

Project 2

2022년 2학기	
제출일	2022. 11. 01.
과목명	데이터베이스및데이터시각화
담당교수	이기훈

학과	컴퓨터정보공학부	학과	컴퓨터정보공학부
학번	2017202006	학번	2017202061
이름	박준용	이름	전승권
학과	컴퓨터정보공학부	학과	컴퓨터정보공학부
학번	2017202076	학번	2019202020
이름	이민석	이름	이세연

KWANGWOON UNIVERSITY





1. 웹 애플리케이션 소개

금융 시장 플랫폼 웹 어플리케이션을 직접 구현한다.

이때 단순히 웹 어플리케이션의 ui만 따라하는 것이 아니라 ER 다이어그램을 이용해 체계적으로 DB를 설계한다. 구현 시 설계를 바탕으로 백 엔드에서 SQL을 활용한 DBMS가 동작하여 실사용이 가능한 데이터베이스와 이를 기반으로 사용자에게 시각적인 웹 화면을 구성하는 것이 궁극적

인 목표이다.

2. 구현 환경 기술

DBMS	Framework	Language	Library
			

3. 모든 기능 설명

● 기본 기능

이용자

- 회원가입
- 로그인
- 개인 정보 조회
- 보유 주식 조회
- 게시글 작성, 삭제

관리자

- 회원 관리
- 주식회사 등록
- 주식회사 정보 변경
- 게시판 관리

주식 관련 기능

- 주식 회사 정보 조회
- 종목 토론 게시판
- 주주 현황
- 데이터 시각화

● 특별 기능

종목 조회

- 관심 종목 조회
관심 있는 종목을 즐겨찾기나 찜 기능과 같이 모아 나열해 조회할 수 있다.
- 최근 본 종목 조회
최근 본 종목을 순서대로 나열하여 조회할 수 있다.

비밀번호 암호화

- 회원가입시 필요한 비밀번호를 랜덤한 문자열로 암호화하여 개인정보를 보호할 수 있다.

수익률 계산

- 보유 주식들의 구매가 대비 현재가를 통해 보유 주식들의 종목별로 수익률을 계산한다.
- 보유한 모든 주식들의 총 수익률도 계산한다.

랭킹 시각화

- 거래량 랭킹 시각화
하루동안 거래량이 가장 많은 종목별로 랭킹을 시각화해 보여준다.
- 주식 상승/하락률 랭킹 시각화
하루동안 가격의 상승률과 하락률이 가장 큰 순서대로 랭킹을 시각화해 보여준다.

연령대별 주식 보유 현황 시각화

● 페이지 디자인

홈페이지 로고

아이디

비밀번호

로그인

회원가입 / 관리자

<로그인 화면>

홈페이지 로고

아이디 *

비밀번호 *

비밀번호 확인 *

사용자명 *

생년월일 *

전화번호 *

이메일 *

가입하기

취소

<회원가입 화면>

오늘의 뉴스



[시황] 코스피, 외인 매도에 2260대 마감...반도체...

코스피가 28일 반도체 대형주 급락에 낙폭이 확대돼 2260대에서 장을 마쳤다. 기업 실적 발표에 대...

데일리안

[fn마켓시황]코스피, 지갑 열던 외인 ... 파이낸셜뉴스

코스닥 하락 마감 뉴스1

[코스닥(마감)] 746포인트(1.0 ... 서울경제

[특징주]목표가 올라간 삼성전자, 외국 ... 아시아경제

[특징주] SK하이닉스, 적자 지속 전 ... 머니S

[특징주]현대모비스, 3분기 호실적에 ... 이데일리

상승률 상위		하락률 상위		거래량		시가총액	
1	이수화학	25,200	▲ 5,800 +29.90%	4	나노	1,335	▲ 260 +24.19%
2	화진기계	4,655	▲ 920 +24.63%	7	야박코	17,100	▲ 2,800 +19.58%
3	효성중공업	57,400	▲ 8,390 +17.02%	8	에이치와이티씨	12,050	▲ 1,950 +19.31%
4	신승홀딩스	8,110	▲ 1,010 +14.23%	9	위드덕	13,700	▲ 2,150 +18.61%
5	한미글로벌	39,650	▲ 4,750 +13.61%	10	한국택키지	3,030	▲ 465 +18.13%

인기 검색

1	카카오	50,700	▲ +4.00%
2	사조씨푸드	4,690	▲ +1.96%
3	삼성전자	59,400	▲ +3.66%
4	이수화학	25,200	▲ +29.90%
5	두산에너빌리티	13,250	▼ -2.57%

업종 상위

1	서비스업	+2.27%	▲ 한미글로벌
2	전기전자	+2.03%	▲ 효성중공업
3	시가총액규모대	+1.30%	▲ 카카오뱅크
4	시가총액규모소	+1.19%	▲ 이수화학
5	철강금속	+1.17%	▲ SIMPAC

나의 관심

최근 검색

로그인 하시면
나의 관심 리스트를
볼 수 있습니다.

로그인

조회금등

인기토론

1	카카오	↑1
2	사조씨푸드	↑1
3	삼성전자	↑1
4	이수화학	↑1
5	두산에너빌리티	—
6	카카오뱅크	↑1
7	금양	↑1
8	NAVER	↑2
9	SK하이닉스	↑1
10	한미글로벌	↑1

<메인 페이지>

홈페이지 로고

종목명,지수명 입력



통합검색

로그인 / 회원가입



국내 해외 뉴스 투자정보 토론 MY 한글

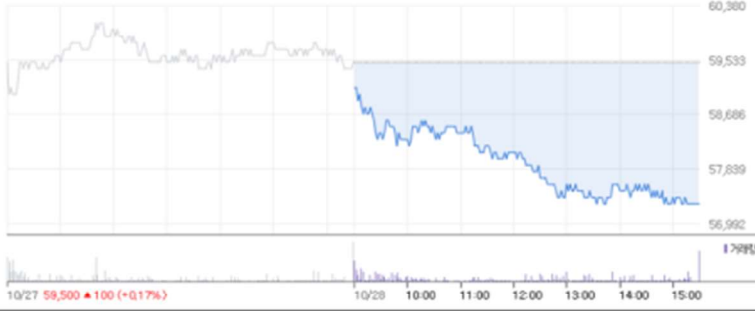
삼성전자 005930

57,300

전일대비 ▼2,200 | -3.70%

전일 59,500 | 고가 59,200 (상한가 77,300) | 거래량 20,608,580
시가 58,900 | 저가 57,200 (하한가 41,700) | 거래대금 1,195,132 백만

선차트 1일 1주일 3개월



기업소개

한국 및 DX부문 해외 9개 지역총괄과 DS부문 해외 5개 지역총괄, SDC, Harman 등 233개의 종속기업으로 구성된 글로벌 전자기업임.

