

APT 4U

아파트 포유

배진경, 황순근, 김소정, 이수민



목차

A table of contents.

01 기획의도

02 ERD모델링

03 역할분담

04 DBConnection, DAO, VO

Part 01 기획의도

금리도 우대하는 '청년우대청약저축'

입력 2020.12.01 15:11 | 수정 2020.12.01 15:11 | 지면 B4

연봉 3000만원 넘지 않으면
年利 최고 3.3% '일석이조'

주택청약종합저축통장은 아파트를
분양받기 위해 꼭 필요한
금융상품이다. 청약통장을 갖고
있으면 만 19세부터 청약에 도전할 수
있다. 통장 가입 기간이 길수록
아파트에 당첨될 가능성이 높기
때문에 최대한 빨리 만들어 두는 것이
좋다. 하지만 연봉 3000만원 이하 만
19~34세인 사람들에게는 보다 특별한
매력이 있다. 아파트 분양 자격
확보뿐만 아니라 재테크 상품으로서 가치도 크기 때문이다.



사진=연합뉴스

일반적인 청약저축의 금리는 1개월 초과~1년 미만은 연 1.0%, 1년 이상~2년

신혼부부 매입임대 주택, '중·장년 신혼' 사이서도 각광

조선비즈 | 최상현 기자

입력 2020.12.07 06:00

서울주택도시공사(SH)가 공급하는 신혼부부 매입임대주택이 중장년층 사이에서도 각광받는 것으로 나타났다. 나이 제한이 없이 혼인신고 7년 이내라는 요건만 충족하면 재혼이라도 신청할 수 있기 때문이다.



