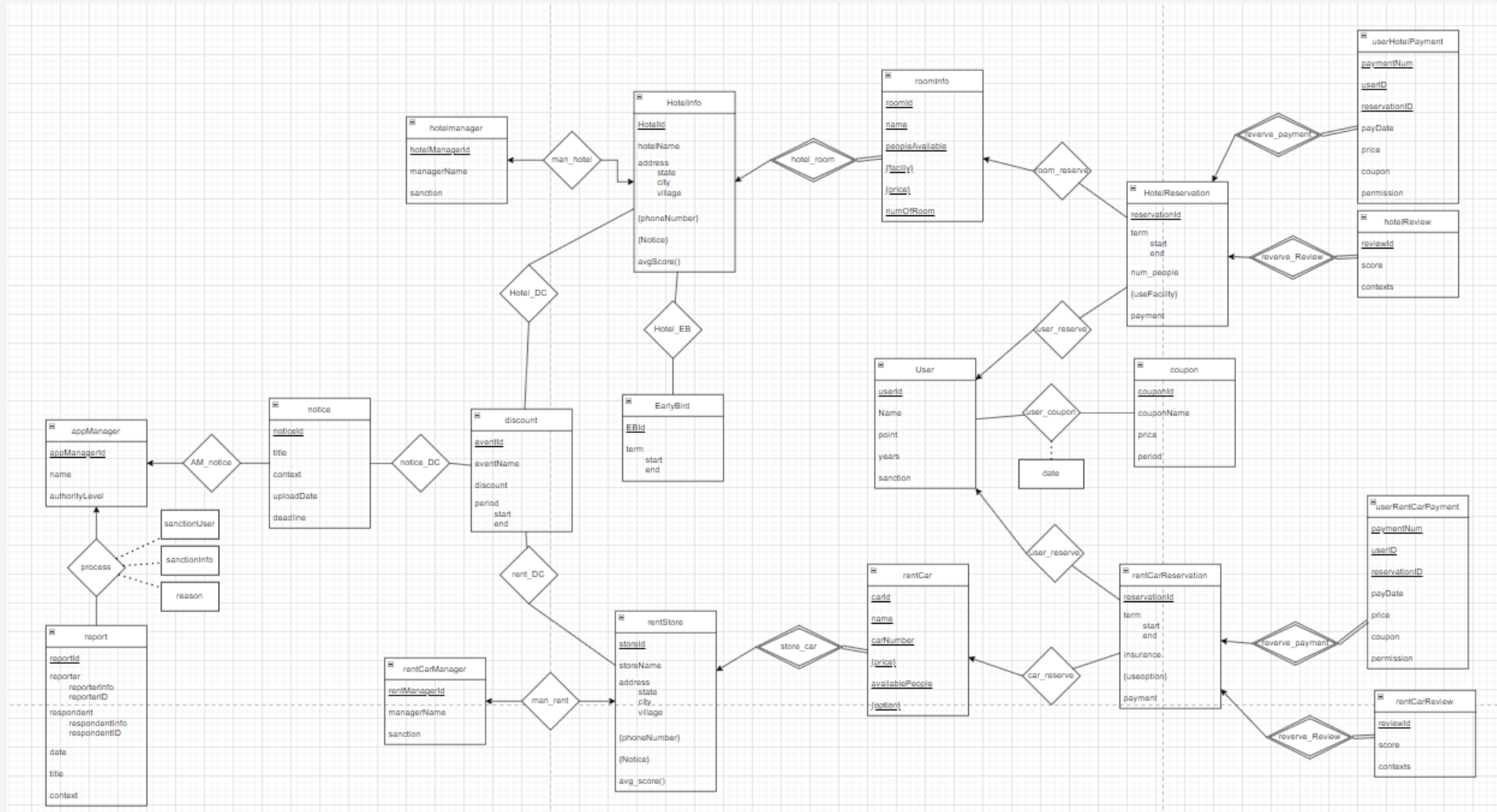


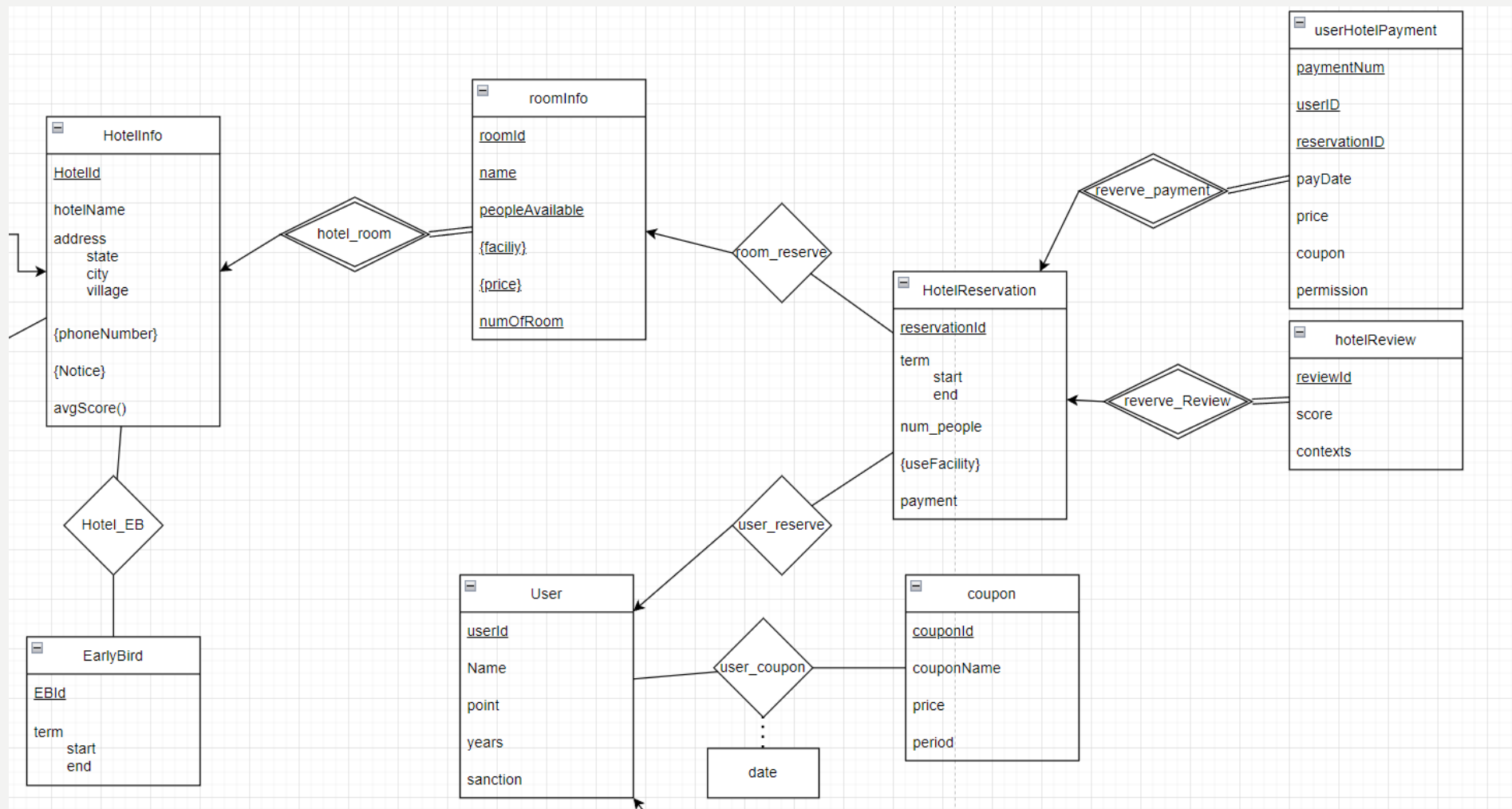
# 숙박 어플리케이션 데이터베이스 설계 및 구현

20170768 이영석

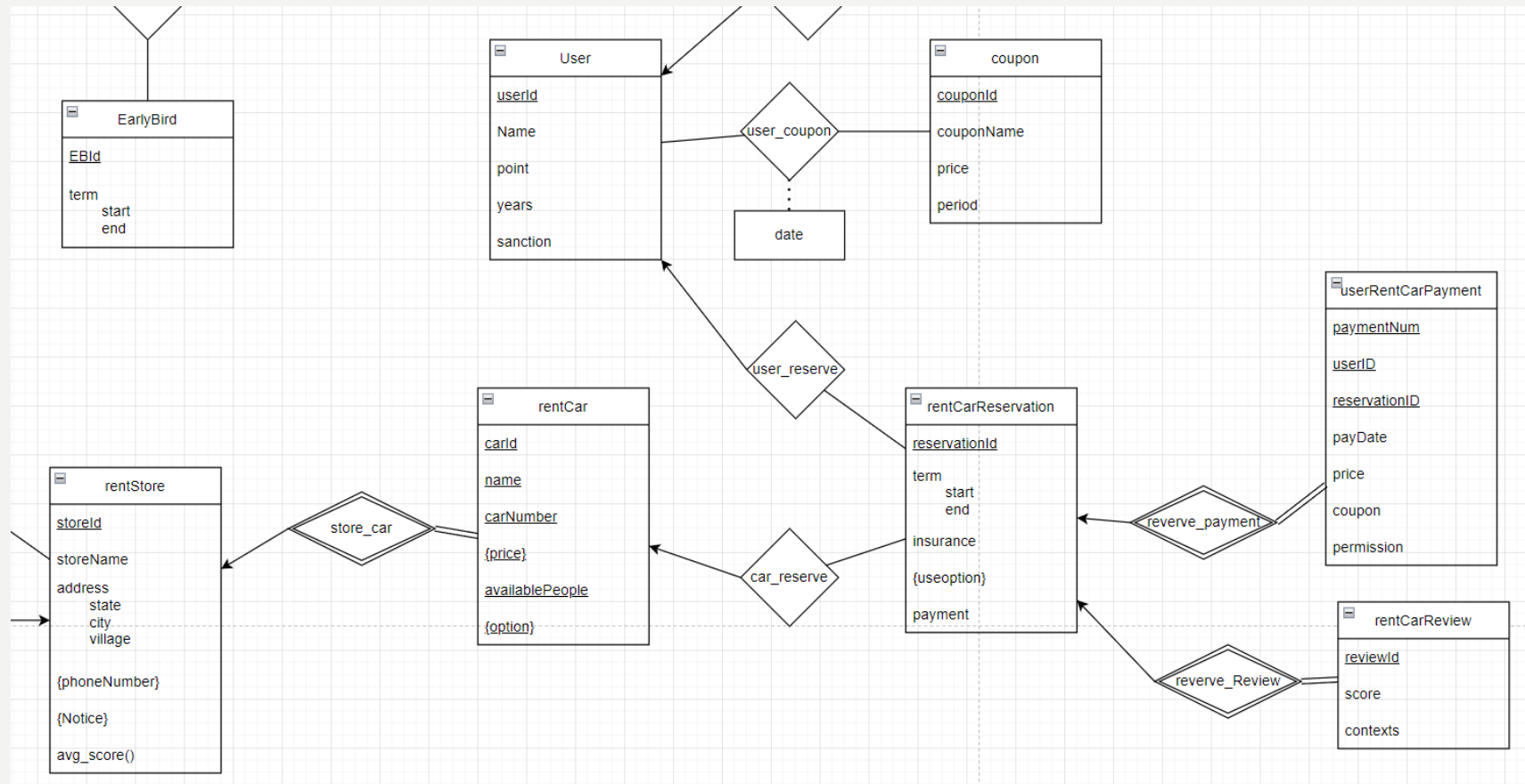