주요주주현황

주요주주	소유주식수(주)	지분율
삼성생명보험 외 15인	1,241,176,035	20.79
국민연금공단	458,637,667	7.68
BlackRock Fund Advisors 외 15인	300,391,061	5.03

종목토론실 글쓰기

더보기 >

이 종목 지금 사는데 좋을까요? [1]

11/01 21:04

나의 관심

최근 검색

로그인 하시면
나의 관심 리스트를
볼 수 있습니다.

로그인

조회등급

인기도론

- 1 카카오 ↑1
- 2 사조씨푸드 ↓1
- 3 삼성전자 ↑1
- 4 이수화학 ↓1
- 5 두산에너지빌리티 —
- 6 카카오뱅크 ↑1
- 7 금영 ↓1
- 8 NAVER ↑2
- 9 SK하이닉스 ↓1
- 10 한미글로벌 ↓1

<기업정보 화면>

공지사항

No	제목	내용	조회수
1	필독	필독	55

종목 추가

종목코드	<input type="text"/>	중복확인
종목명	<input type="text"/>	
상장주식수	<input type="text"/>	
종목가격	<input type="text"/>	
기업이름	<input type="text"/>	
주요주주 및 주식수	<div><div>+</div><div>주요주주</div><div>주식수</div></div>	
기업개요	<div></div>	
등록		취소

[로그] 금융

종목명/지수명 입력

Q

통합검색

주식왕박주식

마이페이지

로그아웃

홈

국내

해외

뉴스

투자정보

토론

공지사항

환율

공지사항

1

필독

필독

공지사항 페이지입니다.

작성자

주식왕박주식

작성시간

22/10/31 22:20

조회수

55

<다음글>

다음글이 없습니다.

<이전글>

이전글이 없습니다.

최근조회

MY STOCK

삼성전자

57,300

▼ 2,200

KODEX 200선..

3,495

▲ 90

SK하이닉스

83,400

▼ 6,600

LG에너지솔루..

532,000

▼ 8,000

인기검색종목

코스피

코스닥

1

삼성전자

57,300

▼ 2,200

2

카카오

48,750

—

3

포오롱인더우

23,700

▼ 100

4

DL이앤씨우

22,500

▼ 250

5

KODEX 골드

11,030

▼ 35

6

한국조선해양

73,200

▲ 1,600

7

LG화학

615,000

▲ 16,000

8

기아

66,200

▲ 200

9

TIGER 200 IT

27,350

▼ 770

10

KODEX 건설

2,720

▼ 15

공지사항

더보기

[서울경 살청] 수익률 300%보장...

<공지사항 페이지>

[로그] 금융

종목명/지수명 입력

Q

통합검색

주식왕박주식

마이페이지

로그아웃

홈

국내

해외

뉴스

투자정보

토론

환율

토론게시판

2

이 종목 꼭 사세요!

삼성전자

지금 사는게 가장 좋아요.

작성자

주식왕김부자

작성시간

22/10/30 22:10

조회수

100

<다음글>

다음글이 없습니다.

<이전글>

이 종목 어때요?

주린이

네 감사합니다!

22/10/31 23:50

최근조회

MY STOCK

삼성전자

57,300

▼ 2,200

KODEX 200선..

3,495

▲ 90

SK하이닉스

83,400

▼ 6,600

LG에너지솔루..