부동산 주요뉴스

balconi 확장하
들여다볼까 걱정
'썬팅' 확산

"브랜드도, 나홀
가린다"... 치솟
청약 인

Part 01 기획의도



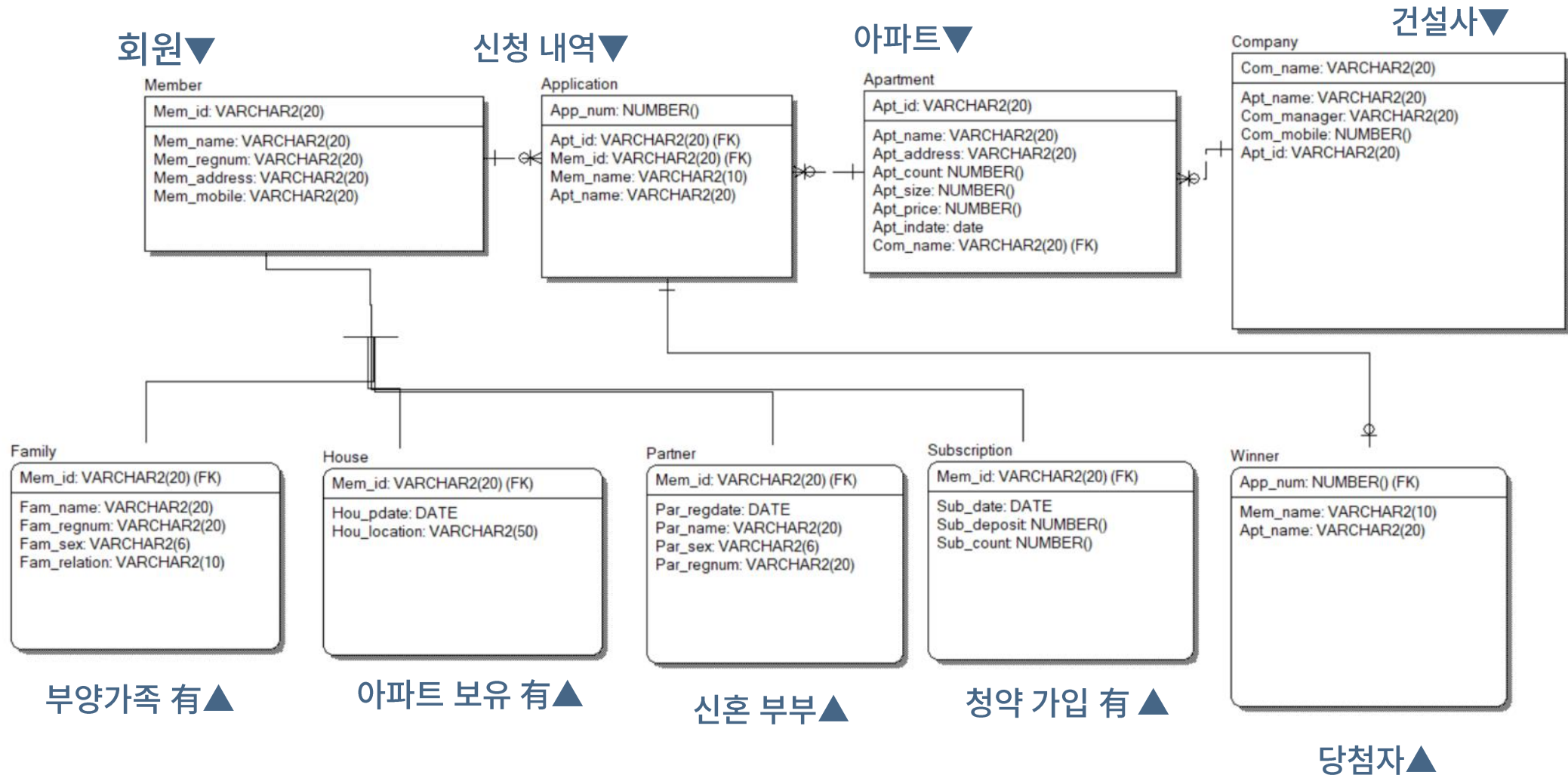
부동산값상승



주택청약지원청년층증가



분양아파트정보조회



ERD모델링 - 설계

Company	테이블												
컬럼명	Com_name	Apt_id	Apt_name	Com_man	Com_mobile								
PK/UK/NOT NULL	NOT NULL												
참조 테이블		Apartment	Apartment										
참조 컬럼		Apt_id	Apt_name										
CHECK													
데이터 타입	VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR2								
길이	30	20	50	20	20								
비고													
컬럼 설명													
Com_name	시공사 이름												
Apt_id	아파트의 고유 번호												
Apt_name	아파트의 이름												
Com_manager	시공사 매니저의 이름												
Com_mobile	시공사 매니저의 번호												
Apartment	테이블												
컬럼명	Apt_id	Apt_name	Apt_addr	Apt_coun	Apt_size	Apt_price	Apt_indat	Com_name					
PK/UK/NOT NULL	PK/NOT NULL							NOT NULL					
참조 테이블								Company					
참조 컬럼								Com_name					
CHECK													
데이터 타입	VARCHAR	VARCHAR	VARCHAR	NUMBER	VARCHAR	NUMBER	date	VARCHAR2					
길이	20	50	50		20			30					
비고								FK					
컬럼 설명													
Apt_id	아파트 별 고유 번호												
Apt_name	아파트의 이름												
Apt_address	아파트의 주소												
Apt_count	아파트의 분양 세대 수												
Apt_size	아파트의 평수												
Apt_price	아파트의 가격												
Apt_indate	아파트의 입주 가능 날짜												
Com_name	시공사의 이름												

ERD모델링 - 설계

Application	테이블													
컬럼명	App_num	Mem_name	Apt_name	Apt_id	Mem_id									
PK/UK/NOT NULL	PK/NOT NULL			NOT NULL	NOT NULL									
참조 테이블		Member	Apartmen	Apartmen	Member									
참조 컬럼		Mem_name	Apt_name	Apt_id	Mem_id									
CHECK														
데이터 타입	NUMBER	VARCHAR	VARCHAR	VARCHAR	VARCHAR2									
길이		20	50	20	20									
비고				FK	FK									
컬럼 설명														
App_num	신청서의 번호													
Mem_name	신청자의 이름													
Apt_name	신청한 아파트의 이름													
Apt_id	신청한 아파트의 고유 번호													
Mem_id	신청자의 ID													
Member	테이블													
컬럼명	Mem_id	Mem_name	Mem_reg	Mem_addr	Mem_mobile									
PK/UK/NOT NULL	PK/NOT NULL													
참조 테이블														
참조 컬럼														
CHECK														
데이터 타입	VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR2									
길이	20	20	20	50	20									
비고														
컬럼 설명														
Mem_id	가입자의 ID													
Mem_name	가입자의 이름													
Mem_regnum	가입자의 주민등록번호													
Mem_address	가입자의 주소													
Mem_mobile	가입자의 전화번호													

©Saebyeol Yu. Saebyeol's PowerPoint

ERD모델링 - 설계

Partner 테이블

컬럼명	Mem_id	Par_name	Par_regnum	Par_sex	Par_regdate
PK/UK/NOT NULL	NOT NULL				NOT NULL
참조 테이블	Member				
참조 컬럼	Mem_id				
CHECK					
데이터 타입	VARCHAR	VARCHAR	VARCHAR	VARCHAR	date
길이	20	20	20	6	
비고	FK				