# ER 다이어그램



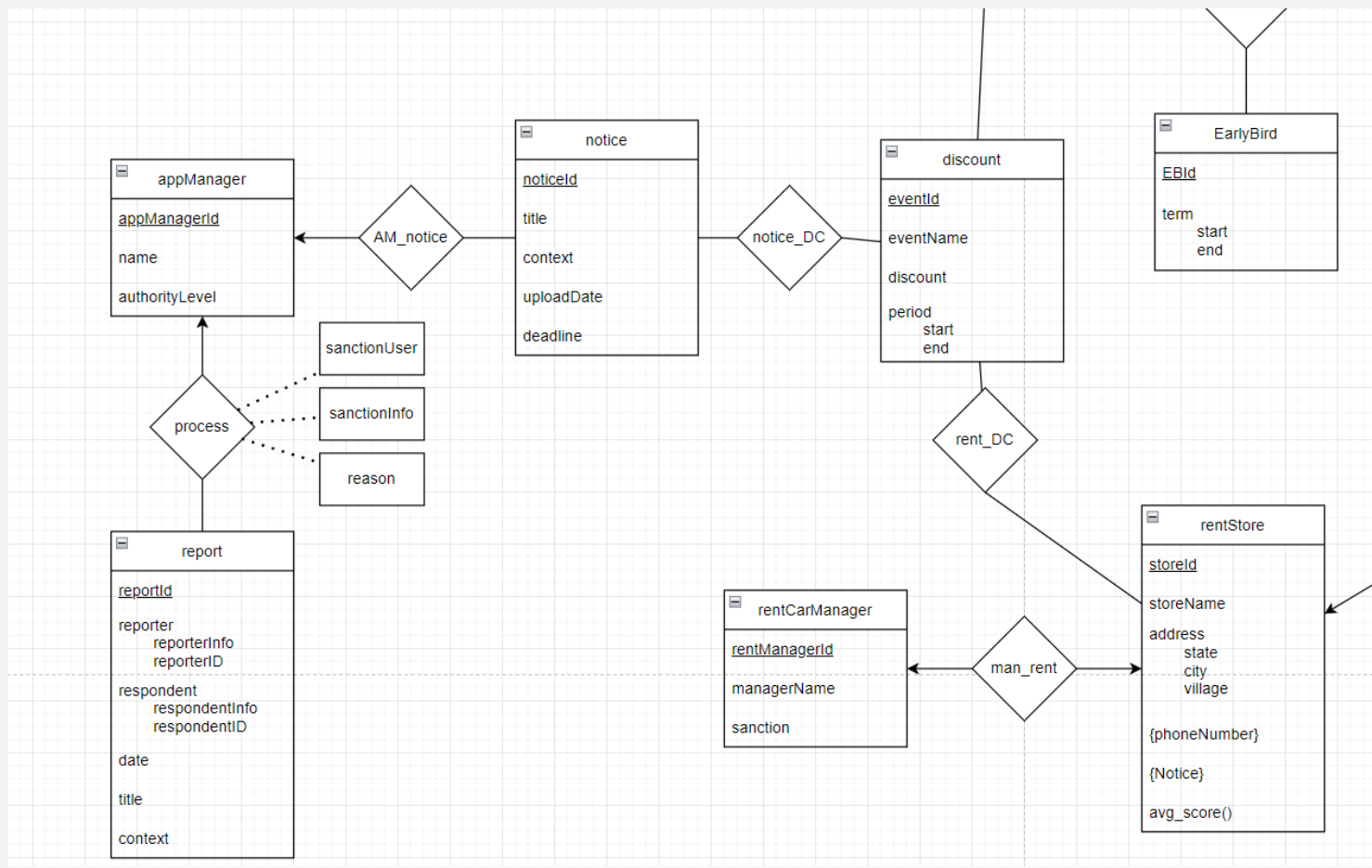
# ER 다이어그램 확대 1



# ER 다이어그램 확대 2



# ER 다이어그램 확대 3



# ER 다이어그램 설명 (ENTITY)

hotelManager – 사용자 중 호텔 관리자의 정보로 id, 이름, 제재 여부등이 저장되어 있다.