532,000

▼ 8,000

인기검색종목

코스피

코스닥

1

삼성전자

57,300

▼ 2,200

2

카카오

48,750

—

3

포오롱인더우

23,700

▼ 100

4

DL이앤씨우

22,500

▼ 250

5

KODEX 골드

11,030

▼ 35

6

한국조선해양

73,200

▲ 1,600

7

LG화학

615,000

▲ 16,000

8

기아

66,200

▲ 200

9

TIGER 200 IT

27,350

▼ 770

10

KODEX 건설

2,720

▼ 15

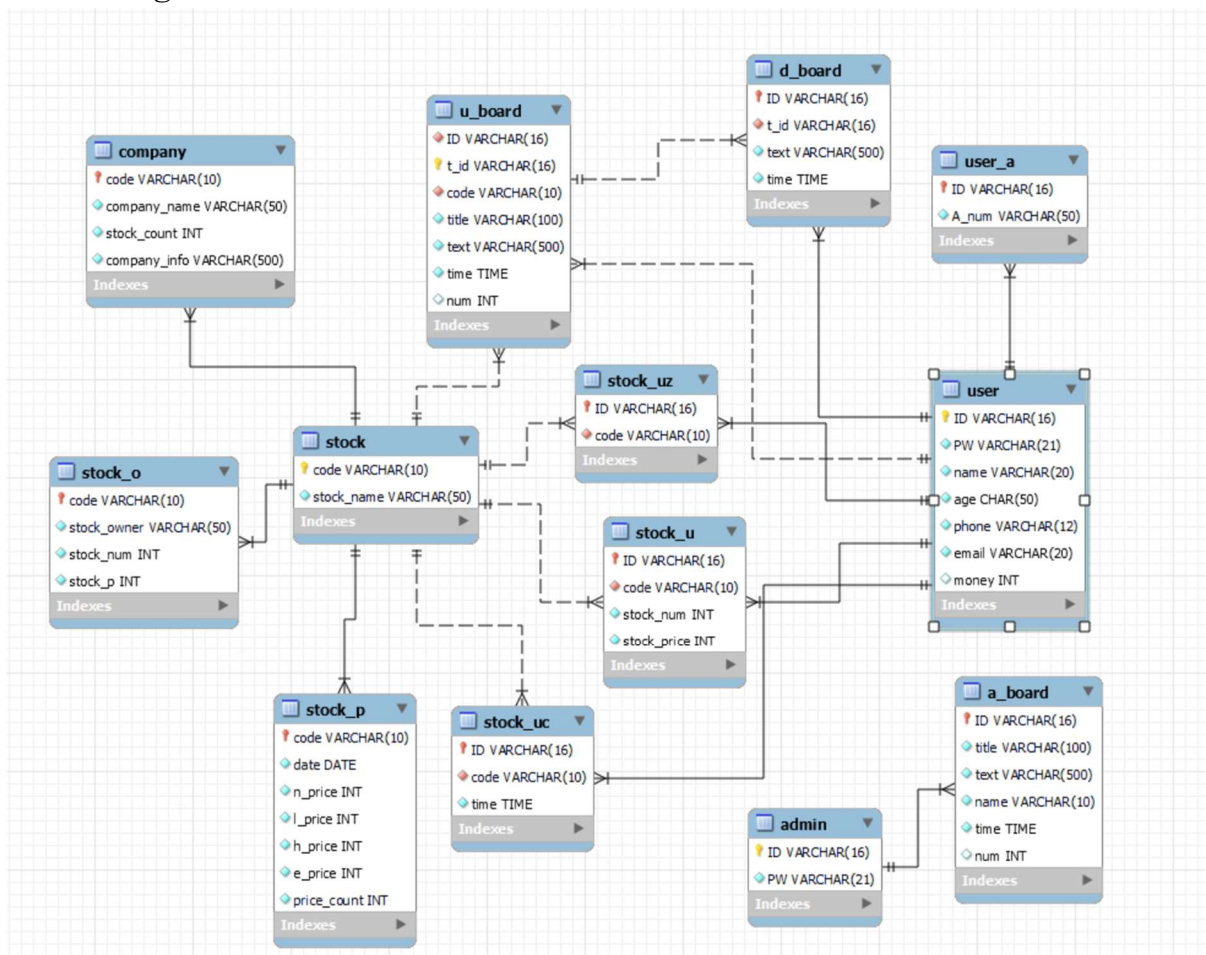
공지사항

더보기

[서울경 살청] 수익률 300%보장...

<토론게시판 페이지>

4. ER diagram



위는 이번 프로젝트에서 구현한 주식 매매 프로그램 DB의 ER-diagram이다. 총 테이블 수는 13개이다. 먼저 주식의 정보인 종목코드, 종목이름의 정보를 가지고 있는 테이블 STOCK을 기준으로 해당 주식의 기업상세정보(COMPANY), 주주현황(STOCK_O), 종목시세(STOCK_P) 테이블이 code를

통해 foreign key로 연결 되어있다. 사용자의 보유주식(STOCK_U), 최근 조회종목(STOCK_UZ), 관심 종목(STOCK_UC) 테이블은 종목코드와 사용자ID를 통해 foreign key로 각각 STOCK 테이블과 USER 테이블로 연결되어 있다. 각 주식의 종목토론게시판에 관련 테이블인 (U_BOARD)테이블은 STOCK, USER 테이블을 code와 사용자ID를 기준 foreign key로 연결되어 있고 해당 게시판에 댓글 작성을 위한 테이블인 D_BOARD 테이블은 U_BOARD의 글id(t_id)를 기준으로 foreign key로 연결 되어있다. 마지막으로 해당 사이트 관리자 정보인 admin 테이블과 관리자가 작성하는 공지사항게시판 테이블인(A_BOARD) 테이블이 관리자ID를 기준으로 연결 되어있다.

5. 모든 테이블 설명

STOCK CREATE TABLE

```
CREATE TABLE STOCK(
    code varchar(10) NOT NULL PRIMARY KEY,
    stock_name varchar(50) NOT NULL
);
```

STOCK TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
code	VARCHAR(10)	NOT NULL, PRIMARY KEY	종목코드
stock_name	VARCHAR(50)	NOT NULL	종목이름

STOCK TABLE DESCRIPTION

각 주식에 대한 상세정보(종목코드, 종목이름) 가 담긴 테이블이다.

COMPANY CREATE TABLE

```
CREATE TABLE COMPANY(
    code varchar(10) NOT NULL PRIMARY KEY,
    company_name varchar(50) NOT NULL,
    stock_count bigint NOT NULL,
    company_info varchar(500) NOT NULL,
    FOREIGN KEY (code) REFERENCES STOCK(code)
);
```

COMPANY TABLE COMPONENT

Column name	Data type	Detailed option	Column
-------------	-----------	-----------------	--------

			Description
code	VARCHAR(10)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF STOCK	종목코드
company_name	VARCHAR(50)	NOT NULL	종목이름
stock_count	BIGINT	NOT NULL	상장주식수
company_info	VARCHAR(500)	NOT NULL	기업개요

COMPANY TABLE DESCRIPTION

현재 사이트에 주식정보가 올라가 있는 회사에 대한 상세정보(종목코드 , 종목이름, 상장주식수, 기업개요) 의 정보가 담긴 테이블이다.

STOCK_O CREATE TABLE

```
CREATE TABLE STOCK_O(
  code varchar(10) NOT NULL,
  stock_owner varchar(50) NOT NULL,
  stock_num int NOT NULL,
  stock_p float NOT NULL,
  FOREIGN KEY (code) REFERENCES STOCK(code)
);
```

STOCK_O TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
code	VARCHAR(10)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF STOCK	종목코드
stock_owner	VARCHAR(50)	NOT NULL	주요주주
stock_num	INT	NOT NULL	보유주식수
stock_p	FLOAT	NOT NULL	보유지분

STOCK_O TABLE DESCRIPTION

STOCK 데이터베이스에 등록된 주식에 대한 주주현황이 담긴 테이블이다. 해당 테이블에는 주식 종목코드, 해당 주식의 주요주주, 주요주주의 보유 주식수 및 보유 지분 퍼센트에 대한 정보를 담고 있다.