컬럼 설명

Mem_id	가입자 중 신혼부부인 사람의 ID
Par_name	가입자 배우자의 이름
Par_regnum	가입자 배우자의 주민등록번호
Par_sex	가입자 배우자의 성별
Par_regdate	가입자의 혼인신고일자

Subscription 테이블

컬럼명	Mem_id	Sub_date	Sub_deposit	Sub_count
PK/UK/NOT NULL	NOT NULL			
참조 테이블	Member			
참조 컬럼	Mem_id			
CHECK				
데이터 타입	VARCHAR	DATE	NUMBER	NUMBER
길이	20			
비고	FK			

컬럼 설명

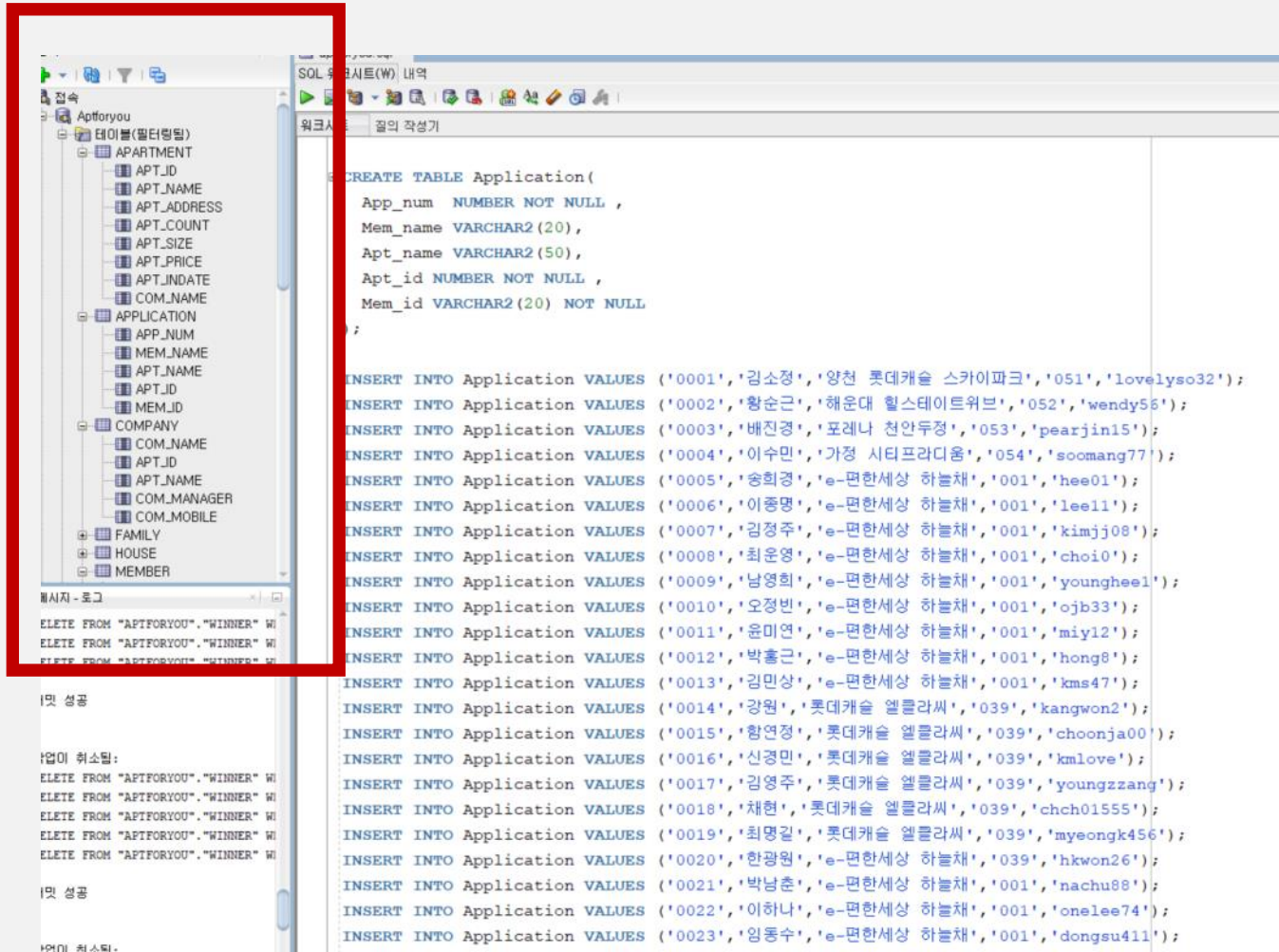
Mem_id	가입자의 ID
Sub_date	청약 계좌를 개설한 날짜
Sub_deposit	청약통장에 현재까지 납입된 금액
Sub_count	청약통장에 돈을 납입한 횟수

