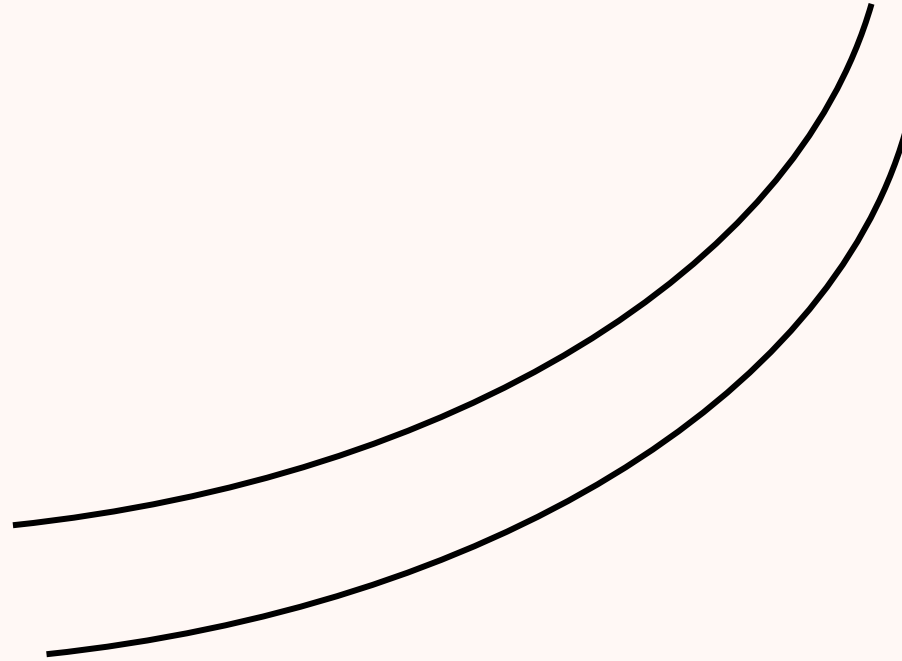


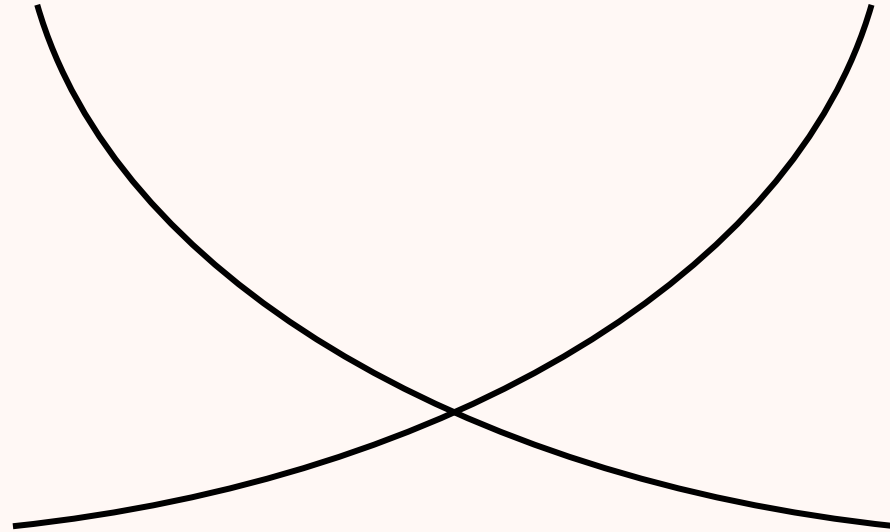
금융데이터로 실습하며 익히는 SQL입문

난이도
difficulty

중요도
importance



난이도
difficulty



중요도
importance

0. 데이터베이스? SQL?

SQL이란?

- Structured Query Language, 구조화 질의어, S-Q-L
- 관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어
- 관계형 데이터베이스 관리 시스템에서 자료의 검색과 관리, 데이터베이스 스키마 생성과 수정, 데이터베이스 객체 접근 조정 관리를 위해 고안
- 많은 수의 데이터베이스 관련 프로그램들이 SQL을 표준으로 채택하고 있다.

출처: <https://ko.wikipedia.org/wiki/SQL>

0. 데이터베이스? SQL?



0. 데이터베이스? SQL?



0. 데이터베이스? SQL?

ID	CpnyName	Ticker	Region	IndCode	MktDate_	CurrCode	Close_
330664	HYBE CO LTD	352820	KR	2163	2020-11-24	KRW	181000
330664	HYBE CO LTD	352820	KR	2163	2020-11-25	KRW	174000
330664	HYBE CO LTD	352820	KR	2163	2020-11-26	KRW	176000
330664	HYBE CO LTD	352820	KR	2163	2020-11-27	KRW	178000