STOCK_P CREATE TABLE

```

CREATE TABLE STOCK_P(
    code varchar(10) NOT NULL PRIMARY KEY,
    date DATE NOT NULL,
    n_price int NOT NULL,
    l_price int NOT NULL,
    h_price int NOT NULL,
    e_price int NOT NULL,
    price_count int NOT NULL,
    FOREIGN KEY (code) REFERENCES STOCK(code)
);

```

STOCK_P TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
code	VARCHAR(10)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF STOCK	종목코드
date	DATE	NOT NULL	날짜
n_price	INT	NOT NULL	시가
l_price	INT	NOT NULL	저가
h_price	INT	NOT NULL	고가
e_price	INT	NOT NULL	종가
price_count	INT	NOT NULL	거래량

STOCK_P TABLE DESCRIPTION

STOCK 데이터베이스에 저장된 주식에 대한 해당 날짜 종목 시세 정보를 가지고 있는 테이블이다. 해당 테이블에는 주식 종목코드, 날짜, 주식의 현재가격을 나타내는 시가, 해당 날짜에서 가장 낮은 가격인 저가, 해당 날짜에서 가장 높은 가격인 고가, 해당 날짜에서 거래가 종료되었을 때 종료시점의 가격인 종가, 해당 날짜에 주식이 거래된 거래량에 대한 정보를 가지고 있다.

USER CREATE TABLE

```
CREATE TABLE USER(  
    ID varchar(16) NOT NULL PRIMARY KEY,  
    PW varchar(21) NOT NULL,  
    name varchar(20) NOT NULL,  
    age char(50) NOT NULL,  
    phone varchar(12) NOT NULL UNIQUE,  
    email varchar(20) NOT NULL,  
    money int default 0  
);  
DELIMITER $$  
CREATE TRIGGER trig_user_check BEFORE INSERT ON USER  
FOR EACH ROW  
BEGIN  
    IF (NEW.ID REGEXP '[A-Za-z0-9]{5,15}') = 0 THEN  
        SIGNAL SQLSTATE '10001'  
        SET MESSAGE_TEXT = "id는 영문이나 숫자고 길이는 5~15자 입니다.";  
    ELSEIF (NEW.PW REGEXP '(?=[A-Za-z])(?=[0-9])[A-Za-z0-9]{8,20}') = 0 THEN  
        SIGNAL SQLSTATE '10002'  
        SET MESSAGE_TEXT = "비밀번호는 8~20자, 최소 하나의 문자 및 하나의 숫자를 포함해야 합니다.";  
    ELSEIF (NEW.age REGEXP '[0-9]{6}-[0-9]{7}') = 0 THEN  
        SIGNAL SQLSTATE '10003'  
        SET MESSAGE_TEXT = "주민번호는 총 13자리 입니다.";  
    ELSEIF (NEW.phone REGEXP '[0-9]{0,12}') = 0 THEN  
        SIGNAL SQLSTATE '10004'  
        SET MESSAGE_TEXT = "전화번호는 9~11자 입니다.";  
    END IF;  
END$$
```

USER TABLE COMPONENT

Column name	Data type	Detailed option	Column
-------------	-----------	-----------------	--------

			Description
ID	VARCHAR(16)	NOT NULL, PRIMARY KEY	사용자ID
PW	VARCHAR(21)	NOT NULL	비밀번호
name	VARCHAR(20)	NOT NULL	사용자 이름
age	CHAR(50)	NOT NULL	나이
Phone	VARCHAR(12)	NOT NULL	전화번호
email	VARCHAR(20)	NOT NULL	이메일
money	INT	Default 0	계좌 잔액

USER TABLE DESCRIPTION

해당 주식 사이트를 이용하는 이용자에 대한 정보를 담고 있는 테이블이다. 해당 테이블은 이용자 ID 및 비밀번호, 사용자 이름, 사용자 주민등록번호, 사용자 전화번호, 이메일, 계좌 잔액에 대한 정보를 가지고 있다. 계좌 잔액에 대한 정보를 제외한 나머지 정보는 회원가입시 입력하는 정보이며 id는 영문이나 숫자 길이는 5~15자입니다. 비밀번호는 8~20자, 최소 하나의 문자 및 하나의 숫자를 포함하게 한다. 주민등록 번호는 총 13자리를 사용하며, 전화번호 9~11자로 제한한다. 사용자의 계좌 잔액 정보는 default 0으로 설정한 후 가입 후 계좌 연동을 통해 update 할 예정이다.

USER_A CREATE TABLE

```

> CREATE TABLE USER_A(
    ID varchar(16) NOT NULL,
    A_num varchar(50) NOT NULL,
    PRIMARY KEY (ID, A_num),
    FOREIGN KEY (ID) REFERENCES USER(ID)
~ );

```

USER_A TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
ID	VARCHAR(16)	NOT NULL, PRIMARY KEY	사용자 ID
A_num	VARCHAR(50)	NOT NULL, PRIMARY KEY	계좌번호

USER_O TABLE DESCRIPTION

사용자의 계좌번호를 가지고 있는 테이블이다. 한 명의 사용자가 여러 개의 계좌를 가지고 있을 수 있기 때문에 따로 계좌정보에 대한 테이블을 생성하였다.

STOCK_U CREATE TABLE

```
CREATE TABLE STOCK_U(  
    ID varchar(16) NOT NULL,  
    code varchar(10) NOT NULL,  
    stock_num int NOT NULL,  
    stock_price int NOT NULL,  
    PRIMARY KEY (ID, code),  
    FOREIGN KEY (ID) REFERENCES USER(ID),  
    FOREIGN KEY (code) REFERENCES STOCK(code)  
);
```

STOCK_U TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
ID	VARCHAR(16)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF USER	사용자ID
code	VARCHAR(10)	NOT NULL, FOREIGN KEY OF STOCK	종목코드
stock_num	INT	NOT NULL	주식수
price	INT	NOT NULL	구매가격

STOCK_U TABLE DESCRIPTION

사용자가 구매한 주식 정보를 가지고 있는 테이블이다. 해당 테이블은 사용자의 ID, 구매한 주식의 종목코드, 해당 주식 구매량, 구매가격 정보를 가지고 있다.

