 변경을 확정하거나 취소하는 역할을 담당 - 예) COMMIT, ROLLBACK 등

SQL(DML - SELECT)

| 연산자

비교 연산자

- 2개 항을 비교할 때 사용

연산자	사용법	의미
>	a > 10	a는 10 초과
<	a < 10	a는 10 미만
>=	a >= 10	a는 10 이상
<=	a <= 10	a 는 10 이하
= <=>	a = 10 a <=> 10	a는 10과 같음
!= <>	a != 10 a <> 10	a는 10이 아님

논리 연산자

- 2개 항을 논리적으로 연결할 때 사용

연산자	사용 방법	의미
and	a and b a && b	a와 b가 모두 참이면 True, 아니면 False
or	a or b a || b	a와 b 중 하나라도 참이면 True, 아니면 False
not	not a !a	a가 참이면 False, a가 거짓이면 True

SQL(DML - SELECT)

| 연산자

연산자의 우선순위

- 연산자 우선순위를 외우기보다는 () 괄호를 사용하자

우선순위	연산자
1	INTERVAL
2	BINARY, COLLATE
3	-(단항 감산,) ~(단항 비트 반전)
4	^
5	*, /, DIV, %, MOD
6	-, +
7	<<, >>
8	&
9	|
10	=, <=>, >=, >, <, <=, <>, !=, IS, LIKE, REGEXP, IN
11	BETWEEN, CASE, WHEN, THEN, ELSE
12	NOT
13	&&, AND
14	XOR
15	||, OR
16	=(대입 등호), :=

SQL(DML – SELECT)

| SELECT

테이블 내의 전체 데이터 조회

- select: 데이터 조회를 위한 SQL명령어
- *: '전체'라는 의미를 가짐

select * from 테이블명;

```
select * from city;
```

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabol	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238

다중 테이블 조회

```
select * from city, country;
```

select * from 테이블명, 테이블명..;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ID	Name	CountryCode	District	Population	Code	Name	Continent	Region	SurfaceArea	Indep
1	Kabul	AFG	Kabol	1780000	ZWE	Zimbabwe	Africa	Eastern Africa	390757.00	1980
1	Kabul	AFG	Kabol	1780000	ZMB	Zambia	Africa	Eastern Africa	752618.00	1964
1	Kabul	AFG	Kabol	1780000	ZAF	South Africa	Africa	Southern Africa	1221037.00	1910
1	Kabul	AFG	Kabol	1780000	YUG	Yugoslavia	Europe	Southern Europe	102173.00	1918
1	Kabul	AFG	Kabol	1780000	YEM	Yemen	Asia	Middle East	527968.00	1918
1	Kabul	AFG	Kabol	1780000	WSM	Samoa	Oceania	Polynesia	2831.00	1962
1	Kabul	AFG	Kabol	1780000	WLF	Wallis and Futuna	Oceania	Polynesia	200.00	NULL
1	Kabul	AFG	Kabol	1780000	VUT	Vanuatu	Oceania	Melanesia	12189.00	1980
1	Kabul	AFG	Kabol	1780000	VNM	Vietnam	Asia	Southeast Asia	331689.00	1945
1	Kabul	AFG	Kabol	1780000	VIR	Virgin Islands, U.S.	North America	Caribbean	347.00	NULL
1	Kabul	AFG	Kabol	1780000	VGB	Virgin Islands, British	North America	Caribbean	151.00	NULL
1	Kabul	AFG	Kabol	1780000	VEN	Venezuela	South America	South America	912050.00	1811
1	Kabul	AFG	Kabol	1780000	VCT	Saint Vincent and ...	North America	Caribbean	388.00	1979
1	Kabul	AFG	Kabol	1780000	VAT	Holy See (Vatican ...	Europe	Southern Europe	0.40	1929
1	Kabul	AFG	Kabol	1780000	UZB	Uzbekistan	Asia	Southern and C...	447400.00	1991