Winner 테이블

컬럼명	App_num	Mem_name	Apt_name
PK/UK/NOT NULL	NOT NULL		
참조 테이블	Applicant	Member	Apartment
참조 컬럼	App_num	Mem_name	Apt_name
CHECK			
데이터 타입	NUMBER	VARCHAR	VARCHAR2
길이		20	20
비고	FK		

컬럼 설명

App_num	신청서의 번호
Mem_name	신청자의 이름
Apt_name	신청한 아파트의 이름



PK(Primary key 지정) ▼

```

ALTER TABLE Member
ADD CONSTRAINT Mem_id_pk PRIMARY KEY (Mem_id);

ALTER TABLE Apartment
ADD CONSTRAINT Apt_id_pk PRIMARY KEY (Apt_id);

ALTER TABLE Application
ADD CONSTRAINT App_num_pk PRIMARY KEY (App_num);

ALTER TABLE company
ADD CONSTRAINT com_pk PRIMARY KEY (apt_id);

ALTER TABLE family
ADD CONSTRAINT fam_regnum_pk PRIMARY KEY (fam_regnum);

ALTER TABLE partner
ADD CONSTRAINT par_regnum_pk PRIMARY KEY (par_regnum);

ALTER TABLE subscription
ADD CONSTRAINT Sub_num_pk PRIMARY KEY (sub_num);

```

ERD모델링 - 구현

APP_NUM	MEM_NAME	APT_NAME	APT_ID	MEM_ID
1	김소정	양천 롯데캐슬 스카이라크	51	lovelyso32
2	황순근	해운대 힐스테이트워브	52	wendy56
3	배진경	포레나 천안두정	53	pearjin15
4	이수민	가정 시티프라디움	54	soomang77
5	송희경	e-편한세상 하늘채	1	hee01
6	이종명	e-편한세상 하늘채	1	lee11
7	김정주	e-편한세상 하늘채	1	kimjj08
8	최운영	e-편한세상 하늘채	1	choi0
9	남영희	e-편한세상 하늘채	1	younghee1
10	오정빈	e-편한세상 하늘채	1	ojb33
11	윤미연	e-편한세상 하늘채	1	miyl2
12	박홍근	e-편한세상 하늘채	1	hong8
13	김민상	e-편한세상 하늘채	1	kms47
14	강원	롯데캐슬 엘클라씨	39	kangwon2
15	함원정	롯데캐슬 엘클라씨	39	choonja00
16	신경민	롯데캐슬 엘클라씨	39	kmlove
17	김영주	롯데캐슬 엘클라씨	39	youngzzang
18	채현	롯데캐슬 엘클라씨	39	chch01555
19	최명길	롯데캐슬 엘클라씨	39	myeongk456
20	한광원	e-편한세상 하늘채	39	hkwon26
21	박남춘	e-편한세상 하늘채	1	nachu88
22	이하나	e-편한세상 하늘채	1	onelee74
23	임동수	e-편한세상 하늘채	1	dongsu411
24	안귀옥	e-편한세상 하늘채	1	kikika00
25	김현중	e-편한세상 하늘채	1	hjung55
26	이미성	e-편한세상 하늘채	1	miseong123
27	윤수혁	e-편한세상 하늘채	1	hckck30
28	안성연	e-편한세상 하늘채	1	sysy774
29	문대성	e-편한세상 하늘채	1	moon486
30	정해용	e-편한세상 하늘채	1	heahea00
31	황소희	e-편한세상 하늘채	1	sohee1105
32	김지찬	e-편한세상 하늘채	1	jijich11
33	김희애	e-편한세상 하늘채	1	heey705
34	이장우	e-편한세상 하늘채	1	leejang
35	강래구	e-편한세상 하늘채	1	liver9

```

CREATE TABLE Subscription(
    Sub_num VARCHAR2(20),
    Mem_id VARCHAR2(20) NOT NULL,
    Sub_date DATE,
    Sub_deposit NUMBER,
    Sub_count NUMBER
);

INSERT INTO Subscription VALUES ('S001','hee01','2015-06-11','100000','62');
INSERT INTO Subscription VALUES ('S002','khjyo22','2018-08-21','70000','23');
INSERT INTO Subscription VALUES ('S003','anmoon43','2019-12-13','80000','8');
INSERT INTO Subscription VALUES ('S004','jisub33','2014-02-09','60000','46');
INSERT INTO Subscription VALUES ('S005','mhmh152','2015-01-17','80000','82');
INSERT INTO Subscription VALUES ('S006','hjin256','2016-10-12','40000','115');
INSERT INTO Subscription VALUES ('S007','jin0625','2011-09-04','60000','100');
INSERT INTO Subscription VALUES ('S008','9dragon','2016-08-05','120000','24');
INSERT INTO Subscription VALUES ('S009','sunnyja','2018-01-04','40000','22');
INSERT INTO Subscription VALUES ('S010','smartyoo','2015-07-21','60000','76');
INSERT INTO Subscription VALUES ('S011','thisbottle','2013-01-21','20000','111');
INSERT INTO Subscription VALUES ('S012','10000way','2017-12-10','80000','34');
INSERT INTO Subscription VALUES ('S013','redyou','2019-10-20','100000','46');
INSERT INTO Subscription VALUES ('S014','khj82','2014-12-21','60000','86');
INSERT INTO Subscription VALUES ('S015','watertank','2016-10-10','70000','96');