STOCK_UZ CREATE TABLE

```
CREATE TABLE STOCK_UZ(  
    ID varchar(16) NOT NULL,  
    code varchar(10) NOT NULL,  
    PRIMARY KEY(ID, code),  
    FOREIGN KEY (ID) REFERENCES USER(ID),  
    FOREIGN KEY (code) REFERENCES STOCK(code)  
);
```

STOCK_UZ TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
-------------	-----------	-----------------	--------------------

ID	VARCHAR(16)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF USER	사용자ID
code	VARCHAR(10)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF STOCK	종목코드

STOCK_UZ TABLE DESCRIPTION

사용자가 관심있는 주식 종목에 대한 즐겨찾기 기능을 추가하여 해당 주식을 쉽게 찾아볼 수 있게 하는 테이블이다. 관심 등록한 주식의 종목코드와 사용자 ID 정보가 들어있다.

STOCK_UC CREATE TABLE

```
CREATE TABLE STOCK_UC(
  ID varchar(16) NOT NULL,
  code varchar(10) NOT NULL,
  time TIME NOT NULL,
  PRIMARY KEY(ID, time),
  FOREIGN KEY (ID) REFERENCES USER(ID),
  FOREIGN KEY (code) REFERENCES STOCK(code)
);
```

STOCK_UC TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
ID	VARCHAR(16)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF USER	사용자ID
code	VARCHAR(10)	NOT NULL, FOREIGN KEY OF STOCK	종목코드
time	TIME	NOT NULL, PRIMARY KEY	조회시간

STOCK_UC TABLE DESCRIPTION

사용자가 최근 조회한 주식 정보를 가지고 있는 테이블이다. 해당 테이블은 사용자의 ID, 주식 종목코드, 해당 주식을 조회한 시간정보를 가지고 있다.

ADMIN CREATE TABLE

```

CREATE TABLE ADMIN(
    ID varchar(16) NOT NULL PRIMARY KEY;
    PW varchar(21) NOT NULL;
);

```

ADMIN TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
ID	VARCHAR(16)	NOT NULL, PRIMARY KEY	관리자ID
PW	VARCHAR(21)	NOT NULL,	관리자 비밀번호

ADMIN TABLE DESCRIPTION

사이트를 관리하는 관리자 정보를 가지고 있는 테이블이다. 관리자의 ID 혹은 비밀번호를 가지고 있다.

A_BOARD CREATE TABLE

```

CREATE TABLE A_BOARD(
    ID varchar(16) NOT NULL PRIMARY KEY,
    title varchar(100) NOT NULL,
    text varchar(500) NOT NULL,
    name varchar(10) NOT NULL,
    time TIME NOT NULL,
    num INT default 0,
    FOREIGN KEY (ID) REFERENCES ADMIN(ID)
);

```

A_BOARD TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
ID	VARCHAR(16)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF ADMIN	관리자ID
title	VARCHAR(100)	NOT NULL	제목
text	VARCHAR(500)	NOT NULL	내용
name	VARCHAR(10)	NOT NULL	작성자
time	TIME	NOT NULL	작성시간

num	INT	NOT NULL	조회수
-----	-----	----------	-----

A_BOARD TABLE DESCRIPTION

관리자가 공지사항을 작성하기 위한 게시판용 테이블이다. 해당 테이블에는 관리자ID 및 게시판 제목, 내용, 작성자, 작성시간, 해당 게시물 조회수에 대한 정보를 저장할 수 있다.

U_BOARD CREATE TABLE

```
CREATE TABLE U_BOARD(
  ID varchar(16) NOT NULL,
  t_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  code varchar(10) NOT NULL,
  title varchar(100) NOT NULL,
  text varchar(500) NOT NULL,
  time TIME NOT NULL,
  num int default 0,
  FOREIGN KEY (ID) REFERENCES USER(ID),
  FOREIGN KEY (code) REFERENCES STOCK(code)
);
```

U_BOARD TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
ID	VARCHAR(16)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF USER	사용자 ID
t_id	INT	NOT NULL, AUTO_INCREMENT	글 ID
code	VARCHAR(10)	NOT NULL, FOREIGN KEY OF STOCK	종목코드
title	VARCHAR(100)	NOT NULL	글 제목
text	VARCHAR(500)	NOT NULL	글 내용
time	TIME	NOT NULL	글 작성 시간
num	INT	Default 0	조회수

U_BOARD TABLE DESCRIPTION

사용자 간의 특정 주식 종목에 대해 토론할 수 있는 종목 토론 게시판에 대한 정보를 가지고 있는 테이블이다. 해당 테이블은 작성자의 ID, 작성한 글에 대한 ID, 토론 주제 주식의 코드, 해당 글 제목, 해당 글 내용, 해당 글 작성시간, 글 조회수에 대한 정보를 가지고 있다.

D_BOARD CREATE TABLE

```
CREATE TABLE D_BOARD(  
    ID varchar(16) NOT NULL PRIMARY KEY,  
    t_id int NOT NULL,  
    text varchar(500) NOT NULL,  
    time TIME NOT NULL,  
    FOREIGN KEY (ID) REFERENCES USER(ID),  
    FOREIGN KEY (t_id) REFERENCES U_BOARD(t_id)  
);
```

D_BOARD TABLE COMPONENT

Column name	Data type	Detailed option	Column Description
ID	VARCHAR(16)	NOT NULL, PRIMARY KEY, FOREIGN KEY OF USER	사용자 ID
t_id	INT	NOT NULL, FOREIGN KEY OF U_BOARD	글 ID
text	VARCHAR(500)	NOT NULL	글 내용
time	TIME	NOT NULL	글 작성 시간

D_BOARD TABLE COMPONENT

위 U_BOARD에서 설명한 것과 같은 종목 토론 게시판에서 어떠한 글에 대해 댓글을 쓰는 정보를 가지고 있는 테이블이다. 해당 테이블은 사용자 ID, 댓글을 다는 글에 대한 ID, 댓글 내용, 댓글 작성 시간에 대한 정보를 가지고 있다.