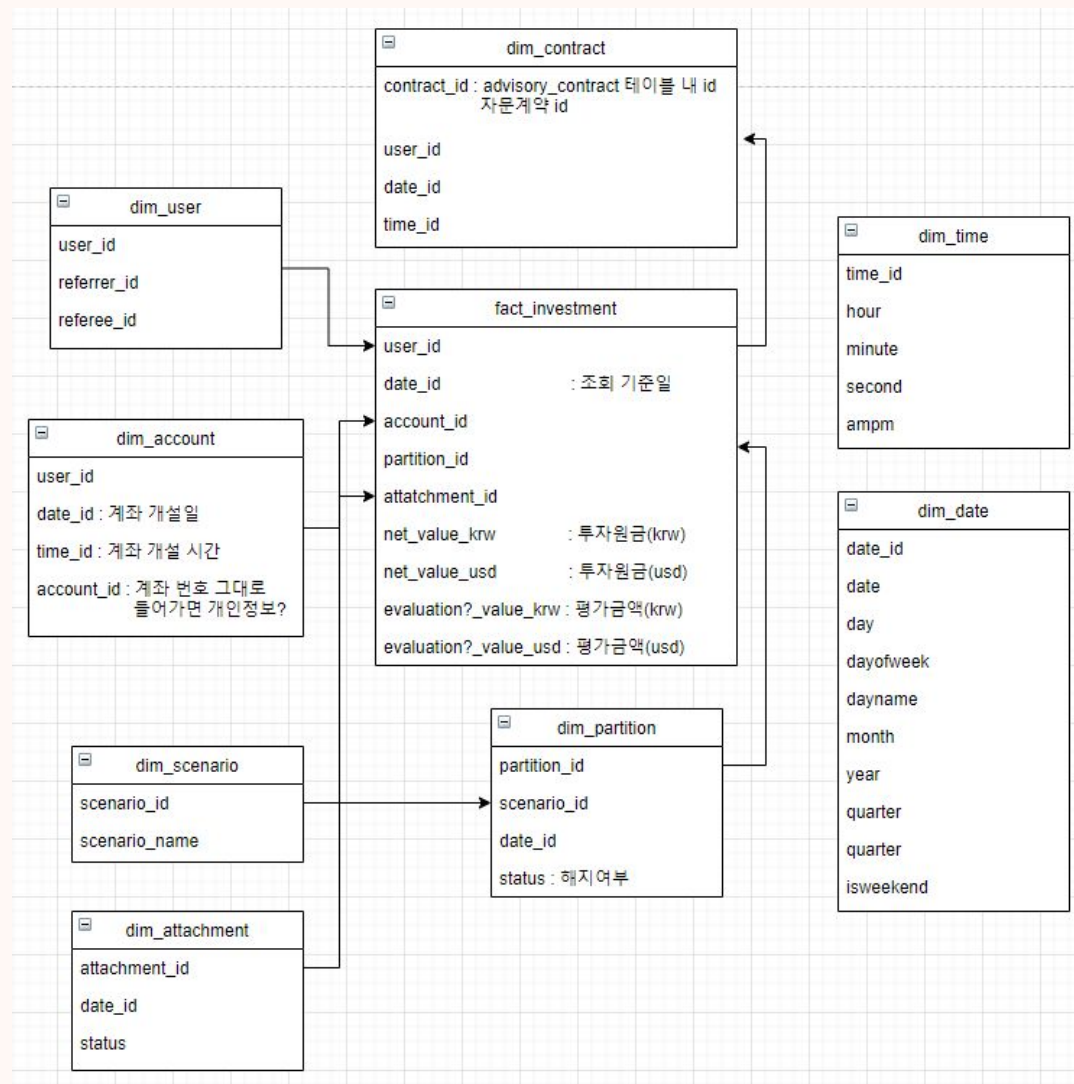
Stock_Info

ID	CpnyName	Ticker	Region	IndCode	Mkt_Capitalization_USD
330664	HYBE CO LTD	352820	KR	2163	8977
39067	KT & G CORPORATION	33780	KR	3029	8677
297191	NETMARBLE CORP	251270	KR	5144	8552
25183	SAMSUNG FIRE & MARINE INSURANC...	810	KR	3958	8306
25200	KOREA ZINC INC	10120	KR	7102	7605