```

분류	이름	구현 기능
회원	김소정	회원 가입 로그인 / 로그아웃 당첨자 조회 *아이디/비번찾기
	배진경	회원 수정 아파트 정보 검색 아파트 분양 신청
관리자	황순근	아파트 등록 아파트 수정 아파트 삭제 *로그인 / 로그아웃
	이수민	회원 정보 조회 당첨자 선정

점수 배점
100점 만점

무주택 : 20
 신혼부부: 20
 (5년 이내/ 1년당 4점)
 부양가족: 20
 (1명당 5점)
 청약기간: 40
 (1년당 4점)

DB Connection, DAO, VO

Data Access Object

DAO

⋮

DB CONNECT

DataBase 접근을 하기 위한 로직과
비즈니스 로직을 분리

Value Object

VO

⋮

계층간 데이터 교환을 위한 객체

Getter, Setter 객체 호출

DAO

Apartment DAO
Company DAO
Application DAO
Subscription DAO
Member DAO
House DAO
Family DAO
Partner DAO
Winner DAO

VO

Apartment VO
Company VO
Application VO
Subscription VO
Member VO
House VO
Family VO
Partner VO
Winner VO

Part 04 DB Connection, DAO

```
1 package project.apr.foryou;
2
3 import java.sql.Connection;
4
5
6
7 public class DBConnection {
8
9     private Connection con;
10
11     public Connection getConnection() {
12
13         try {
14             String url = "jdbc:oracle:thin:@localhost:1521:xe";
15             Class.forName("oracle.jdbc.driver.OracleDriver");
16             con = DriverManager.getConnection(url, "aprforyou", "aprforyou");
17
18         } catch (Exception e) {
19             System.out.println("연결에러");
20             e.printStackTrace();
21         }
22         return con;
23     }
24 }
25
26
```

DB Connection 객체 호출

```
package project.apr.foryou;

import java.sql.*;

public class ApartmentDAO {
    ArrayList<ApartmentVO> list = new ArrayList<ApartmentVO>();
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    public ArrayList<ApartmentVO> getAllApt() {
        String select = "SELECT * FROM apartment";

        try {
            con = new DBConnection().getConnection();

            Statement stmt = con.createStatement();
            rs = stmt.executeQuery(select);

            while (rs.next()) {
                int id = rs.getInt(1);
                String name = rs.getString(2);
                String addr = rs.getString(3);
                int count = rs.getInt(4);
                String size = rs.getString(5);
                int price = rs.getInt(6);
                String inDate = rs.getString(7);
                String comName = rs.getString(8);

                list.add(new ApartmentVO(id, name, addr, count, size, price, inDate, comName));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return list;
    }
}
```



```
public class MemberDAO {
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    public void memberJoin(MemberVO mvo) { // 회원가입
        PreparedStatement pstmt = null;

        try {
            con = new DBConnection().getConnection();
            String sqlInsert = "INSERT INTO member(Mem_Id, Mem_pw, Mem_Name, Mem_Regnum, Mem_Address, Mem_Mobile) VALUES (?, ?, ?, ?, ?, ?)";
            pstmt = con.prepareStatement(sqlInsert);
            System.out.println("=====");
            ArrayList<MemberVO> list = new ArrayList<MemberVO>();

            pstmt.setString(1, mvo.getMemId());
            pstmt.setString(2, mvo.getMemPw());
            pstmt.setString(3, mvo.getMemName());
            pstmt.setString(4, mvo.getMemRegnum());
            pstmt.setString(5, mvo.getMemAddress());
            pstmt.setString(6, mvo.getMemMobile());

            pstmt.executeUpdate();
            list.add(mvo);

        } catch (SQLException e) {
            System.out.println("##### 가입 실패 #####");
            e.printStackTrace();
        } finally {
            try {
                System.out.println("***** 성공적으로 회원가입 되었습니다 *****");
                if (rs != null)
                    rs.close();
                if (pstmt != null)
                    pstmt.close();
            }
        }
    }
}
```