6. 사용되는 SQL문 설명

구분	SQL 문 종류
Definition	CREATE, CREATE USER, ALTER, DROP USER
Modification	INSERT, UPDATE, DELETE, TRIGGER
Query	SELECT

join	JOIN, OUTER JOIN(LEFT, RIGHT, FULL)
aggregation	GROUP BY, HAVING
DATE	TIME, DATE, NOW
view	VIEW
Operator	IS NOT NULL, AND, BETWEEN
Transaction	IF-ELSEIF, COUNT, FLUSH
etc	SET, DISTINCT, OFFSET, EXISTS, NOT EXISTS, LIMIT, OFFSET, SHOW, EXPLAIN, CASE, SUM, AS, DESC, ASC

7. 각 SQL문 실행 화면 캡처 및 설명

SQL

```
INSERT STOCK(code, stock_name)
VALUES
('005930', '삼성전자'),
('373220', 'LG에너지솔루션'),
('000660', 'SK하이닉스'),
('207940', '삼성바이오로직스'),
('051910', 'LG화학'),
('005380', '현대차');
```

```
INSERT user(ID, PW, name, age, phone, email, money)
VALUES
('aaaaa', '123aaaaa', '홍길동', '21', '01011111111', 'aaaaa@naver.com', '0'),
('bbbbb', '123bbbbb', '최부자', '35', '01022222222', 'bbbbb@naver.com', '0'),
('ccccc', '123ccccc', '박주식', '42', '01033333333', 'ccccc@naver.com', '0'),
('ddddd', '123ddddd', '박준웅', '26', '01044444444', 'ddddd@naver.com', '0'),
('eeeee', '123eeee', '전승권', '51', '01055555555', 'eeee@naver.com', '0'),
('fffff', '123fffff', '이민석', '33', '01066666666', 'fffff@naver.com', '0'),
('ggggg', '123ggggg', '이세연', '25', '01077777777', 'ggggg@naver.com', '0'),
('hhhhh', '123hhhhh', '이기업', '27', '01088888888', 'hhhhh@naver.com', '0'),
('iiiii', '123iiiii', '김무식', '39', '01099999999', 'iiiii@naver.com', '0'),
('jjjjj', '123jjjjj', '신짱구', '58', '01000000000', 'jjjjj@naver.com', '0'),
('kkkkk', '123kkkkk', '우영우', '26', '01012121212', 'kkkkk@naver.com', '0'),
('lllll', '123lllll', '이수만', '32', '01013131313', 'lllll@naver.com', '0'),
('mmmmm', '123mmmmm', '양현석', '25', '01014141414', 'mmmmm@naver.com', '0');
```

사용 SQL

- INSERT

SQL 설명

INSERT를 사용하여 company 테이블에 종목코드, 종목이름, 상장주식수, 기업개요에 대한 정보를 insert 한다. Company뿐만 아니라 다른 DB 구축에도 사용된다.

결과

	ID	PW	name	age	phone	email	money
▶	aaaaa	123aaaaa	홍길동	21	01011111111	aaaaa@naver.com	0
	bbbbb	123bbbbb	최부자	35	01022222222	bbbbb@naver.com	0
	ccccc	123ccccc	박주식	42	01033333333	ccccc@naver.com	0
	ddddd	123ddddd	박준용	26	01044444444	ddddd@naver.com	0
	eeeee	123eeeee	전승권	51	01055555555	eeeee@naver.com	0
	fffff	123fffff	이민석	33	01066666666	fffff@naver.com	0
	ggggg	123ggggg	이세연	25	01077777777	ggggg@naver.com	0
	hhhhh	123hhhhh	이기업	27	01088888888	hhhhh@naver.com	0
	iiii	123iiii	김무식	39	01099999999	iiii@naver.com	0
	jjjjj	123jjjjj	신짱구	58	01000000000	jjjjj@naver.com	0
	kkkkk	123kkkkk	우영우	26	01012121212	kkkkk@naver.com	0
	lllll	123lllll	이수만	32	01013131313	lllll@naver.com	0
	mmm...	123mmm...	양현석	25	01014141414	mmmmm@naver.c...	0
★	NULL	NULL					
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	code	stock_name
▶	000660	SK하이닉스
	005380	현대차
	005930	삼성전자
	051910	LG화학
	207940	삼성바이오로직스
	373220	LG에너지솔루션
★	NULL	NULL

SQL

```
DELETE FROM company
WHERE code="005930";
```

사용 SQL

- DELETE

SQL 설명

DELETE 문을 사용하여 company 테이블의 주식정보를 삭제한다. 관리자를 통해 기존 테이블의 주식 정보를 개별로 삭제하는데 사용된다. 사용자가 보유 주식을 판매하였을 때 보유주식 테이블에서 delete를 통해 삭제한다.

결과

	code	company_name	stock_count	company_info
▶	005930	삼성전자	596978255	한국 및 DX 부문 해외 9개 지역총괄과 DS 부문 해...
	373220	LG에너지솔루션	23400000	동사는 EV, ESS, IT기기, LEV 등에 적용되는 전...
★	NULL	NULL	NULL	NULL

	code	company_name	stock_count	company_info
▶	373220	LG에너지솔루션	23400000	동사는 EV, ESS, IT기기, LEV 등에 적용되는 전...
★	NULL	NULL	NULL	NULL

위 예시 데이터에서 code 005930에 해당하는 주식의 정보가 삭제된 것을 확인할 수 있다.

SQL

```
SHOW TABLES;
```

사용 SQL

- SHOW

SQL 설명

DB에 존재하는 모든 TABLE들을 보여주는 SQL이다. DB 관리자에 의해 사용된다.