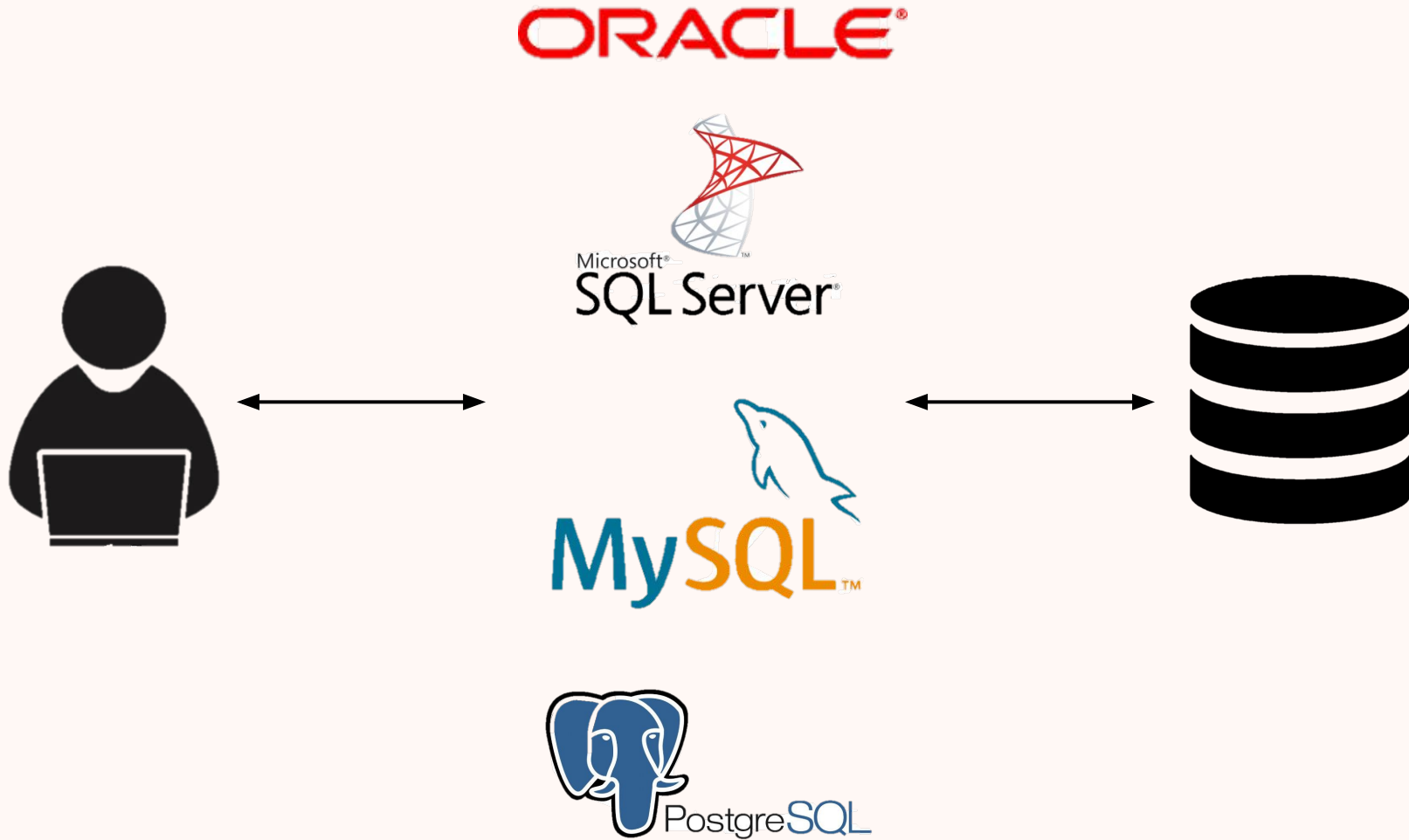
Stock_Price

ID	MktDate_	CurrCode	Close_
330664	2020-11-09	KRW	163000
330664	2020-11-10	KRW	168000
330664	2020-11-11	KRW	164000
330664	2020-11-12	KRW	157500
330664	2020-11-13	KRW	160000
330664	2020-11-16	KRW	171500

0. 데이터베이스? SQL?



0. 데이터베이스? SQL?



0. 데이터베이스 용어



통합문서

시트

행 (1,2,3)

열 (A,B,C)



SCHEMA

TABLE

ROW

COLUMN

0. 데이터베이스 용어

TABLE

COLUMN

InfoCode	CpnyName	Ticker	Region	IndCode	Market Capitalization
72990	APPLE INC.	AAPL	US	15	2,466,488
39988	MICROSOFT CORPORATION	MSFT	US	22	2,223,443
60734	ALPHABET INC	GOOGL	US	22	1,881,081
45293	AMAZON.COM, INC.	AMZN	US	67	1,755,390
261387	FACEBOOK INCORPORATION	FB	US	22	1,070,178

ROW

1. 데이터 불러오기(SELECT)

1. 데이터 불러오기(SELECT)

Case 1. 테이블의 특정 컬럼을 조회

```
SELECT 컬럼이름1, 컬럼이름2  
FROM 테이블이름
```

Case 2. 테이블의 모든 컬럼을 조회

```
SELECT *  
FROM 테이블이름
```

1. 데이터 불러오기(SELECT)

Case 3. 중복 제거된 컬럼을 조회

```
SELECT DISTINCT 컬럼이름1, 컬럼이름2  
FROM 테이블이름
```

Case 4. 컬럼에 별명을 붙여서 조회

```
SELECT 컬럼이름1 (AS) 별명1, 컬럼이름2 (AS) 별명2  
FROM 테이블이름
```

1. 데이터 불러오기(SELECT)

Case 5. 결과 행 개수 제한하여 조회

```
SELECT 컬럼이름1 , 컬럼이름2  
FROM 테이블이름  
LIMIT N
```

1. 데이터 불러오기(SELECT)

과제 1. StockETF_Price의 모든 테이블을 조회해보세요.

□ 모든 열 이름을 나열하는 방법과 문자 * 사용하는 두가지 방법 사용

과제 2. ETF_Info의 GEO_FOCUS 컬럼을 중복 제거하여 조회해보세요.

과제 3. StockETF_Price에서 30개의 행만 조회해보세요.

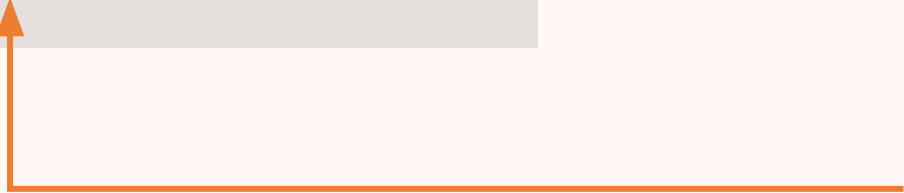
□ LIMIT 활용

2. 데이터 필터링하기(**WHERE**)

2. 데이터 필터링하기(WHERE)

Syntax

```
SELECT 컬럼이름1, 컬럼이름2  
FROM 테이블이름  
WHERE 조건1;
```



비교연산자	설명
=	같음
<> , !=	같지 않음
<	보다 작음
<=	보다 작거나 같음
>	보다 큼
>=	보다 크거나 같음
BETWEEN	지정된 두 값 사이에 있음
IS NULL	Null 값임