ToString OVERRIDING ▼

```

1 package project.apr.foryou;
2 //package apt;
3
4 import java.util.Comparator;
5
6 public class ApplicationVO implements Comparator<ApplicationVO> {
7     private int appNum;
8     private String memName;
9     private String aptName;
10    private int aptId;
11    private String memId;
12
13    private int score;
14
15
16    ApplicationVO() {
17    }
18
19    ApplicationVO(int appNum, String memName, int aptId, String aptName, String memId) {
20        this.appNum = appNum;
21        this.memName = memName;
22        this.aptId = aptId;
23        this.aptName = aptName;
24        this.memId = memId;
25    }
26
27    ApplicationVO(String memId, int score, int aptId) {
28        this.memId = memId;
29        this.score = score;
30        this.aptId = aptId;
31    }
32
33    ApplicationVO(String memId, int score) {
34        this.memId = memId;
35        this.score = score;
36    }
37
38

```

```

@Override
public String toString() {
    return "[신청번호] " + appNum + " [이름] " + memName + " [아파트이름] " + aptName + " [아파트ID] " + aptId + " [회원ID] "
        + memId;
}

```

◀ 생성자 호출

```
public class HouseVO {
    String memId;
    String houPdate;
    String houLocation;

    HouseVO(){}

    HouseVO(String memId, String houPdate, String houLocation){
        this.memId = memId;
        this.houPdate = houPdate;
        this.houLocation = houLocation;
    }

    HouseVO(String memId, String houPdate){
        this.memId = memId;
        this.houPdate = houPdate;
    }

    public String getMemId() {
        return memId;
    }
    public void setMemId(String memId) {
        this.memId = memId;
    }
    public String getHouPdate() {
        return houPdate;
    }
    public void setHouPdate(String houPdate) {
        this.houPdate = houPdate;
    }
    public String getHouLocation() {
        return houLocation;
    }
    public void setHouLocation(String houLocation) {
        this.houLocation = houLocation;
    }
}
```

```
public class PartnerVO {

    private String memId;
    private String parName;
    private String parRegnum;
    private String parSex;
    private String parRegdate;

    PartnerVO(){}

    PartnerVO(String memId, String parName, String parSex, String parRegdate){
        this.memId = memId;
        this.parName = parName;
        this.parSex = parSex;
        this.parRegdate = parRegdate;
    }

    PartnerVO(String memId, String parName, String parRegnum, String parSex, String parRegdate){
        this.memId = memId;
        this.parName = parName;
        this.parRegnum = parRegnum;
        this.parSex = parSex;
        this.parRegdate = parRegdate;
    }

    PartnerVO(String memId, String parRegdate){
        this.memId = memId;
        this.parRegdate = parRegdate;
    }

    public String getMemId() {
        return memId;
    }
    public void setMemId(String memId) {
        this.memId = memId;
    }
}
```

after that
프로젝트를 마치며...



“가장 위대한 영광은 ”
한 번도 실패하지 않음이 아니라
실패할 때마다 다시 일어서는 데에 있다

공자

“

나의 코드 9할은
조원분들 덕분이다

- *sojung kom*

”

이 코드는 **내** 덕이다

- *jinkyung pear*

“

아니다 **내** 덕이다

- *wendy Hwang*



조원분들의 노력이 있어서 프로젝트가
완성될 수 있었던 것 같습니다.

늘 겸손한 마음으로 배워 나아가겠습니다.
모든 분들께 진심으로 감사드립니다.

Soom in

THANK YOU

감사합니다