hotelInfo - 호텔의 정보로 호텔의 id, 이름, 주소, 전화번호, 공지사항, 평점이 저장되어 있다.

roomInfo – 호텔의 방에 대한 정보로 방 id, 이름, 숙박가능인원수, 시설, 가격, 방 수가 저장되어있다.

hotelReservation – 호텔예약목록의 정보로, 목록 ID, 기간, 숙박인원, 사용시설, 결제여부가 저장되어있다.

userHotelPatment – 결제 목록정보로, 결제번호, 유저id, 예약id, 결제날짜, 결제허가여부 등이 저장되어있다.

hotelReview – 호텔 사용 후 고객 리뷰에 대해 저장되어있다.

EarlyBird – 얼리버드 이벤트 목록이다. id와 기간이 저장되어있다.

user – 사용자 중 고객의 정보로 id, 이름, 포인트, 나이, 제재 여부등이 저장되어있다.

coupon – 쿠폰의 정보로 쿠폰id, 쿠폰이름, 쿠폰가격, 사용기한이 저장되어있다.

# ER 다이어그램 설명 (ENTITY)

rentCarManager – 사용자 중 렌터카 업체 관리자로 id, 관리자이름, 제재 여부가 저장되어있다.

rentStore – 렌터카 업체 정보로, id, 업체이름, 주소, 전화번호, 공지사항, 평점이 저장되어있다.

rentCar – 등록된 렌터카 정보로, 자동차ID, 이름, 차번, 가격, 몇 인승, 옵션에 대해 저장되어있다.

rentCarReservation – 렌터카 예약정보로, 예약 id, 기간, 보험등급, 사용옵션, 결제여부가 저장되어있다.

userRentCarPayment – 렌터카 결제정보로, 결제번호, 유저id, 예약id, 결제날짜, 결제허가여부 등이 저장되어있다.

rentCarReview – 렌터카 사용 후 리뷰로, 리뷰id, 점수, 내용이 저장되어있다.

discount – 할인 이벤트 정보로, 이벤트id, 이름, 할인 룰, 기간이 저장되어있다.

Notice – 전체 어플리케이션 공지사항으로 공지id, 제목, 내용, 업로드 날짜, 데드라인이 저장되어있다.

appManager – 사용자 중 어플리케이션 관리자정보로, id, 이름, 권한 등급이 저장되어있다.

report – 신고에 대한정보로 신고id, 신고자, 피신고자, 날짜, 제목, 내용이 저장되어있다.

# ER 다이어그램 설명 (RELATION)

Man\_hotel – 어떤 관리자가 어떤 호텔을 관리하는지에 대한 관계

Hotel\_room – 어떤 호텔에 어떤 방이 있는지에 대한 관계

Room\_reserve – 어떤 방이 어떤 예약정보를 가지고 있는지에 대한 관계

Hotel\_EB – 호텔이 현재 등록한 얼리버드 이벤트에 대한 정보

Hotel\_DC – 호텔이 현재 등록한 할인 이벤트에 대한 정보

User\_reserve – 사용자와 예약정보 사이의 관계

Reserve\_payment – 예약과 그 결제에 대한 관계

Reserve\_review – 예약과 그 리뷰에 대한 관계

User\_coupon – 각 유저가 가지고 있는 쿠폰에 대한 관계, 획득한 날짜가 추가로 저장되어있다.



# ER 다이어그램 설명 (RELATION)

Man\_rent – 렌터카업체 관리자와 업체 사이의 관계

Store\_car – 렌터카 업체와 보유한 렌터카 사이의 관계

Rent\_DC – 렌터카 업체와 업체가 등록한 할인 이벤트 정보 사이의 관계

Car\_reserve – 렌터카와 그 예약정보 사이의 관계

User\_reserve – 사용자와 예약정보 사이의 관계

Reserve\_payment – 예약과 그 결제에 대한 관계

Reserve\_review – 예약과 그 리뷰에 대한 관계

Notice\_DC – 공지사항과 그에 등록된 할인이벤트에 대한 관계

AM\_notice – 어플리케이션 관리자와 관리자가 등록한 공지사항에 대한 관계

Process – 어플리케이션 관리자와 신고 사이의 관계, 처리에 대한 정보이므로 제재유저, 제재내용, 이유가 추가로 저장되어 있다.

# ERD -> RDB 과정

ERD에서 Entity 사이의 relation이 일대일 일 경우에는 두 엔티티 중 한 곳에 추가로 하나의 속성을 추가하여 다른 Entity의 primary key를 가질 수 있도록 해 주었습니다.

ERD에서 Entity 사이의 relation이 일대다 일 경우에는 두 엔티티 중 다 인 곳에 추가로 하나의 속성을 추가하여 다른 Entity의 primary key를 가질 수 있도록 해 주었습니다.

예를 들면 호텔과 방 사이의 relation이라면 방 정보에 추가로 하나의 속성을 추가하여 호텔의 primary key를 가질 수 있도록 해 주었습니다.