2. 데이터 필터링하기(WHERE)

Case 1. 숫자형 데이터 조건 걸기

```
SELECT *  
FROM 테이블이름  
WHERE 비교컬럼이름 = 숫자
```

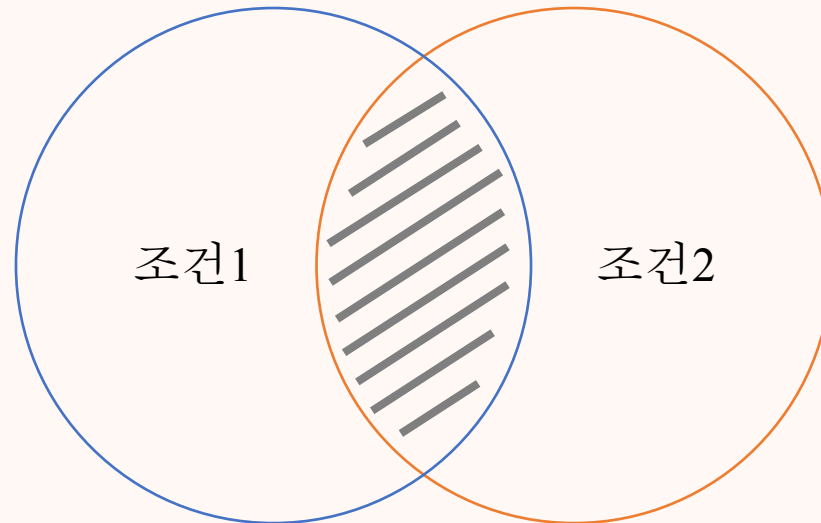
Case 2. 문자형 데이터 조건 걸기

```
SELECT *  
FROM 테이블이름  
WHERE 비교컬럼이름 = '문자'
```

2. 데이터 필터링하기(WHERE)

Case 3. 여러 조건을 모두 만족하는 데이터 조회하기

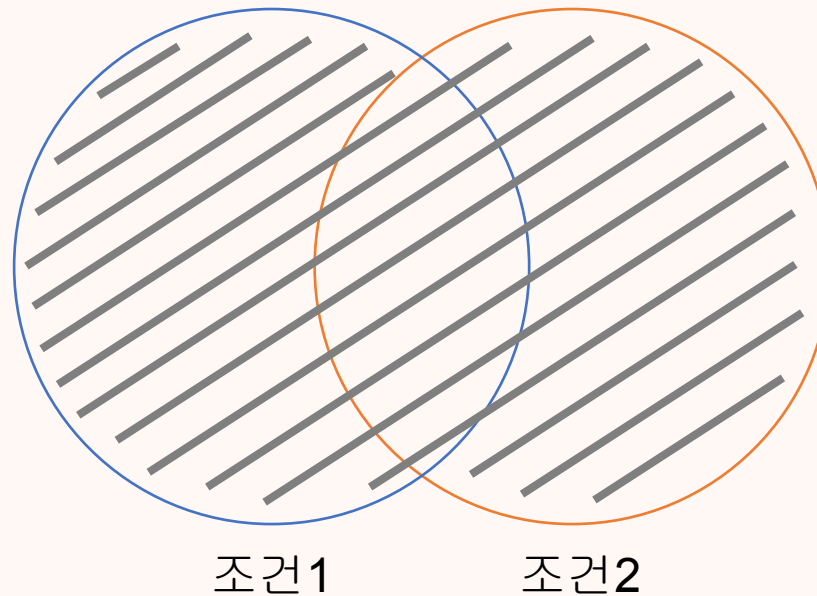
```
SELECT *  
FROM 테이블이름  
WHERE 조건1 AND 조건2
```



2. 데이터 필터링하기(WHERE)

Case 4. 여러 조건을 하나 이상 만족하는 데이터 조회하기(1)

```
SELECT *  
FROM 테이블이름  
WHERE 조건1 OR 조건2
```



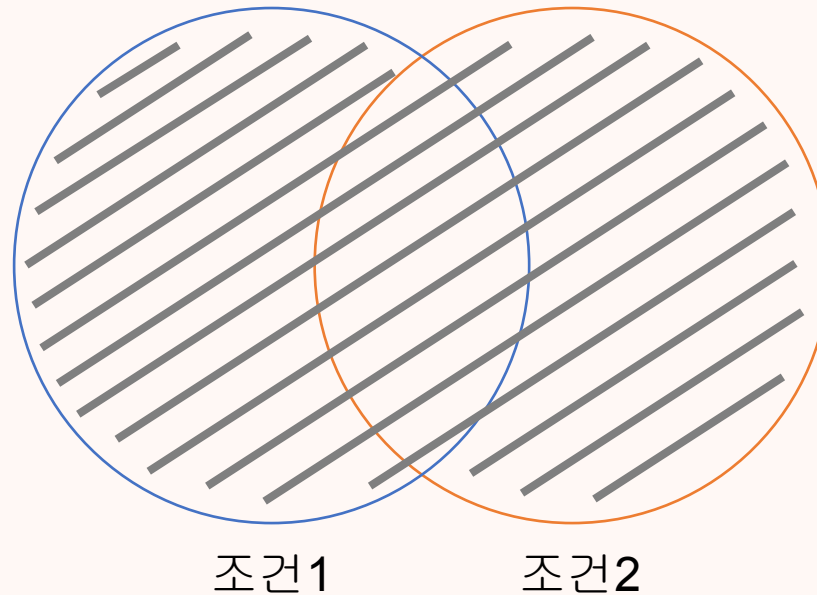
2. 데이터 필터링하기(WHERE)

