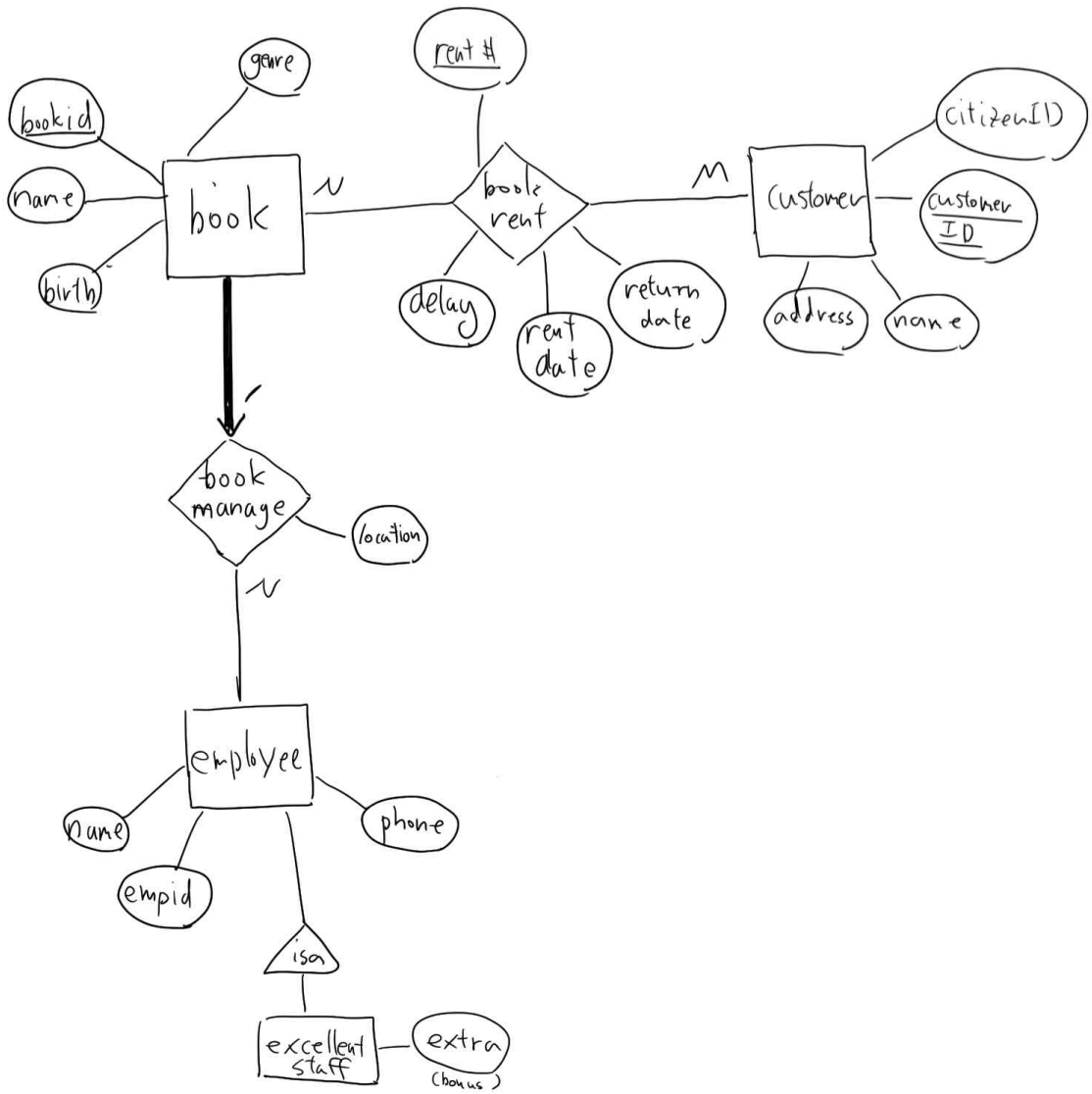
과제 2 B511226 홍현승

E-R diagram 변동사항!