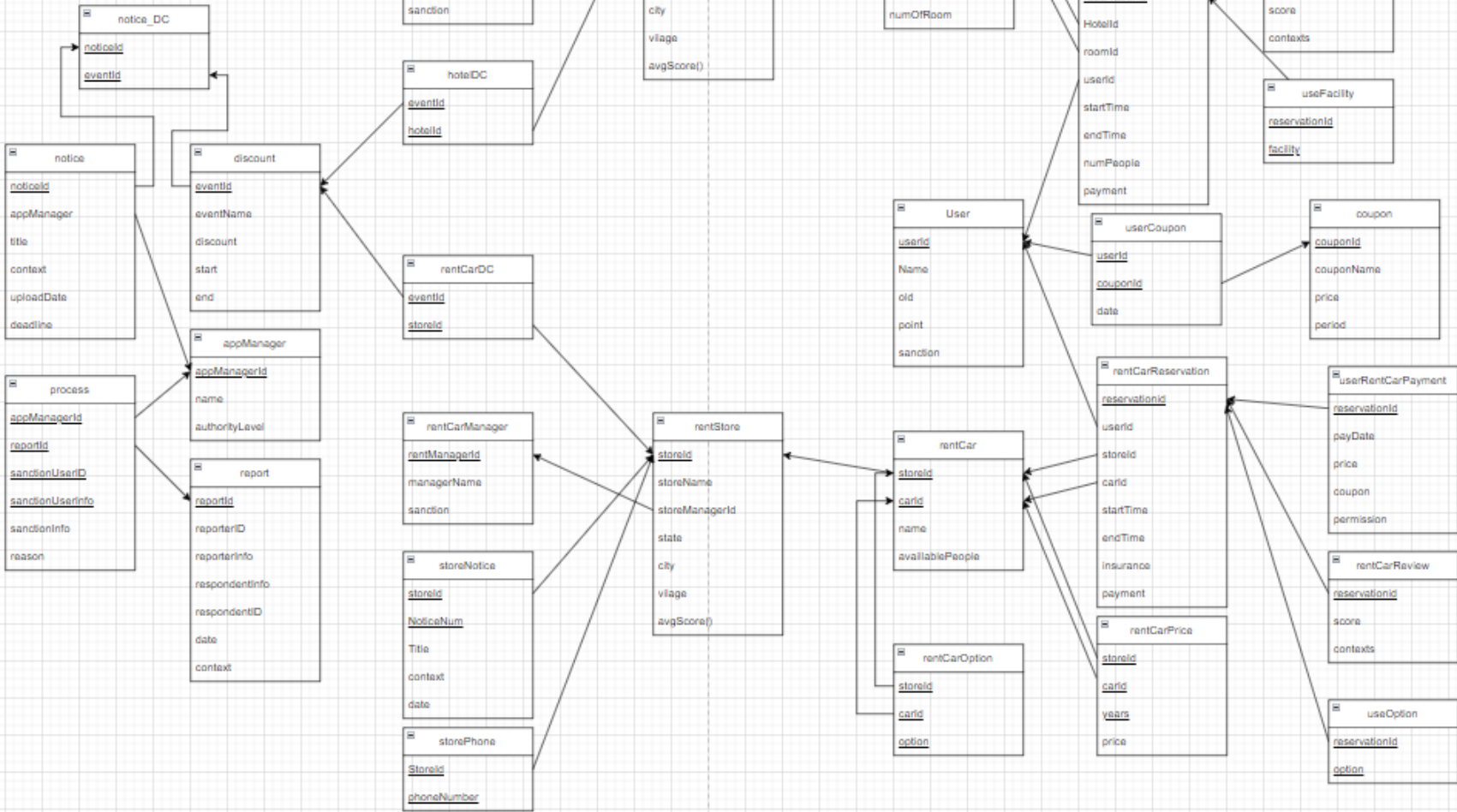
ERD에서 Entity 사이의 relation이 다대다 일 경우에는 두 entity의 primary key를 가지는 테이블을 새로 생성해 주었습니다.

# ERD -> RDB 과정

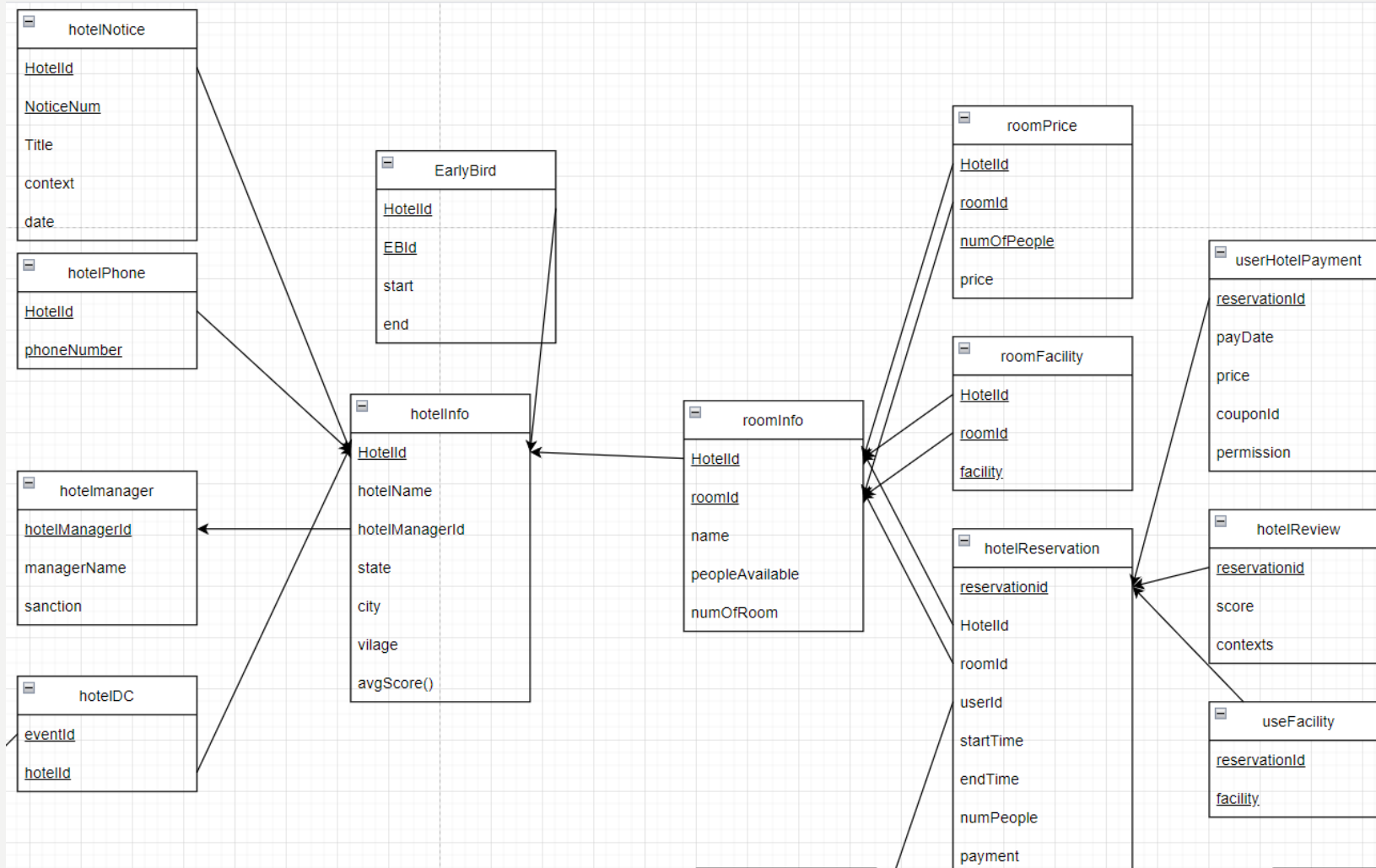
호텔이나 업체의 전화번호, 공지사항, 편의시설 등, 리스트로 나타냈던 속성들은 추가로 테이블을 만들어서 나타내 주었습니다.