Case 4. 여러 조건을 하나 이상 만족하는 데이터 조회하기(2)

```
SELECT *
```

```
FROM 테이블이름
```

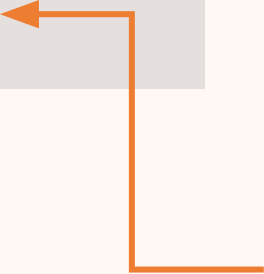
```
WHERE 비교컬럼이름 (NOT) IN (조건1, 조건2)
```



2. 데이터 필터링하기(WHERE)

Case 5. 특정한 패턴을 가지는 값을 조회할 때

```
SELECT *  
FROM 테이블이름  
WHERE 컬럼이름 LIKE  
패턴
```



LIKE Operator	Description
'a%'	'a'로 시작
'%a'	'a'로 끝나는
'%sl%'	'sl'이 포함된
'_m%'	두번째 글자가 'm'인
'a_%_ %'	'a'로 시작하고 글자수가 3개 이상
't%a'	't'로 시작하고 'a'로 끝나는

2. 데이터 필터링하기(WHERE)

과제 1. Financial_Info 테이블에서

Total_Assets의 값이 10,000,000 이상인 값만 조회해보세요.

과제 2. Financial_Info 테이블에서 Income과 관련된 항목 중

값이 5,000 ~ 10,000 사이인 데이터를 조회해보세요.

3. 데이터 정렬하기(**ORDER BY**)

3. 데이터 정렬하기(ORDER BY)

Case 1. 오름차순으로 정렬하여 조회

```
SELECT *  
FROM 테이블이름  
ORDER BY 컬럼이름1 (ASC)
```

** Ascending 오름차순*

Case 2. 내림차순으로 정렬하여 조회

```
SELECT *  
FROM 테이블이름  
ORDER BY 컬럼이름1 DESC
```

** Descending 내림차순*

3. 데이터 정렬하기(ORDER BY) ▲

과제 1. ETF_Info 테이블에서 Asset_Type과 Name을 기준으로 정렬한 값을 조회해보세요

□ 컬럼이름을 사용하는 방법과 순서를 입력하는 방법을 모두 활용해보세요

과제 2. Financial_Info 테이블에서 Retained_Earnings 값을 기준으로 상위 10개의 데이터만 조회해보세요.

4. 여러가지 함수

4. 여러가지 함수(CASE WHEN)

Syntax.

```
SELECT CASE WHEN 조건1 THEN 결과값1  
        WHEN 조건2 THEN 결과값2  
        ELSE 결과값3  
        END  
FROM 테이블이름
```

4. 여러가지 함수

Syntax.

```
SELECT 함수(컬럼이름1), 함수(컬럼이름2)  
FROM 테이블이름
```

4. 여러가지 함수(집계 & 숫자)

집계함수	설명
SUM()	합계
COUNT()	행 개수
MIN()	최소값
MAX()	최대값
AVG()	평균

숫자함수	설명
TRUNCATE(,n)	소수점 n자리까지 표시
ROUND(, n)	소수점 자리 지정하여 반올림
CEIL()	올림
FLOOR()	내림
ABS()	절대값으로 변환
SIGN()	양수면 1, 음수면 -1, 0이면 0

4. 여러가지 함수(문자)

문자함수	설명
LENGTH()	문자열 길이 반환
TRIM()	앞, 뒤 공백 제거
UPPER(), LOWER()	모두 대문자 / 소문자로
LEFT(, n)	왼쪽에서 n번째까지
RIGHT(, n)	오른쪽에서 n번째까지
REPLACE()	특정 패턴 찾아서 변경
LPAD(), RPAD()	특정 글자 반복
SUBSTRING()	특정 위치에서 자르기
CONCAT()	문자 붙이기
CONCAT_WS()	문자 사이에 값 넣기

4. 여러가지 함수(날짜 & 시간)

날짜함수	설명
CURDATE()	현재 날짜 반환
CURTIME()	현재 시간 반환
NOW()	현재 날짜&시간 반환
YEAR()	연도 반환
MONTH()	월 반환
DAY()	일 반환
LAST_DAY()	해당 월의 마지막 일
WEEKDAY()	요일 값 반환(월요일 : 0)
DAYNAME()	요일 이름 반환

날짜함수	설명
ADDDATE()	시간/날짜 더하기
SUBDATE()	시간/날짜 빼기
DATEDIFF()	두 시간/날짜 일 차이
TIMEDIFF()	두 시간/날짜 시간 차이

DATE_FORMAT()	설명
%Y	4자리 년도
%y	2자리 년도
%M	긴 월(영문)
%b	짧은 월(영문)
%W	긴 요일 이름(영문)
%a	짧은 요일 이름(영문)
%i	분
%T	hh:mm:ss
%m	숫자 월 (두자리)
%c	숫자 월(한자리는 한자리)
%d	일자 (두자리)
%e	일자(한자리는 한자리)
%I	시간 (12시간)
%H	시간(24시간)
%r	hh:mm:ss AM,PM
%S	초

5. 데이터 그룹화(GROUP BY)

5. 데이터 그룹화(GROUP BY)