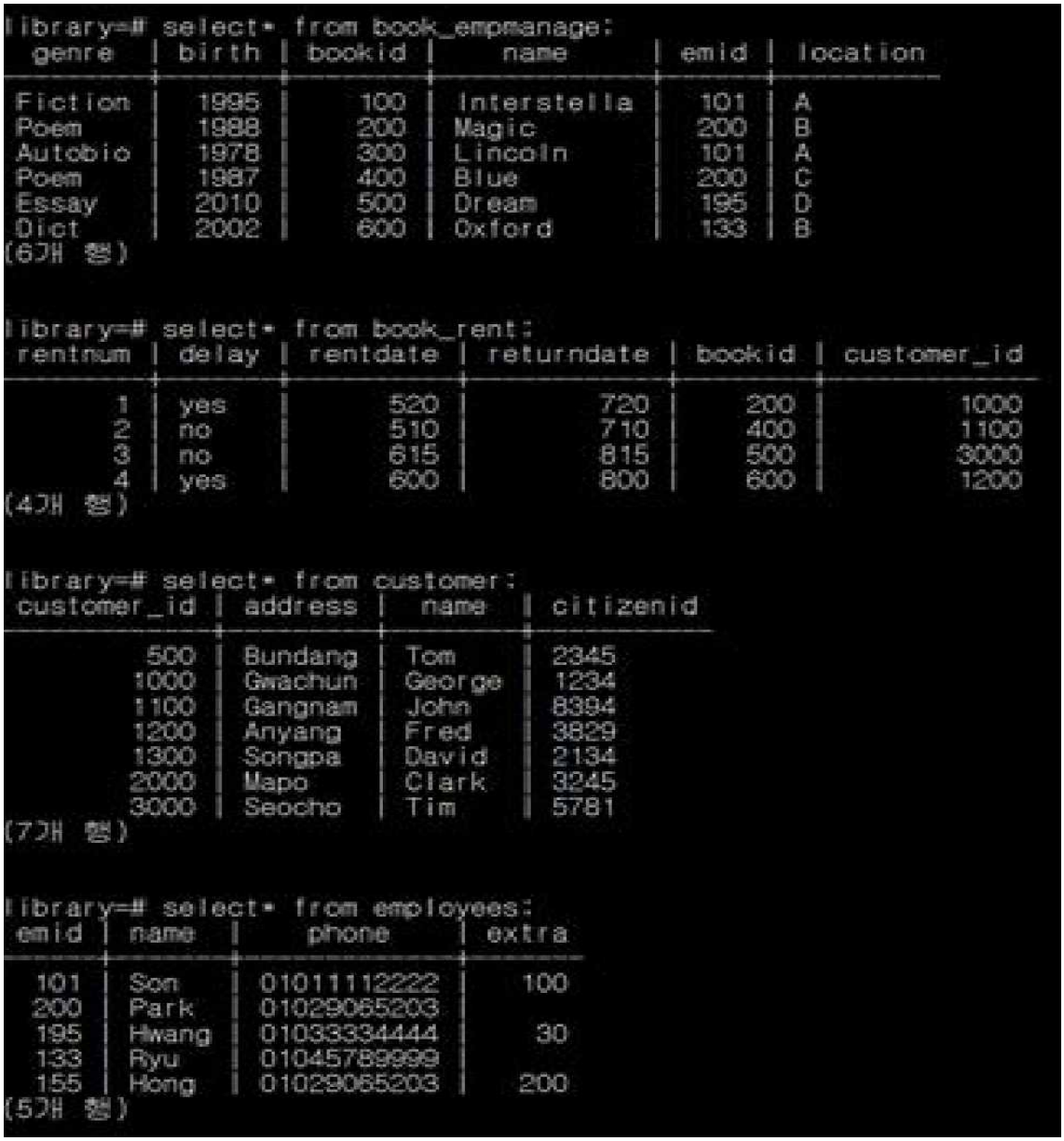
기존 er-diagram과 조금은 차이를 보이고 있습니다. 원래 er diagram에서는 employee가 monitor하는 관계가 aggregation관계였는데, 현재 er-diagram에서는 book manage relation관계로 내려왔습니다. 이러한 변화는 relation table을 만들 때 er-diagram에서 짠 것이 문제가 생기는 것을 알게 되어서 이와 같은 작업을 하였습니다. 또한 기존에 “genre”가 isa관계로 Book 밑에 붙어 있던 것을 도서의 한 속성으로 만들었습니다. 이 또한 table을 만들 때 생기는 문제에 의하여 수정한 것입니다. 또한 직원에 isa 관계로, excellent staff entity를 추가하여 추가수당(extra)을 그 속성으로 묶었습니다. er diagram이 수정되어 가는 과정을 밑에 사진으로 추가했습니다