SQL(DML - SELECT)

| SELECT

특정 Column(열) 데이터 조회

select column명, colum명.. from 테이블명;

```
select countrycode, district from city;
```

	countrycode	district
▶	AFG	Kabol
	AFG	Qandahar
	AFG	Herat
	AFG	Balkh
	NLD	Noord-Holland
	NLD	Zuid-Holland
	NLD	Zuid-Holland
	NLD	Utrecht
	NLD	Noord-Brabant
	NLD	Noord-Brabant

select 조회 시 Column(열)의 title 설정

select column명 as title명 from 테이블명;

```
select name as 나라이름, district as 구역
from city;
```

	나라이름	구역
▶	Kabul	Kabol
	Qandahar	Qandahar
	Herat	Herat
	Mazar-e-Sharif	Balkh
	Amsterdam	Noord-Holland
	Rotterdam	Zuid-Holland
	Haag	Zuid-Holland
	Utrecht	Utrecht
	Eindhoven	Noord-Brabant

SQL(DML - SELECT)

| SELECT: 조건

특정 조건에 해당하는 데이터 조회

- where: 조건을 지정하는 명령어
- where절이 추가되면 해당 조건에 해당하는 데이터만 조회

select * from 테이블명 where 조건;

```
select * from city where countrycode = "KOR";
```

Result Grid | Filter Rows: | Edit:

ID	Name	CountryCode	District	Population
2331	Seoul	KOR	Seoul	9981619
2332	Pusan	KOR	Pusan	3804522
2333	Inchon	KOR	Inchon	2559424
2334	Taegu	KOR	Taegu	2548568
2335	Taejon	KOR	Taejon	1425835
2336	Kwangju	KOR	Kwangju	1368341
2337	Ulsan	KOR	Kyongsangnam	1084891
2338	Songnam	KOR	Kyonggi	869094
2339	Puchon	KOR	Kyonggi	779412
2340	Suwon	KOR	Kyonggi	755550

다중 조건 조회

- 논리 연산자를 통해 여러 조건 지정 가능
- and(&&), or(||), not(!) 등

```
select * from city
where countrycode = "KOR" and district = "kyonggi";
```

ID	Name	CountryCode	District	Population
2338	Songnam	KOR	Kyonggi	869094
2339	Puchon	KOR	Kyonggi	779412
2340	Suwon	KOR	Kyonggi	755550
2341	Anyang	KOR	Kyonggi	591106
2344	Koyang	KOR	Kyonggi	518282
2345	Ansan	KOR	Kyonggi	510314
2349	Kwangmyong	KOR	Kyonggi	350914
2353	Pyongtaek	KOR	Kyonggi	312927
2355	Uijongbu	KOR	Kyonggi	276111
2362	Yong-in	KOR	Kyonggi	242643

SQL(DML - SELECT)

| SELECT: 제한

DISTINCT

- 선택한 행에서 중복된 값을 제거하고 싶을 때 사용하는 명령어
- 선택한 조건에 해당하는 값이 여러 개가 있다면 가장 처음에 나타난 행 값을 가져옴

select distinct column명 from 테이블명;

```
select distinct district from city where countrycode = 'kor';
```

LIMIT

- 출력 개수를 제한하고 싶을 때 사용하는 명령어

select distinct column명 from 테이블명 LIMIT 시작 행, 개수;

```
select district, name from city where countrycode = 'kor' LIMIT 3;
select district, name from city where countrycode = 'kor' LIMIT 3, 5;
```

SQL(DML - SELECT)

| SELECT: 정렬

ORDER BY

- order by: 검색결과의 정렬에 사용되는 명령어
- 열의 순서는 select 구문 뒤의 명령 순서에 따라 조회된다.
- 따로 명시적으로 명령해 준 것이 없다면 데이터를 넣은 순서대로 표시된다.
- ASC: ascend(오르다) 라는 의미로 오름차순 정렬을 의미 -> 기본 정렬방식
- DESC: descend(내리다) 라는 의미로 내림차순 정렬을 의미

select ~ order by 열1[, 열2, ...];

```
select district, name from city order by district;

select * from city order by district asc, population desc;
```

SQL(DML - SELECT)

| SELECT: 조건 표현

BETWEEN..AND

- 조건에서 사이 값을 지정하고자 할 때 사용
- 반대의 경우에는 NOT BETWEEN을 사용

```
select * from city where population between 100 and 2000;
```