# RDB

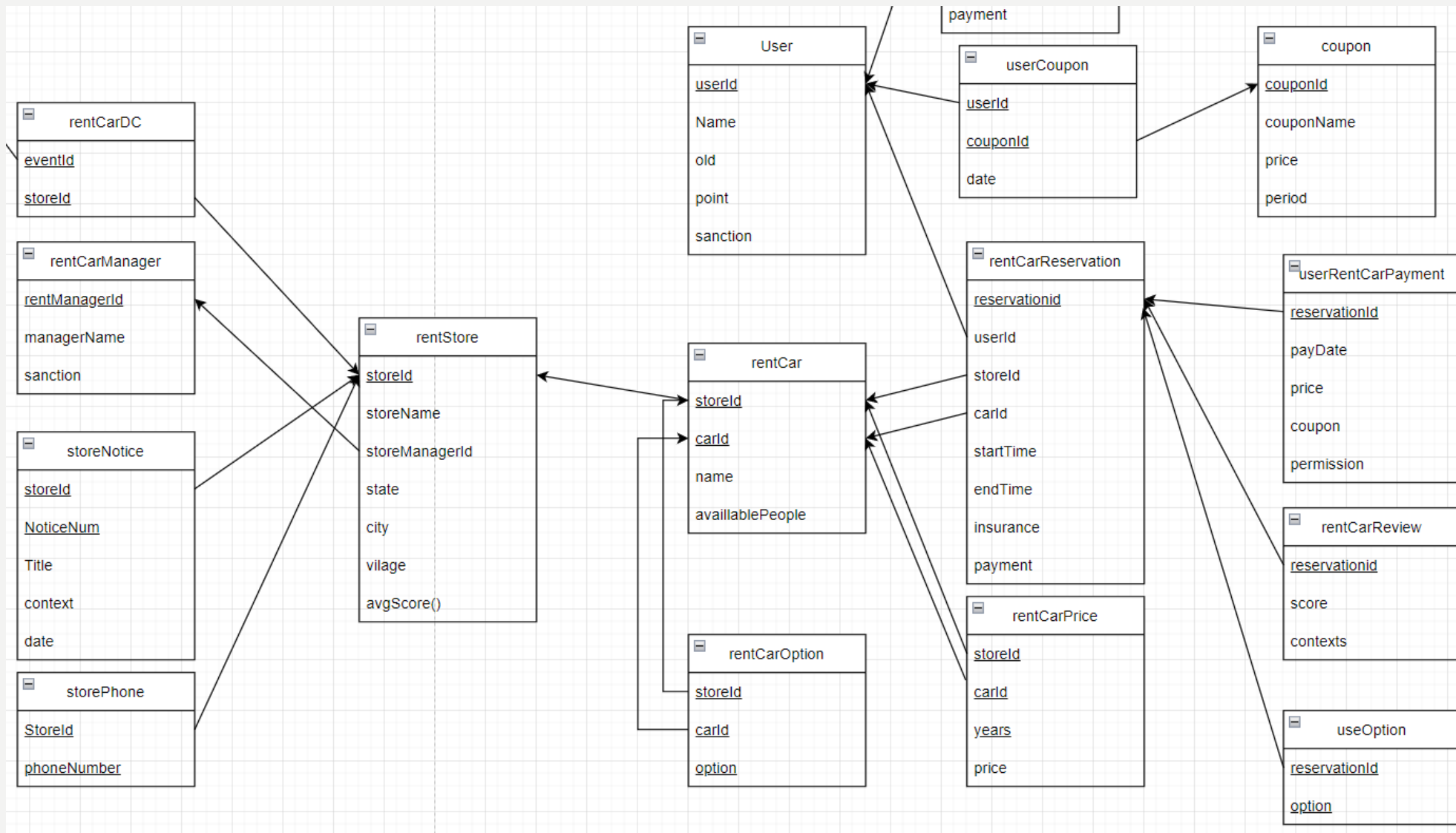
# 다이어그램



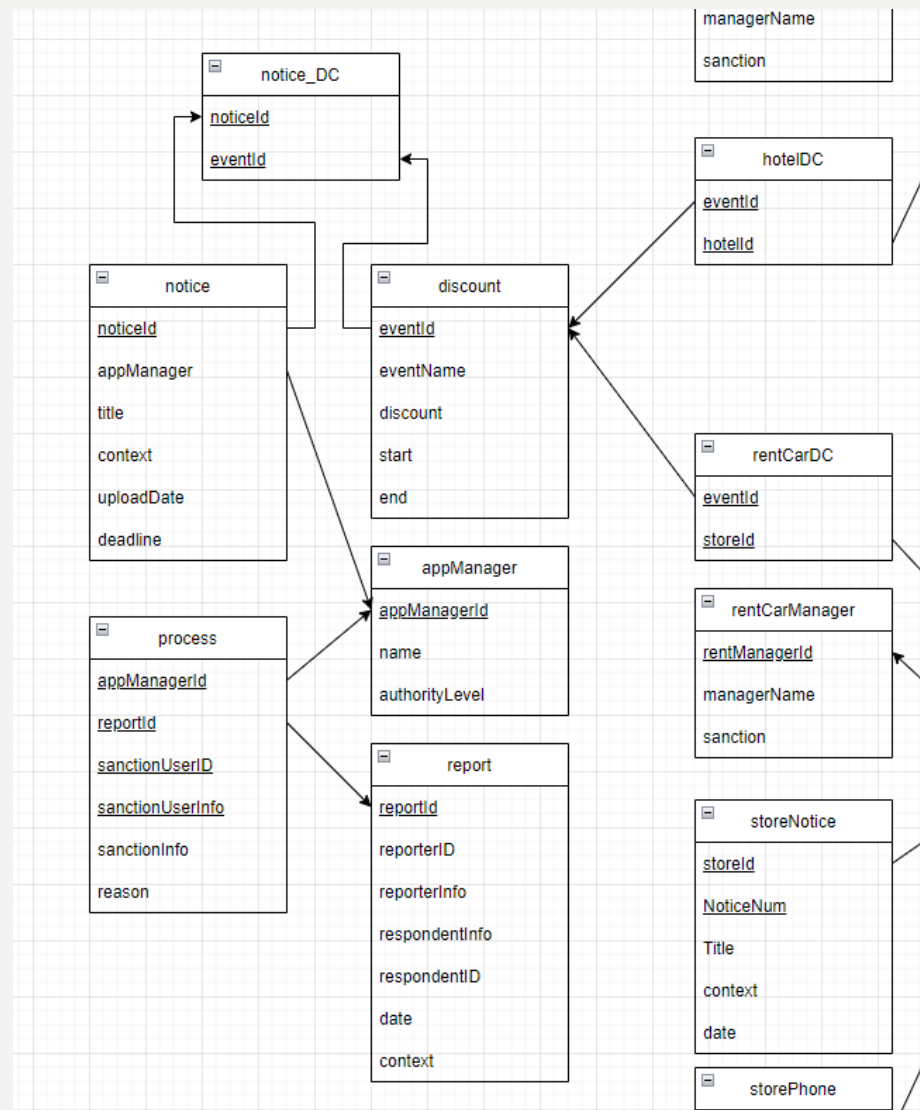
# RDB 다이어그램 확대 1



# RDB 다이어그램 확대 2



# RDB 다이어그램 확대 3



# SQL문

1. ID가 'AA'인 호텔을 예약했던 사람들이 이용하고, 리뷰점수를 평균4점 이상 준 렌터카 업체들의 ID와 업체이름, 평균 리뷰 점수

```
select rentstore.storeId, rentstore.storeName, avg(score)
from rentStore, rentCarReservation, rentCarReview
where rentcarreservation.reservationId = rentcarreview.reservationId and
      rentcarreservation.storeID = rentstore.storeID
      and userId in (select userId
from hotelreservation
where HotelId = 'AA')
group by rentstore.storeId, rentstore.storeName
having avg(score)>=4;
```