Syntax. 특정 컬럼을 기준으로 그룹화

```
SELECT 그룹화_기준_컬럼이름1, 그룹화_기준_컬럼이름2, 집계함수  
FROM 테이블이름  
GROUP BY 그룹화_기준_컬럼이름1, 그룹화_기준_컬럼이름2
```

5. 데이터 그룹화(HAVING)

Syntax. 그룹화 & 요약한 컬럼을 기준으로 조건 걸기

```
SELECT 그룹화_기준_컬럼이름1, 그룹화_기준_컬럼이름2, 집계함수  
FROM 테이블이름  
GROUP BY 그룹화_기준_컬럼이름1, 그룹화_기준_컬럼이름2  
HAVING 집계함수 조건
```

5. 데이터 그룹화

과제 1. Financial_Info 테이블에서 각 연도별 Operating_Income의 최소값, 최대값, 평균값을 구해보세요.

과제 2. Stock_Info 테이블에서 각 IndCode별 시가총액 평균을 구하고 시가총액 평균값이 100,000 이상인 데이터만 조회해보세요.

6. 테이블 분할 & 분석

6. 테이블 분할 & 분석(순위함수)

Syntax. 순위함수를 이용하여 순위 매기기

`SELECT` 컬럼이름, 순위함수() `OVER (ORDER BY 컬럼이름)`
`FROM` 테이블이름

순위함수	Description
RANK()	중복 순위 적용, 다음 순위가 중복 순위 적용 후 순위 Ex) 1, 2, 2, 4
DENSE_RANK()	중복 순위 적용, 다음 순위는 중복 순위 적용 +1 Ex) 1, 2, 2, 3

6. 테이블 분할 & 분석(이동함수)

Syntax. 이전 행, 다음 행 가져오기

SELECT 컬럼이름, 이동함수(컬럼이름, offset) OVER (ORDER BY 컬럼이름)
FROM 테이블이름

이동함수	Description
LAG()	이전 행 가져오기
LEAD()	다음 행 가져오기

6. 테이블 분할 & 분석(PARTITION BY)

Syntax.

```
SELECT 컬럼이름, 함수() OVER (PARTITION BY 컬럼이름 [ORDER BY 컬럼이름])  
FROM 테이블이름
```


6. 테이블 분할 & 분석(FRAME절)

Syntax.

SELECT 컬럼이름, 함수() OVER (ORDER BY 컬럼이름

ROWS BETWEEN [FRAME절] AND [FRAME절])

FROM 테이블이름



Frame 절 명령어	설명
CURRENT ROW	현재 로우
n PRECEDING	n번째 뒤의 로우
n FOLLOWING	n번째 앞의 로우
UNBOUNDED PRECEDING	가장 첫번째 로우
UNBOUNDED FOLLOWING	가장 마지막 로우

6. 테이블 분할 & 분석(PARTITION BY)

과제 1. Financial_Info 테이블에서 각 연도별
총자산 순위를 구해보세요.(내림차순)

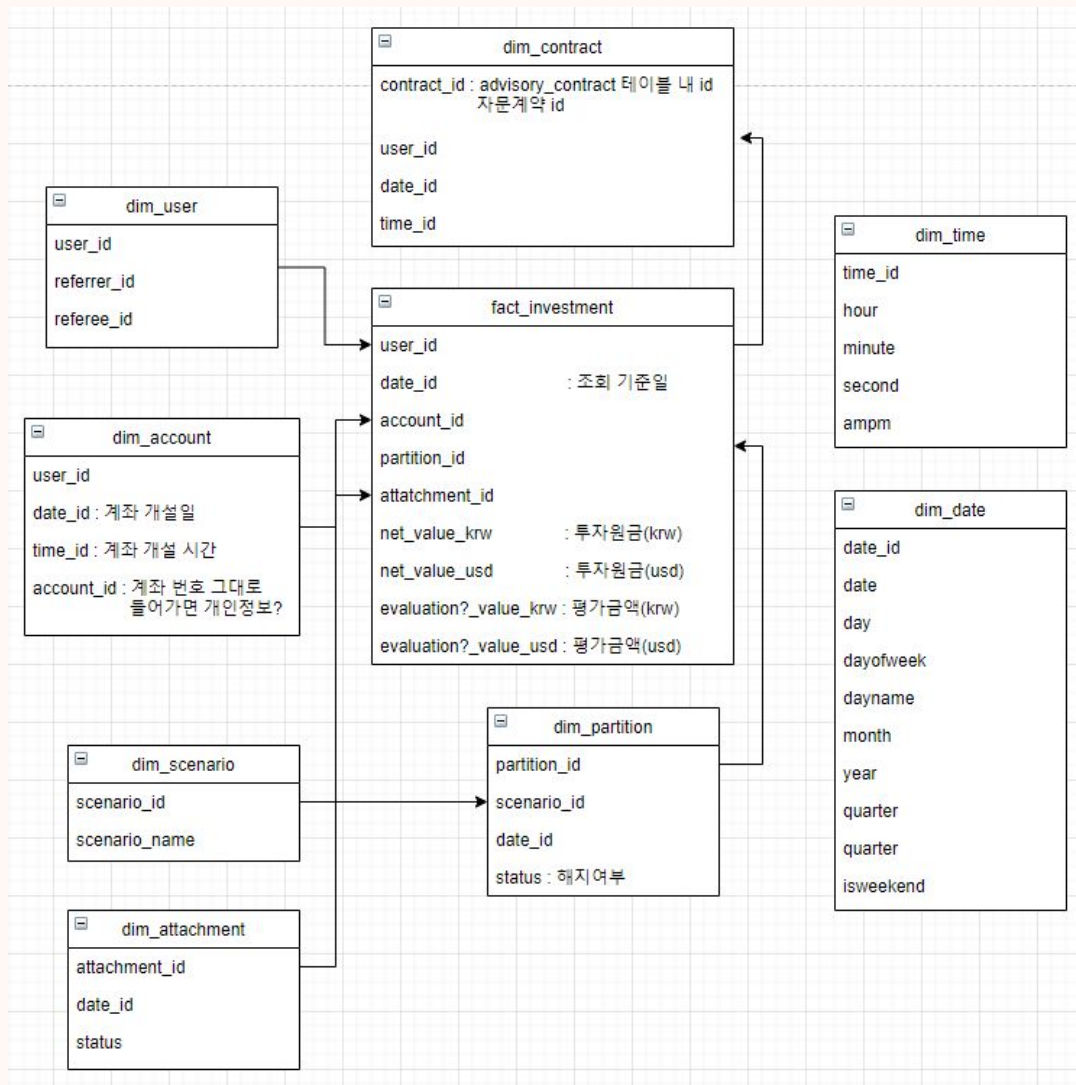
과제 2. StockETF_Price 테이블에서 삼성전자의
5일 /20일 이동평균 값을 구해보세요