1.PostgresSQL의\dt 명령어를 통해 현재 만들어진 테이블 목록 출력하고 \d 테이블명 을 이용해 각각 테이블의 정보 보여줘야 함 (모든 테이블 보여 줘야 함)
답:

```
library=# Wdt
          릴레이션(relation) 목록
 스키마 |      이름      |  종류  |  소유주
--------+----------------+--------+----------
 public | book_empmanage | 테이블 | postgres
 public | book_rent      | 테이블 | postgres
 public | customer       | 테이블 | postgres
 public | employees      | 테이블 | postgres
(4개 행)


library=# Wd book_empmanage
                "public.book_empmanage" 테이블
  필드명   |         종류          | Collation | NULL허용 | 초기값
-----------+-----------------------+-----------+----------+--------
 genre     | character varying(11) |           | not null |
 birth     | integer               |           | not null |
 bookid    | integer               |           | not null |
 name      | character varying(11) |           |          |
 emid      | integer               |           | not null |
 location | character varying(1)  |           |          |
인덱스들:
    "book_empmanage_pkey" PRIMARY KEY, btree (bookid)
참조키 제약 조건:
    "fk_emid" FOREIGN KEY (emid) REFERENCES employees(emid) ON DELETE CASCADE
다음에서 참조됨:
    TABLE "book_rent" CONSTRAINT "bookid" FOREIGN KEY (bookid) REFERENCES book_empmanage(bookid)


library=# Wbook_rent
잘못된 명령: Wbook_rent
도움말을 보려면 W?를 입력하십시오.
library=# Wd book_rent
                    "public.book_rent" 테이블
   필드명    |         종류          | Collation | NULL허용 | 초기값
-------------+-----------------------+-----------+----------+--------
 rentnum     | integer               |           | not null |
 delay       | character varying(11) |           | not null |
 rentdate    | integer               |           | not null |
 returndate  | integer               |           | not null |
 bookid      | integer               |           |          |
 customer_id | integer               |           |          |
인덱스들:
    "book_rent_pkey" PRIMARY KEY, btree (rentnum)
참조키 제약 조건:
    "bookid" FOREIGN KEY (bookid) REFERENCES book_empmanage(bookid)
    "customer_id" FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
```

```
library=# Wd customer
                  "public.customer" 테이블
   필드명    |         종류          | Collation | NULL허용 | 초기값
-------------+-----------------------+-----------+----------+--------
 customer_id | integer               |           | not null |
 address     | character varying(30) |           | not null |
 name        | character varying(20) |           | not null |
 citizenid   | character varying(20) |           | not null |
인덱스들:
    "customer_pkey" PRIMARY KEY, btree (customer_id)
다음에서 참조됨:
    TABLE "book_rent" CONSTRAINT "customer_id" FOREIGN KEY (customer_id) REFERENCES customer(customer_id)


library=# Wd employees
                 "public.employees" 테이블
 필드명 |         종류          | Collation | NULL허용 | 초기값
--------+-----------------------+-----------+----------+--------
 emid   | integer               |           | not null |
 name   | character varying(20) |           | not null |
 phone  | character varying(20) |           | not null |
 extra  | integer               |           |          |
인덱스들:
    "employees_pkey" PRIMARY KEY, btree (emid)
다음에서 참조됨:
    TABLE "book_empmanage" CONSTRAINT "fk_emid" FOREIGN KEY (emid) REFERENCES employees(emid) ON DELETE CASCADE
```