# SQL문

2. 서울시 동작구에 있는 호텔 중 4명이 숙박할 수 있는, 오늘부터 내일까지 예약 가능한 방이 있는 호텔들의 HotelId와 이름

```
select distinct hotelinfo.hotelId, hotelName
from hotelinfo, roominfo
where state = '서울시' and city = '동작구'
      and hotelinfo.hotelId = roominfo.hotelId and peopleAvailable >= 4 and
      (roominfo.hotelId, roominfo.roomId) not in (
          select hotelId, roomId
          from hotelreservation
          where startTime <= STR_TO_DATE(20211224, '%Y%m%d')
              and endTime >= STR_TO_DATE(20211225, '%Y%m%d')
      );
```

# SQL문

3. 서울시의 '중앙호텔'이 현재 참여하고 있는 할인 이벤트를 알려주는 전체 공지사항의 제목과 내용

```
select title, context
```

```
from notice
```

```
where noticeId in (select noticeId
```

```
    from notice_DC, discount
```

```
    where notice_DC.eventId = discount.eventId start <= now() and end >= now() and
```

```
        eventId in (select eventId
```

```
            from HotelInfo, hotelDC
```

```
            where HotelInfo.HotelId = HotelDC.HotelId and hotelName = '중앙호텔'
```

```
            and state = '서울시')
```

```
);
```

# SQL문

4. 'BB'라는 이름의 쿠폰을 사용해서 숙박을 예약한 적 있는 호텔ID와 호텔 당 사용한 쿠폰 수

```
Select hotelId, COUNT(hotelId)
```

```
From hotelReservation
```

```
Where reservationId in (select reservationId
```

```
    from userHotelPayment
```

```
    where couponId in (select couponId
```

```
        from coupon
```

```
        where couponName = 'BB')
```

```
)
```

```
group by HotelId;
```

# SQL문

5. '서울시'의 'AA'호텔에 3점 이하를 준 유저 중에 제제를 당한적이 있는 유저id

```
select distinct sanctionUserId
```

```
from process
```

```
where sanctionUserInfo = 'user' and sanctionUserId in (select userid
```

```
from hotelReservation, hotelReview
```

```
where hotelReservation.reservationId = hotelReview.reservationId and score<=3 and
```

```
hotelId in (select hotelId
```

```
from hotelInfo
```

```
where hotelname = 'AA호텔' and state='서울시')
```

```
);
```



# CREATE TABLE

select \* from hotelinfo;

	hotelId	hotelName	hotelManagerId	state	city	village
▶	aaaa1	AA 호텔	managerA	서울시	동작구	상도동
	aaaa2	AB 호텔	managerAB	서울시	동작구	상도동
	aassaa	ruru 호텔	manager1	대구시	중구	삼덕동
	abcabc	ABC 호텔	koreantop	서울시	동작구	상도동
	abcde	Flower 호텔	myosotis	서울시	동작구	상도동
	bbbb1	캐슬링	castling	서울시	동작구	흑석동
	bbbb2	QWERTY 호텔	qwerty	경기도	수원시	메탄동
	bbbb3	LOL 호텔	Darius	충청남도	아산시	배방읍
	qweqwe	프리티	manager2	충청남도	천안시	불당동
	zgqsdw	freedom	manager3	경기도	성남시	북정동
✱	NULL	NULL	NULL	NULL	NULL	NULL

# CREATE TABLE

```
select * from roominfo;
```

	hotelId	roomId	roomName	peopleAvailable	numOfRoom
▶	aaaa1	101	별방	4	2
	aaaa1	102	달방	2	2
	aaaa2	101	101호	4	3
	aaaa2	102	102호	2	1
	abccbc	101	honoka	4	2
	abccbc	102	kotori	2	1
	abcde	101	물망초방	4	2
	abcde	102	달맞이꽃방	2	1
	bbbb1	101	폰방	4	2
	bbbb1	102	룩방	4	2
●	NULL	NULL	NULL	NULL	NULL

# CREATE TABLE

- `select * from rentStore;`

	storeId	storeName	storeManagerId	state	city	village
▶	rent1	마세라티렌트카	rentmaster	서울시	동작구	상도동
	rent10	HH렌트카	HHrent	서울시	영등포구	여의도동
	rent2	해피렌트카	happyrent	서울시	동작구	흑석동
	rent3	AA렌트카	AArent	서울시	강남구	논현동
	rent4	BB렌트카	BBrent	서울시	관악구	미성동
	rent5	CC렌트카	CCrent	충청남도	천안시	불당동
	rent6	DD렌트카	DDrent	충청남도	아산시	탕정면
	rent7	EE렌트카	EErent	대구시	서구	봉덕동
	rent8	FF렌트카	FFrent	서울시	마포구	합정동
	rent9	GG렌트카	GGrent	서울시	송파구	잠실동
●	NULL	NULL	NULL	NULL	NULL	NULL





# CREATE TABLE

- `select * from rentcarReview;`

	reservationId	score	contexts
▶	r1	3	차가 좋아요
	r10	3	차가 좋아요1010
	r2	4	차가 좋아요22
	r3	5	차가 좋아요33
	r4	4	차가 좋아요44
	r5	4	차가 좋아요55
	r6	5	차가 좋아요66
	r7	3	차가 좋아요77
	r8	4	차가 좋아요88
	r9	5	차가 좋아요99
✱	NULL	NULL	NULL

# JDBC/MYSQL 프로그램

기능메뉴:

1. 예약가능한 호텔을 찾기,
2. 내가 이용할 호텔에서 이용할 만한 렌터카업체 (이 호텔을 이용한사람들이 평가한 평점)
3. 종료

1번 선택 시, 숙박 시작 날짜, 마지막 날짜, 숙박 지역, 숙박 인원을 입력 받아서 해당 지역에 해당 인원이 같이 쓸 수 있는 방이 있는지, 또 그 때에 예약이 이미 되어있는 것은 아닌지 확인한다.

2번 선택 시, 숙박하는 호텔 ID와 최소 평점을 입력 받아서, 이전에 이 호텔을 사용했던 사람들이 평가한 렌터카 업체들의 평점이 해당 점수 이상이라면 출력해준다.

# 프로그램의 SQL 문1

```
sc.nextLine();
try{
    Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost/hotel_db?useUnicode=true&useJDBCCompliantTimezoneShift=true
        + "&useLegacyDatetimeCode=false&serverTimezone=UTC","root","lys1823");
    Statement stmt = conn.createStatement();
    String sql = "select distinct hotelinfo.hotelId, hotelName "
        + "from hotelinfo, roominfo "
        + "where state = ? and city = ? and village = ? "
        + "and hotelinfo.hotelId = roominfo.hotelId and peopleAvailable>=? and "
        + "(roominfo.hotelId, roominfo.roomId) not in ("
        + "    select hotelId, roomId "
        + "    from hotelreservation "
        + "    where startTime<=STR_TO_DATE(?, '%Y%m%d') and endTime>=STR_TO_DATE(?, '%Y%m%d'))";
    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setString(1,state);
    pstmt.setString(2,city);
    pstmt.setString(3,village);
    pstmt.setInt(4, peopleNum);
    pstmt.setString(5,start);
    pstmt.setString(6,end);

    ResultSet rs = pstmt.executeQuery();
```

목표한 예약되어 있지 않은 방을 가진 호텔을 구하기 위해서, 해당 지역의 숙박가능 인원수 조건에 맞는 호텔, 방에서 이미 예약된 호텔의 방을 제외하는 식으로 계산했습니다. 그 후 중복 없이 호텔 ID와 이름만 출력하는 것으로 해결했습니다.