과제 3. Financial_Info 테이블에서 각 기업별로 전년도 대비
Pretax_Income 증감률을 확인해보세요

$$\square \text{증감률} = (\text{이번 금액} - \text{이전 금액}) / \text{abs}(\text{이전 금액})$$

7. 테이블 합치기

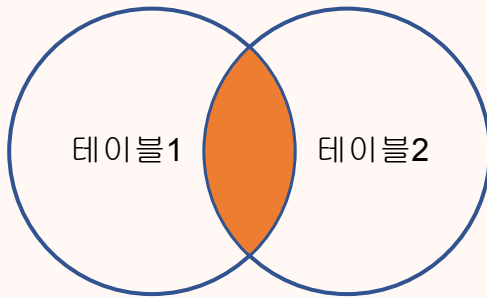
7. 테이블 합치기(JOIN)



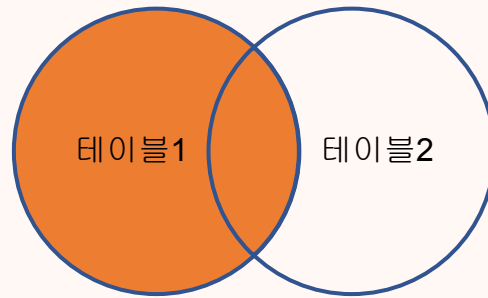
7. 테이블 합치기(JOIN)

Syntax.

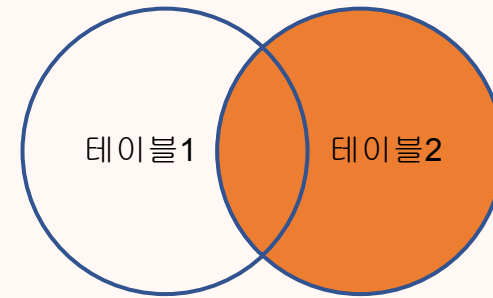
```
SELECT 컬럼이름1, 컬럼이름2  
FROM 테이블이름1 JOIN 테이블이름2  
ON 테이블이름1.컬럼A = 테이블이름2.컬럼A
```



INNER JOIN



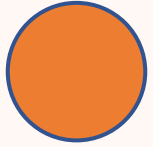
LEFT JOIN



RIGHT JOIN

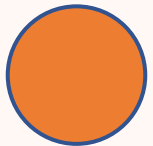
7. 테이블 합치기(JOIN)

테이블1



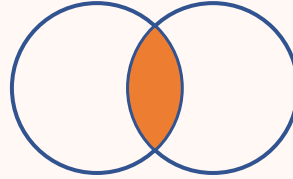
1		
2		

테이블2



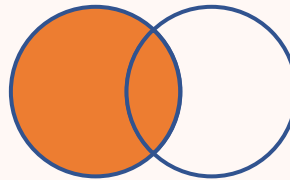
1		
4		
6		

INNER JOIN



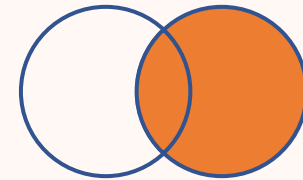
1				
---	--	--	--	--

LEFT JOIN



1				
2				

RIGHT JOIN



1				
4				
6				

7. 테이블 합치기(UNION)

테이블1

1		
2		

테이블2

1		
4		
6		

UNION ALL

1		
2		
1		
4		
6		

UNION

1		
2		
4		
6		

7. 테이블 합치기(JOIN & UNION)

과제 1. Financial_Info 테이블과 JOIN을 활용하여
산업 별 평균 세전 이익을 구해보세요.

과제 2. ETF_Info 테이블과 JOIN을 활용하여
Asset_Type 별로 종가의 평균, 최대값, 최소값을 구해보세요.