IN

- 해당되는 여러 조건을 한번에 작성할 때 사용
- 여러 개의 값에 대해 동등 비교 연산을 수행하며, 일치하는 것이 하나라도 있을 경우 결과로 가져옴

```
select * from city where countrycode in('kor', 'nld');
```

SQL(DML - SELECT)

| SELECT: 조건 표현

IS NULL / IS NOT NULL

- 해당 조건에서 특정 속성의 값이 NULL인지를 비교하는 키워드
- NULL값을 찾을 때: IS NULL
- NULL값이 아닌 값을 찾을 때: IS NOT NULL
- NULL 값에 비교문을 작성하면 결과는 FALSE로 받음

```
select * from country where indepYear is null;
```

LIKE

- 해당하는 문자가 포함된 문자열을 검색할 때 사용
- %A: A로 끝나는 모든 문자 // A%: A로 시작하는 모든 문자
- _: 문자의 개수를 지정
- LIKE 연산자는 인덱스를 이용해 처리할 수 있지만, 와일드카드 문자(%, _)가 검색어의 앞쪽에 있다면 인덱스 레인지 스캔을 사용할 수 없음
- %나 _ 문자 자체를 찾아야 한다면 이스케이프 문자 설정인 "LIKE 'a₩%' ESCAPE '₩'"를 사용해야 함

```
select * from city where countrycode = 'kor' and name like 'k%';

select * from city where countrycode = 'kor' and name like 'k___';
```

SQL(DML – SELECT)

| SQL – SELECT 실습 문제

practiceDB에 아래의 사진과 같이 buytbl과 usertbl을 생성하고 결과를 출력하세요

1. buytbl에서 구매양(amount)이 5개 이상인 행을 출력

2. buytbl에서 groupName이‘서적’이거나 ‘전자’인 행을 출력

3. buytbl에서 groupName이 NULL인 userID와 prodName데이터만 출력

4. buytbl에서 구매양(amount)이 3 이상이고 10이하인 행을 price를 기준으로 오름차순 출력

5. usertbl에서 userID가 k로 시작하는 행을 출력

6. buytbl에서 userID를 중복제거하여 내림차순 정렬

7. usertbl에서 1970년대 이후에 태어난 행의 Name, birthyear, addr 행을 출력

num	userID	prodName	groupName	price	amount
1	KBS	운동화	NULL	30	2
2	KBS	노트북	전자	1000	1
3	JYP	모니터	전자	200	1
4	BBK	모니터	전자	200	5
5	KBS	청바지	의류	50	3
6	BBK	메모리	전자	80	10
7	SSK	책	서적	15	5
8	EJW	책	서적	15	2
9	EJW	청바지	의류	50	1
10	BBK	운동화	NULL	30	2
11	EJW	책	서적	15	1
12	BBK	운동화	NULL	30	2
NULL	NULL	NULL	NULL	NULL	NULL

userID	Name	birthYear	addr	mobile1	mobile2	height	mDate
BBK	바비킴	1973	서울	010	0000000	176	2013-05-05
EJW	은지원	1972	경북	011	8888888	174	2014-03-03
JKW	조관우	1965	경기	018	9999999	172	2010-10-10
JYP	조용필	1950	경기	011	4444444	166	2009-04-04
KBS	김범수	1979	경남	011	2222222	173	2012-04-04
KKH	김경호	1971	전남	019	3333333	177	2007-07-07
LJB	임재범	1963	서울	016	6666666	182	2009-09-09
LSG	이승기	1987	서울	011	1111111	182	2008-08-08
SSK	성시경	1979	서울	NULL	NULL	186	2013-12-12
YJS	윤종신	1969	경남	NULL	NULL	170	2005-05-05
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SQL(DML - SELECT)

| 집계함수

집계함수

- 함수: 데이터에 대해 조작하거나 계산을 수행하기위한 도구
- 집계함수: 복수 행(이나 행의 값)에 대해 집계를 수행하는 함수(기능)