# 프로그램의 SQL 문2

```
int score = sc.nextInt();
sc.nextLine();
try{
    Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost/hotel_db?useUnicode=true&useJDBCCompliantTimezoneShift=true
        + "&useLegacyDatetimeCode=false&serverTimezone=UTC", "root", "lys1823");
    Statement stmt = conn.createStatement();
    String sql = "select rentstore.storeId, rentstore.storeName, avg(score)\r\n"
        + "from rentStore, rentCarReservation, rentCarReview\r\n"
        + "where rentcarreservation.reservationId = rentcarreview.reservationId and \r\n"
        + "rentcarreservation.storeID = rentstore.storeID\r\n"
        + "and userId in (select userId\r\n"
        + "from hotelreservation\r\n"
        + "where HotelId = ?)\r\n"
        + "group by rentstore.storeId, rentstore.storeName\r\n"
        + "having avg(score)>=?;";
    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, hotelId);
    pstmt.setInt(2, score);

    ResultSet rs = pstmt.executeQuery();
```

지정한 호텔의 사용자들을 구하고, 그 유저들이 예약했던 렌터카 업체의 예약과 리뷰들을 group by로 업체 단위로 묶어서 그 리뷰 점수들의 평균 점수를 구했습니다. 해당 호텔을 이용한 사람들이 이용하기 어땠는지 알아보기 쉽게 구했습니다.

# 프로그램 실행 및 결과

상도동의 호텔은 총 4개로 모두 최소 3인이 숙박  
할 수 있지만,

24일에서 25일 사이에 3명이 숙박해야 하므  
로 aaaa1 호텔은 이미 4인실인 101이 예약되  
어 있으므로 사용할 수 없습니다.

hotelId	roomId	roomName	peopleAvailable	numOfRoom
aaaa1	101	별방	4	2
aaaa1	102	달방	2	2
aaaa2	101	101호	4	3
aaaa2	102	102호	2	1
abcabc	101	honoka	4	2
abcabc	102	kotori	2	1
abcde	101	물망초방	4	2
abcde	102	달맞이꽃방	2	1
bbbb1	101	폰방	4	2
bbbb1	102	룩방	4	2
NULL	NULL	NULL	NULL	NULL

Main [Java Application] C:\Users\Waass0\p2\pool\plugins\org.eclipse.justj.openjdk.h

1. 예약가능한 호텔을 찾기,
2. 내가 이용할 호텔에서 이용할 만한 렌터카업체(이 호텔을 이용한사람들이 평가한 평점)
3. 종료

1

숙박 시작날짜를 입력하세요 ex)20211224

20211224

숙박 마지막날짜를 입력하세요 ex)20211225

20211225

숙박 지역을 입력하세요 ex)서울시, 경기도, 충청남도 등

서울시

숙박 지역을 입력하세요 ex)동작구, 수원시, 천안시 등

동작구

숙박 지역을 입력하세요 ex)상도동, 메탄동, 두정동 등

상도동

숙박 인원 수를 입력하세요

3

Hotel ID = aaaa2, Hotel name = AB호텔

Hotel ID = abcabc, Hotel name = ABC호텔

Hotel ID = abcde, Hotel name = Flower호텔

1. 예약가능한 호텔을 찾기,

reservationId	hotelId	roomId	userId	startTime	endTime	numPeople	payment
1	aaaa1	101	user1	2021-12-24	2021-12-26	3	1
2	aaaa1	101	user2	2021-12-20	2021-12-23	2	1
3	aaaa2	101	user3	2021-12-20	2021-12-23	4	1
4	aaaa2	102	user4	2021-12-24	2021-12-25	2	1
5	abcabc	101	user5	2021-12-19	2021-12-24	4	1
6	abcabc	101	user6	2021-12-17	2021-12-19	3	1
7	abcde	101	user7	2021-12-19	2021-12-20	3	1
8	abcde	101	user8	2021-12-22	2021-12-23	3	1
9	abcde	101	user9	2021-12-26	2021-12-27	4	1
10	abcde	102	user10	2021-12-20	2021-12-22	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

aaaa1	AA호텔	managerA	서울시	동작구	상도동
aaaa2	AB호텔	managerAB	서울시	동작구	상도동
abcabc	ABC호텔	koreantop	서울시	동작구	상도동
abcde	Flower호텔	myosotis	서울시	동작구	상도동

# 프로그램 실행 및 결과

11월 25일부터 11월 26일까지는 예약이 된  
방이 없고 모두 4인실을 보유하고 있으므로

상도동의 모든 호텔이 검색됩니다.

hotelId	roomId	roomName	peopleAvailable	numOfRoom
aaaa1	101	별방	4	2
aaaa1	102	달방	2	2
aaaa2	101	101호	4	3
aaaa2	102	102호	2	1
abcabc	101	honoka	4	2
abcabc	102	kotori	2	1
abcde	101	물망초방	4	2
abcde	102	달맞이꽃방	2	1
bbbb1	101	폰방	4	2
bbbb1	102	룩방	4	2
NULL	NULL	NULL	NULL	NULL

1. 예약가능한 호텔을 찾기,
2. 내가 이용할 호텔에서 이용할 만한 렌터카업체 (이 호텔을 이용한 사람들이 평가한 평점)
3. 종료

1

숙박 시작날짜를 입력하세요 ex) 20211224

20211125

숙박 마지막날짜를 입력하세요 ex) 20211225

20211126

숙박 지역을 입력하세요 ex) 서울시, 경기도, 충청남도 등

서울시

숙박 지역을 입력하세요 ex) 동작구, 수원시, 천안시 등

동작구

숙박 지역을 입력하세요 ex) 상도동, 메탄동, 두정동 등

상도동

숙박 인원 수를 입력하세요

4

Hotel ID = aaaa1, Hotel name = AA호텔

Hotel ID = aaaa2, Hotel name = AB호텔

Hotel ID = abcabc, Hotel name = ABC호텔

Hotel ID = abcde, Hotel name = Flower호텔

reservationId	hotelId	roomId	userId	startTime	endTime	numPeople	payment
1	aaaa1	101	user1	2021-12-24	2021-12-26	3	1
2	aaaa1	101	user2	2021-12-20	2021-12-23	2	1
3	aaaa2	101	user3	2021-12-20	2021-12-23	4	1
4	aaaa2	102	user4	2021-12-24	2021-12-25	2	1
5	abcabc	101	user5	2021-12-19	2021-12-24	4	1
6	abcabc	101	user6	2021-12-17	2021-12-19	3	1
7	abcde	101	user7	2021-12-19	2021-12-20	3	1
8	abcde	101	user8	2021-12-22	2021-12-23	3	1
9	abcde	101	user9	2021-12-26	2021-12-27	4	1
10	abcde	102	user10	2021-12-20	2021-12-22	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

aaaa1	AA호텔	managerA	서울시	동작구	상도동
aaaa2	AB호텔	managerAB	서울시	동작구	상도동
abcabc	ABC호텔	koreantop	서울시	동작구	상도동
abcde	Flower호텔	myosotis	서울시	동작구	상도동

# 프로그램 실행 및 결과

12월 21일과 12월 23일까지 3인 예약을 하려고 했으나 abcde 호텔의 물망초방 말고는 모두 4인실이 예약되어 있어 사용할 수 없습니다.