8. 서브쿼리 & 임시 테이블

8. 서브쿼리 & 임시 테이블(서브쿼리) ▾

쿼리 결과 값의 종류

1. 단일 값
2. 단일 열 테이블
3. 다중 열 테이블

8. 서브쿼리

Case 1. 다중 열 테이블 서브쿼리

```
SELECT 서브쿼리이름.칼럼1, 서브쿼리이름.컬럼2  
FROM (서브쿼리) 서브쿼리이름
```

Case 2. 단일 열 or 단일 값 서브쿼리

```
SELECT 칼럼1, 칼럼2  
FROM 테이블이름  
WHERE 칼럼1 in (서브쿼리)
```

8. 임시테이블(VIEW & WITH)

Syntax. VIEW 생성

```
CREATE VIEW 뷰이름  
AS (쿼리)
```

Syntax. VIEW 삭제

```
DROP VIEW 뷰이름
```

8. 임시테이블(VIEW & WITH)

Syntax. WITH로 임시테이블 생성

```
WITH 임시테이블이름 AS(  
    임시테이블쿼리  
)  
임시테이블을 사용하는 메인쿼리
```

9. 사용자 정의 함수

9. 사용자 정의 함수(데이터타입)

데이터 타입	설명	최대 바이트
CHAR(n)	고정 문자열	255
VARCHAR(n)	가변 문자열	65,535
NCAHR(n)	고정 유니코드 문자열	255
NVARCHAR(n)	가변 유니코드 문자열	65,535
TINYTEXT	텍스트	255
TEXT		65,535
MEDIUMTEXT		16,777,215
LONGTEXT		4,294,967,295

9. 사용자 정의 함수(데이터타입)

데이터 타입	설명	크기
TINYINT	-128~127	1 바이트
SMALLINT	-32,768~32,767	2 바이트
INT	-2,147,483,648~2,147,483,647	4 바이트
BIGINT	$-2^{63} \sim 2^{63}-1$	8 바이트
DECIMAL(p,s)	$-10^{38}+1 \sim 10^{38}-1$ p는 자리수, s는 소수점자리수	5~17 바이트
NUMERIC(p,s)	decimal과 기능상 동일	5~17 바이트
SMALLMONEY	-21억~21억 화폐	4 바이트
MONEY	$-2^{63} \sim 2^{63}-1$ 화폐	8 바이트
FLOAT	$-1.79E+308 \sim 1.79E+308$ 실수	4~8 바이트
REAL	$-3.40E+38 \sim 3.40E+38$ 실수	4 바이트

9. 사용자 정의 함수(데이터타입)

데이터 타입	설명	크기
datetime	1753-1-1~9999-12-31, 3.33ms	8 바이트
datetime2	0001-1-1~9999-12-31, 100ns	6~8 바이트
smalldatetime	1900-1-1~2079-6-6, 1min	4 byte
date	0001-1-1~9999-12-31, 1day	3 byte
time	100ns단위 날짜 없이 시간만	3~5 byte

9. 사용자 정의 함수

Case 1. 단일 값 반환 사용자 정의 함수

```
CREATE FUNCTION 함수이름 (  
    매개변수1,  
    매개변수2  
) RETURNS 데이터타입  
BEGIN  
    <SQL 쿼리>;  
    RETURN 반환값  
END;
```

SET GLOBAL log_bin_trust_function_creators = 1;

9. 사용자 정의 함수

과제 1. 'StartDate', 'EndDate', 'ID' 를 매개변수로 하여
해당 기간 수익률을 구하는 사용자 함수를 만들어보세요.

10. 테이블 & 데이터 핸들링

10. 테이블 핸들링

Syntax. 테이블 생성

```
CREATE TABLE 테이블이름 (  
    컬럼이름1 데이터타입,  
    컬럼이름2 데이터타입  
);
```

Syntax. 테이블 제거

```
DROP TABLE 테이블이름;
```

10. 테이블 핸들링

Syntax.

ALTER TABLE 테이블이름

RENAME TO 바꿀테이블이름,

변경

← 테이블 이름

ADD COLUMN 추가컬럼이름 데이터타입

← 컬럼 추가

CHANGE COLUMN 기존컬럼이름 바꿀컬럼이름 데이터타입,

← 컬럼 이름 변경

CHANGE COLUMN 컬럼이름 컬럼이름 바꿀데이터타입,

← 데이터타입 변경

DROP COLUMN 컬럼이름;

← 컬럼 삭제

10. 데이터 핸들링(데이터 추가)

Case 1. 특정 컬럼 데이터를 추가

```
INSERT INTO 테이블이름  
    (컬럼이름1, 컬럼이름2)  
VALUES(입력값1, 입력값2);
```

Case 2. 전체 컬럼의 데이터를 추가

```
INSERT INTO 테이블이름  
VALUES(입력값1, 입력값2, 입력값3);
```


10. 데이터 핸들링(데이터 수정)

Case 1. 데이터 삭제

```
DELETE FROM 테이블이름  
WHERE 조건;
```

** Preferences → SQL Editor → Safe Updates 체크 해제*

Case 2. 데이터 수정

```
UPDATES 테이블이름  
SET 컬럼이름1 = 입력값1, 컬럼이름2 = 입력값2  
WHERE 조건;
```

10. 테이블 핸들링(제약조건)

데이터 타입	설명
AUTO_INCREMENT	행 추가 되면 자동으로 1씩 증가(Numbering)
NOT NULL	NULL 값 불허용
UNIQUE	열에 있는 모든 값들은 중복되면 안됨
PRIMARY KEY	해당 열이 Primary Key가 된다는 뜻. (Not null & Unique)
DEFAULT	사용자가 지정하지 않으면 입력되는 기본 값