연산자	의미
AVG()	평균을 구한다
MIN()	최소값을 구한다
MAX()	최대값을 구한다
COUNT()	행의 개수를 센다
COUNT(DISTINCT)	중복을 1개만 인정하고 행의 개수를 센다
SUM()	합계를 구한다
STDEV()	표준편차를 구한다
VAR()	분산을 구한다

```
select max(district)
from city where countrycode = 'kor';
```

max(district)
Taejon

```
select count(name) from country;
```

count(name)
239

```
select count(distinct district) from city
where countrycode = 'jpn';
```

count(distinct district)
47

SQL(DML – SELECT)

| SELECT: 그룹화

GROUP BY

- 데이터를 그룹화 할 때 사용하는 명령어
- 그룹을 나눌 때 KEY가 되는 열을 지정하여 GROUP BY 뒤의 열의 합계를 구해줌

SELECT ~ FROM ~ GROUP BY 열이름;

```
select count(*) from city where countrycode = 'kor';
select district, count(*) from city where countrycode = 'kor' group by district;
```

HAVING

- 집계결과에 대한 조건을 지정할 때 사용하는 명령어

SELECT ~ FROM ~ GROUP BY ~ HAVING 조건;

불가능

```
select district, count(*) from city
where countrycode = 'kor' and count(*) = 4
group by district;
```

가능

```
select district, count(*) from city
where countrycode = 'kor'
group by district having count(*) = 6;
```

SQL(DML – SELECT)

| SELECT: 합계

ROLLUP

- 소합계와 총합계를 만드는 명령 쿼리문
- GROUP BY와 함께 사용됨

SELECT ~ FROM ~ GROUP BY ~ WITH ROLLUP

GROUP BY로 그룹화 한 그룹의 합계와 총 합계를 알고 싶을 경우

```
SELECT district, sum(population) from city
where countrycode = 'kor'
group by district with rollup;
```

district	sum(population)
Cheju	258511
Chollabuk	1510489
Chollanam	910194
Chungchongbuk	873652
Chungchongnam	1020120
Inchon	2559424
Kang-won	787863
Kwangju	1368341
Kyonggi	6489573
Kyongsangbuk	2063477
Kyongsangnam	3397705
Pusan	3804522
Seoul	9981619
Taegu	2548568
Taejon	1425835
NULL	38999893

정렬이 이루어지며 마지막에 모든 결과의 합계가 이루어짐

GROUP BY로 그룹화 한 그룹의 소합계와 총 합계를 알고 싶을 경우

```
SELECT name, district, sum(population) from city
where countrycode = 'kor'
group by district, name with rollup;
```

name	district	sum(population)
Cheju	Cheju	258511
NULL	Cheju	258511
Chong-up	Chollabuk	139111
Chonju	Chollabuk	563153
Iksan	Chollabuk	322685
Kimje	Chollabuk	115427
Kunsan	Chollabuk	266569
Namwon	Chollabuk	103544
NULL	Chollabuk	1510489
Kwang-yang	Chollanam	122052
Mokpo	Chollanam	247452
Naju	Chollanam	107831
Sunchon	Chollanam	249263
Yosu	Chollanam	183596
NULL	Chollanam	910194
Chechon	Chungchongbuk	137070

나눈 그룹에 해당하는 부분과 해당하지 않는 부분에 대한 합계가 동시에 이루어짐

SQL(DML - SELECT)

| SQL – SELECT 실습 문제2

buytbl과 usertbl, world db를 사용하여 아래의 결과를 출력하세요

1. 한국에서 (countryCode가 kor) 가장 인구수가 높은 도시와 인구수가 낮은 도시의 인구수를 출력

2. countrylanguage 테이블에서 세계의 모든 나라 언어 개수를 출력

3. city 테이블에서 각 나라의 도시 개수를 나라코드와 함께 내림차순 출력

4. city 테이블에서 한국의 도시 중, district 별 도시 개수가 5개 이상인 district만 출력

5. usertbl에서 각 addr 별 키의 평균 중 하위 3개의 addr과 평균을 출력

6. usertbl에서 mobile1과 mobile2의 값이 존재하는 행만 출력

7. buytbl에서 물건을 구매한 유저별 유저아이디와 구매액 합계를 출력하는데, 총 합계도 함께 출력

8. buytbl에서 prodName별 amount * price의 합계와 총 합을 구해서 prodName과 함께 출력

수고하셨습니다