hotelId	roomId	roomName	peopleAvailable	numOfRoom
aaaa1	101	별방	4	2
aaaa1	102	달방	2	2
aaaa2	101	101호	4	3
aaaa2	102	102호	2	1
abcabc	101	honoka	4	2
abcabc	102	kotori	2	1
abcde	101	물망초방	4	2
abcde	102	달맞이꽃방	2	1
bbbb1	101	폰방	4	2
bbbb1	102	룩방	4	2
NULL	NULL	NULL	NULL	NULL

1. 예약가능한 호텔을 찾기,
2. 내가 이용할 호텔에서 이용할 만한 렌터카업체(이 호텔을 이용한사람들이 평가한 평점)
3. 종료

1

숙박 시작날짜를 입력하세요 ex)20211224

20211221

숙박 마지막날짜를 입력하세요 ex)20211225

20211223

숙박 지역을 입력하세요 ex)서울시, 경기도, 충청남도 등

서울시

숙박 지역을 입력하세요 ex)동작구, 수원시, 천안시 등

동작구

숙박 지역을 입력하세요 ex)상도동, 메탄동, 두정동 등

상도동

숙박 인원 수를 입력하세요

3

Hotel ID = abcde, Hotel name = Flower호텔

4. 예약가능한 호텔을 찾기

reservationId	hotelId	roomId	userId	startTime	endTime	numPeople	payment
1	aaaa1	101	user1	2021-12-24	2021-12-26	3	1
2	aaaa1	101	user2	2021-12-20	2021-12-23	2	1
3	aaaa2	101	user3	2021-12-20	2021-12-23	4	1
4	aaaa2	102	user4	2021-12-24	2021-12-25	2	1
5	abcabc	101	user5	2021-12-19	2021-12-24	4	1
6	abcabc	101	user6	2021-12-17	2021-12-19	3	1
7	abcde	101	user7	2021-12-19	2021-12-20	3	1
8	abcde	101	user8	2021-12-22	2021-12-23	3	1
9	abcde	101	user9	2021-12-26	2021-12-27	4	1
10	abcde	102	user10	2021-12-20	2021-12-22	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

aaaa1	AA호텔	managerA	서울시	동작구	상도동
aaaa2	AB호텔	managerAB	서울시	동작구	상도동
abcabc	ABC호텔	koreantop	서울시	동작구	상도동
abcde	Flower 호텔	myosotis	서울시	동작구	상도동



# 프로그램 실행 및 결과

Abcabc 호텔을 이용한 사람은 user5,6 이고,  
이 사람들이 사용한 업체는 rent3 하나 뿐이고  
평점을 4점, 4점 주었으므로 평점은 4점이 나옵니다

1. 예약가능한 호텔을 찾기,
2. 내가 이용할 호텔에서 이용할 만한 렌터카업체 (이 호텔을 이용한사람들이 평가한 평점)
3. 종료

2

숙박하는호텔ID를 입력하세요 ex)aaaa1

abcabc

원하는 최소평균평점을 입력하세요 ex) 4

4

```
store ID = rent3, store name = AA렌트카, 평점 4.5
```

	storeId	storeName	storeManagerId	state	city	village
▶	rent1	마세라티렌트카	rentmaster	서울시	동작구	상도동
	rent10	HH렌트카	HHrent	서울시	영등포구	여의도동
	rent2	해피렌트카	happyrent	서울시	동작구	흑석동
	rent3	AA렌트카	AArent	서울시	강남구	논현동
	rent4	BB렌트카	BBrent	서울시	관악구	미성동
	rent5	CC렌트카	CCrent	충청남도	천안시	불당동
	rent6	DD렌트카	DDrent	충청남도	아산시	탕정면
	rent7	EE렌트카	EErent	대구시	서구	봉곡동
	rent8	FF렌트카	FFrent	서울시	마포구	합정동
	rent9	GG렌트카	GGrent	서울시	송파구	잠실동
▶	NULL	NULL	NULL	NULL	NULL	NULL

[illegible]

# 프로그램 실행 및 결과

Abcde를 이용한 고객은 user 7 8 9 10이고  
 사용한 업체와 평점은

Rent4 3점, 4점

## Rent5 5점, 3점이므로

각각 평점은 3.5점, 4점으로 나오게 됩니다.

숙박하는호텔ID를 입력하세요 ex)aaaa1

abcde

원하는 최소평균평점을 입력하세요 ex) 4

3

store ID = rent5, store name = CC렌트카, 평점 4.0

```
store ID = rent4, store name = BB렌트카, 평점 3.5
```

	storeId	storeName	storeManagerId	state	city	village
▶	rent1	마세라티렌트카	rentmaster	서울시	동작구	상도동
	rent10	HH렌트카	HHrent	서울시	영등포구	여의도동
	rent2	해피렌트카	happyrent	서울시	동작구	죽석동
	rent3	AA렌트카	AArent	서울시	강남구	논현동
	rent4	BB렌트카	BBrent	서울시	관악구	미성동
	rent5	CC렌트카	CCrent	충청남도	천안시	불당동
	rent6	DD렌트카	DDrent	충청남도	아산시	탕정면
	rent7	EE렌트카	EErent	대구시	서구	봉덕동
	rent8	FF렌트카	FFrent	서울시	마포구	합정동
	rent9	GG렌트카	GGrent	서울시	송파구	잠실동
*	NULL	NULL	NULL	NULL	NULL	NULL

[illegible]

# 프로그램 실행 및 결과

만약 최소 평점을 4로 선택하면

앞의 상황에서 3.5 평점을 가졌던 rent4는 사라집니다

1. 예약가능한 호텔을 찾기,
2. 내가 이용할 호텔에서 이용할 만한 렌터카업체 (이 호텔을 이용한사람들이 평가한 평점)
3. 종료

2 숙박하는호텔ID를 입력하세요 ex)aaaa1

원하는 최소평균평점을 입력하세요 ex) 4

```
4 store ID = rent5, store name = CC렌트카, 평점 4.0
```

	storeId	storeName	storeManagerId	state	city	village
▶	rent1	마세라티렌트카	rentmaster	서울시	동작구	상도동
	rent10	HH렌트카	HHrent	서울시	영등포구	여의도동
	rent2	해피렌트카	happyrent	서울시	동작구	흑석동
	rent3	AA렌트카	AArent	서울시	강남구	논현동
	rent4	BB렌트카	BBrent	서울시	관악구	미성동
	rent5	CC렌트카	CCrent	충청남도	천안시	불당동
	rent6	DD렌트카	DDrent	충청남도	아산시	탕정면
	rent7	EE렌트카	EErent	대구시	서구	봉덕동
	rent8	FF렌트카	FFrent	서울시	마포구	합정동
	rent9	GG렌트카	GGrent	서울시	송파구	잠실동
▶	NULL	NULL	NULL	NULL	NULL	NULL

[illegible]

# 함수 종속성

지역을 표기할 때, state, city, village 가 함수 종속성이 있을 수 있다고 생각 했습니다.

하지만, 지역 이름이 겹치는 경우가 있습니다. 우리나라에 광주시는 2개 있고 동, 면, 리 등은 겹치는 이름이 매우 많습니다. 따라서 함수 종속성이 있을 수 없습니다.

그 외에 제작한 RDB에서 고유식별자를 제외하고 함수 종속성은 보이지 않았습니다.

따라서 스키마 정제가 따로 필요하지 않습니다.

만약 함수 종속성이 추가로 발생하였다면 테이블을 나누어서 BCNF로 만들어 중복되는 요소를 없애야 합니다.