2.만들어진 모든 테이블에 대해 모든 데이터를 조회하는 select문



```
library=# select* from book_empmanage;
  genre   | birth | bookid |    name     | emid | location
----------+-------+--------+-------------+------+----------
 Fiction  |  1995 |    100 | Interstella |  101 | A
 Poem     |  1988 |    200 | Magic       |  200 | B
 Autobio  |  1978 |    300 | Lincoln     |  101 | A
 Poem     |  1987 |    400 | Blue        |  200 | C
 Essay    |  2010 |    500 | Dream       |  195 | D
 Dict     |  2002 |    600 | Oxford      |  133 | B
(6개 행)

library=# select* from book_rent;
 rentnum | delay | rentdate | returndate | bookid | customer_id
---------+-------+----------+------------+--------+-------------
       1 | yes   |      520 |        720 |    200 |        1000
       2 | no    |      510 |        710 |    400 |        1100
       3 | no    |      615 |        815 |    500 |        3000
       4 | yes   |      600 |        800 |    600 |        1200
(4개 행)

library=# select* from customer;
 customer_id | address |  name  | citizenid
-------------+---------+--------+-----------
         500 | Bundang | Tom    | 2345
        1000 | Gwachun | George | 1234
        1100 | Gangnam | John   | 8394
        1200 | Anyang  | Fred   | 3829
        1300 | Songpa  | David  | 2134
        2000 | Mapo    | Clark  | 3245
        3000 | Seocho  | Tim    | 5781
(7개 행)

library=# select* from employees;
 emid | name  |     phone     | extra
------+-------+---------------+-------
  101 | Son   | 01011112222   |   100
  200 | Park  | 01029065203   |
  195 | Hwang | 01033334444   |    30
  133 | Ryu   | 01045789999   |
  155 | Hong  | 01029065203   |   200
(5개 행)
```

3.특정 테이블의 특정 컬럼만 조회하는 select문 3개

```
library=# select genre, birth from book_empmanage;
  genre  | birth
---------+-------
 Fiction |  1995
 Poem    |  1988
 Autobio |  1978
 Poem    |  1987
 Essay   |  2010
 Dict    |  2002
(6개 행)


library=# select bookid, customer_id from book_rent;
 bookid | customer_id
--------+-------------
    200 |        1000
    400 |        1100
    500 |        3000
    600 |        1200
(4개 행)


library=# select emid, name from employees;
 emid | name
------+------
  101 | Son
  200 | Park
  195 | Hwang
  133 | Ryu
  155 | Hong
(5개 행)
```

->book_empmange 테이블에서 genre, birth column을 추출

->book_rent 테이블에서 bookid와 customerid를 추출

->emplotee 테이블에서 emid와 name을 추출

4.특정 테이블의 특정 조건의 특정 컬럼만 조회하는 select문 3개

->book_empmanage table에서 location이 A인 bookid를 추출

->customer table에서 customer_id가 1250보다 큰 name을 추출

->book_rent table에서 delay가 yes로 표시되어있는 row의 returndate추출

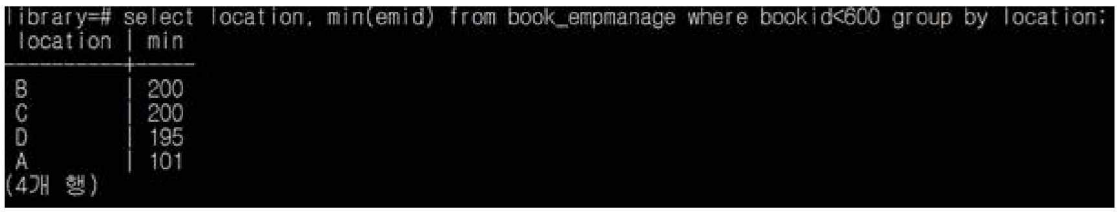
5.특정 조건(where)에 대한 update문 1개

```
library=# select* from customer;
 customer_id | address |  name  | citizenid
-------------+---------+--------+-----------
         500 | Bundang | Tom    |      2345
        1000 | Gwachun | George |      1234
        1100 | Gangnam | John   |      8394
        1200 | Anyang  | Fred   |      3829
        1300 | Songpa  | David  |      2134
        2000 | Mapo    | Clark  |      3245
        3000 | Seocho  | Tim    |      5781
(7개 행)


library=# update customer set address='Incheon' where customer_id=2000;
UPDATE 1
library=# select*from customer;
 customer_id | address |  name  | citizenid
-------------+---------+--------+-----------
         500 | Bundang | Tom    |      2345
        1000 | Gwachun | George |      1234
        1100 | Gangnam | John   |      8394
        1200 | Anyang  | Fred   |      3829
        1300 | Songpa  | David  |      2134
        3000 | Seocho  | Tim    |      5781
        2000 | Incheon | Clark  |      3245
(7개 행)
```