결과

	ID	PW	nickname	age	phone	email	money
▶	aaaaa	123aaaaa	홍길동	21	01011111111	aaaaa@naver.com	0
	bbbbb	123bbbbb	최부자	35	01022222222	bbbbb@naver.com	0
	ccccc	123ccccc	박주식	42	01033333333	ccccc@naver.com	0
	ddddd	123ddddd	박준용	26	01044444444	ddddd@naver.com	0
	eeeee	123eeeee	전승권	51	01055555555	eeeee@naver.com	0
	fffff	123fffff	이민석	33	01066666666	fffff@naver.com	0
	ggggg	123ggggg	이세연	25	01077777777	ggggg@naver.com	0
	hhhhh	123hhhhh	이기업	27	01088888888	hhhhh@naver.com	0
	iiii	123iiii	김무식	39	01099999999	iiii@naver.com	0
	jjjjj	123jjjjj	신짱구	58	01000000000	jjjjj@naver.com	0
	kkkkk	123kkkkk	우영우	26	01012121212	kkkkk@naver.com	0
	lllll	123lllll	이수만	32	01013131313	lllll@naver.com	0
	mmm...	123mmm...	양현석	25	01014141414	mmmmm@naver.c...	0
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SQL

```
select *
from stock_uc
order by time asc
limit 0,5;
```

사용 SQL

- Order by, ASC

SQL 설명

최근 조회 순서대로 오름차순 정렬을 진행하는 SQL문이다. 최근조회종목과 같은 기능에 사용된다.
Limit를 통해 5개 항목 출력

결과

	ID	code	time
▶	aaaaa	005380	00:00:20
	jjjjj	005380	00:00:50
	eeeee	000660	00:00:55
	kkkkk	207940	00:01:52
	mmmmm	005380	00:02:21
★	NULL	NULL	NULL

SQL

```
SELECT id, ((sum(stock_num*n_price)-sum(stock_num*stock_price))/sum(stock_num*stock_price) * 100 AS 수익률
FROM STOCK_U natural join STOCK_P
group by id;
```

사용 SQL

- SUM, AS, NATURAL JOIN, GROUP BY

SQL 설명

테이블 STOCK_U와 STOCK_P의 유일 공통 컬럼인 code로 natural join하여 사용자가 구매한 주식의 시세 정보로부터 수익률을 계산한다. 이 때 컬럼의 이름은 '수익률'로 하며 id끼리 그룹화하여 출력한다.

결과

id	수익률
aaaaa	-97.6989
ddddd	-90.3468
jjjjj	-90.0000
bbbbbb	-90.0000
ggggg	-80.2381
hhhhh	-69.7431
mmmmm	-80.0000
ccccc	-80.0000
kkkkk	-86.2833
iiii	-50.0000
eeeeee	0.0000
lllll	-50.0000
fffff	-50.0000

SQL

```
SELECT code, stock_name, price_count
FROM STOCK_P natural join STOCK
WHERE date='2022-10-31'
ORDER BY price_count DESC
limit 0,5;
```

사용 SQL

- NATURAL JOIN, ORDER BY, DESC, LIMIT

SQL 설명

해당 날짜의 주식 거래량 랭킹을 내림차순으로 최대 5개까지 출력한다. 이 때 출력하는 주식의 정보는 종목 코드, 종목 이름, 거래량이다.

결과

code	stock_name	price_count
005930	삼성전자	18926454
000660	SK하이닉스	6190029
051910	LG화학	740191
373220	LG에너지솔루션	626581
005380	현대차	612141

SQL

```
SELECT *
FROM STOCK_P JOIN STOCK ON STOCK_P.code = STOCK.code
WHERE date='2022-10-31'
ORDER BY (n_price-e_price)/e_price DESC
limit 0,5;
```

사용 SQL

- JOIN, ORDER BY, DESC, LIMIT

SQL 설명

주식 종목의 전날 종가 대비 현재 시가를 계산해서 해당 날짜의 종목 상승률 순으로 최대 5개까지 출력한다.

결과

code	date	n_price	l_price	h_price	e_price	price_count	code	stock_name
051910	2022-10-31	627000	609000	627000	615000	740191	051910	LG화학
005930	2022-10-31	58100	58000	59900	57300	18926454	005930	삼성전자
373220	2022-10-31	539000	516000	542000	532000	626581	373220	LG에너지솔루션
005380	2022-10-31	166000	162000	166500	164000	612141	005380	현대차
000660	2022-10-31	84300	82400	85400	83400	6190029	000660	SK하이닉스

SQL

```
SELECT *
FROM STOCK_P JOIN STOCK ON STOCK_P.code = STOCK.code
WHERE date='2022-10-31'
ORDER BY (n_price-e_price)/e_price ASC
limit 0,5;
```

사용 SQL

- JOIN, ORDER BY, ASC, LIMIT

SQL 설명

주식 종목의 전날 종가 대비 현재 시가를 계산해서 해당 날짜의 종목 하락률 순으로 최대 5개까지 출력한다.

결과

code	date	n_price	l_price	h_price	e_price	price_count	code	stock_name
207940	2022-10-31	884000	871000	892000	876000	39500	207940	삼성바이오로직스
000660	2022-10-31	84300	82400	85400	83400	6190029	000660	SK하이닉스
005380	2022-10-31	166000	162000	166500	164000	612141	005380	현대차
373220	2022-10-31	539000	516000	542000	532000	626581	373220	LG에너지솔루션
005930	2022-10-31	58100	58000	59900	57300	18926454	005930	삼성전자

SQL

```

WITH AGE AS (
    SELECT ID AS ID,
    ( CASE
        WHEN age>=20 AND age<=29 THEN '20대'
        WHEN age>=30 AND age<=39 THEN '30대'
        WHEN age>=40 AND age<=49 THEN '40대'
        WHEN age>=50 AND age<=59 THEN '50대'
        WHEN age>=60 THEN '60대 이상'
        ELSE '미성년자'
    END ) AS age
FROM user )
SELECT age, stock_name, sum(stock_num)
FROM AGE natural join STOCK_U natural join STOCK
group by age, code
order by age ASC;

```

사용 SQL

- WITH, AS, CASE, SUM, NATURAL JOIN, GROUP BY, ORDER BY

SQL 설명

CASE문으로 연령대를 나누고 WITH문으로 AGE라는 임시테이블을 만들어 저장한다. 그리고 연령별 주식 보유 현황을 나이와 종목 코드로 그룹화해서 나이 오름차순으로 출력한다.

결과

▶	20대	SK하이닉스	50
	20대	현대차	16
	20대	삼성전자	60
	20대	LG화학	1
	20대	삼성바이오로직스	3
	20대	LG에너지솔루션	3
	30대	현대차	10
	30대	LG화학	2
	30대	삼성바이오로직스	2
	30대	LG에너지솔루션	2
	40대	삼성전자	5
	50대	SK하이닉스	10
	50대	삼성바이오로직스	1

SQL

```

SELECT distinct ID, stock_name
FROM STOCK_U natural join STOCK
order by ID;

```

사용 SQL

- DISTINCT, NATURAL JOIN, ORDER BY

SQL 설명

테이블 STOCK_U와 STOCK을 JOIN해서 사용자의 주식 보유 현황을 ID 순서대로 출력한다.

결과

	ID	stock_name
▶	aaaaa	SK하이닉스
	aaaaa	현대차
	aaaaa	삼성전자
	bbbbb	현대차
	ccccc	삼성전자
	ddddd	SK하이닉스
	ddddd	LG화학
	eeeee	삼성바이오로직스
	fffff	LG에너지솔루션
	ggggg	현대차
	hhhhh	현대차
	hhhhh	LG에너지솔루션
	iiii	LG화학
	jjjjj	SK하이닉스
	kkkkk	삼성전자
	kkkkk	삼성바이오로직스
	kkkkk	LG에너지솔루션
	lllll	삼성바이오로직스
	mmm...	현대차

SQL

```
SELECT user.ID, COUNT(A_num) AS 보유계좌수
FROM user LEFT OUTER JOIN USER_A ON user.ID = USER_A.ID
GROUP BY user.ID;
```

사용 SQL

- COUNT, OUTER JOIN(LEFT), GROUP BY

SQL 설명

테이블 USER와 USER_A를 OUTER JOIN해서 사용자별로 등록한 계좌의 개수를 COUNT, 사용자 ID로 그룹화해 컬럼명 '보유계좌수'로 출력한다.

결과

ID	보유계좌수
aaaaa	4
bbbbbb	1
ccccc	2
dddddd	3
eeeeee	0
ffffff	0
ggggg	2
hhhhh	1
iiii	0
jjjjj	1
kkkkk	0
lllll	0
mmm...	0

SQL

```
update stock_u
set stock_num='10',stock_price='20000'
where ID='aaaaa' and code='000660';
```

사용 SQL

- UPDATE, AND, SET

SQL 설명

테이블의 데이터 값을 update한다. 만일 사용자가 보유주식을 매도, 혹은 주식 매매와 같은 동작을 하였을 때 사용자 보유 주식 테이블인 stock_u에서 update가 발생한다. 혹은 사용자의 가입정보, phone, email 정보 수정 시 update

	ID	code	stock_num	stock_price
▶	aaaaa	000660	35	2950500
	aaaaa	005380	1	166000
	aaaaa	005930	20	3320000
	bbbbbb	005380	10	1660000
	ccccc	005930	5	290500
	dddddd	000660	15	1264500
	dddddd	051910	1	627000
	eeeeee	207940	1	884000
	ffffff	373220	2	1078000
	ggggg	005380	5	840000
	hhhhh	005380	5	830000
	hhhhh	373220	2	1078000
	iiii	051910	2	1254000
	jjjjj	000660	10	843000
	kkkkk	005930	20	1162000
	kkkkk	207940	3	2652000
	kkkkk	373220	1	539000
	lllll	207940	2	1768000
	mmm...	005380	5	830000
*	NULL	NULL	NULL	NULL

	ID	code	stock_num	stock_price
▶	aaaaa	000660	10	20000
	aaaaa	005380	1	166000
	aaaaa	005930	20	3320000
	bbbbbb	005380	10	1660000
	ccccc	005930	5	290500
	dddddd	000660	15	1264500
	dddddd	051910	1	627000
	eeeeee	207940	1	884000
	ffffff	373220	2	1078000
	ggggg	005380	5	840000
	hhhhh	005380	5	830000
	hhhhh	373220	2	1078000
	iiii	051910	2	1254000
	jjjjj	000660	10	843000
	kkkkk	005930	20	1162000
	kkkkk	207940	3	2652000
	kkkkk	373220	1	539000
	lllll	207940	2	1768000
	mmm...	005380	5	830000
*	NULL	NULL	NULL	NULL

Update를 통해 ID aaaaa 인 사용자의 code 000660 주식 정보가 변경된 것을 확인할 수 있다.

SQL

```
select count(*)=1
from user
where ID = "aaaaa" and PW = "123aaaaa";
```

사용 SQL

- COUNT, AND

SQL 설명

로그인 시 로그인 정보가 user 테이블에 존재하면 count 값 1 반환, 없으면 0을 반환한다.

결과

	count(*)=1	
▶	1	로그인 데이터가 존재할 경우이다.
	count(*)=1	
▶	0	로그인 데이터가 존재하지 않을 경우이다.

SQL

```
select *
from stock_u
where code in(
    select code
    from stock_u
    where code = "005930");
```

사용 SQL

- NESTED SELECT, IN

SQL 설명

삼성전자 주식(code = 005930)을 가지고 있는 사용자의 ID 및 보유 주식 수, 가격을 출력한다.

결과

	ID	code	stock_num	stock_price
▶	aaaaa	005930	20	3320000
	cccc	005930	5	290500
	kkkkk	005930	20	1162000
*	NULL	NULL	NULL	NULL