6. Join을 이용한 select문 3개

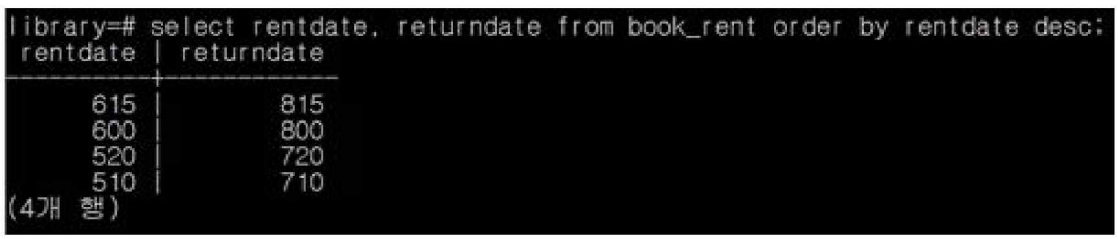
```
library=# select*from book_empmanage b, book_rent r where b.bookid=r.bookid and b.bookid>190;
 genre | birth | bookid |  name  | emid | location | rentnum | delay | rentdate | returndate | bookid | customer_id
-------+-------+--------+--------+------+----------+---------+-------+----------+------------+--------+-------------
 Poem  |  1988 |    200 | Magic  |  200 | B        |       1 | yes   |      520 |        720 |    200 |        1000
 Poem  |  1987 |    400 | Blue   |  200 | C        |       2 | no    |      510 |        710 |    400 |        1100
 Essay |  2010 |    500 | Dream  |  195 | D        |       3 | no    |      615 |        815 |    500 |        3000
 Dict  |  2002 |    600 | Oxford |  133 | B        |       4 | yes   |      600 |        800 |    600 |        1200
(4개 행)


library=# select* from book_rent b, customer c where b.customer_id=c.customer_id;
 rentnum | delay | rentdate | returndate | bookid | customer_id | customer_id | address |  name  | citizenid
---------+-------+----------+------------+--------+-------------+-------------+---------+--------+-----------
       1 | yes   |      520 |        720 |    200 |        1000 |        1000 | Gwachun | George |      1234
       2 | no    |      510 |        710 |    400 |        1100 |        1100 | Gangnam | John   |      8394
       3 | no    |      615 |        815 |    500 |        3000 |        3000 | Seocho  | Tim    |      5781
       4 | yes   |      600 |        800 |    600 |        1200 |        1200 | Anyang  | Fred   |      3829
(4개 행)


library=# select* from book_empmanage b, employees e where b.emid = e.emid;
  genre  | birth | bookid |    name     | emid | location | emid | name |    phone    | extra
---------+-------+--------+-------------+------+----------+------+------+-------------+-------
 Fiction |  1995 |    100 | Interstella |  101 | A        |  101 | Son  | 01011112222 |   100
 Poem    |  1988 |    200 | Magic       |  200 | B        |  200 | Park | 01029065203 |
 Autobio |  1978 |    300 | Lincoln     |  101 | A        |  101 | Son  | 01011112222 |   100
 Poem    |  1987 |    400 | Blue        |  200 | C        |  200 | Park | 01029065203 |
 Essay   |  2010 |    500 | Dream       |  195 | D        |  195 | Hwang| 01033334444 |    30
 Dict    |  2002 |    600 | Oxford      |  133 | B        |  133 | Ryu  | 01045789999 |
(6개 행)
```

->book_empmange table과 book_rent 테이블 각각에서 bookid가 동일하면서, book_empmange의 bookid가 190보다 큰 조건을 만족시키는 행들끼리 join
-> book_rent table과 customer 테이블 각각에서 customerid 가 같으면서 중복되는 것 추출해서 join
->book_empmange table과 employee 테이블 각각에서 emid 가 같으면서 중복되는 것 추출해서 join
7.집계함수와 group by, having, orderby를 이용한 select무 3개



```
library=# select location, min(emid) from book_empmanage where bookid<600 group by location;
 location | min
----------+-----
 B        | 200
 C        | 200
 D        | 195
 A        | 101
(4개 행)
```

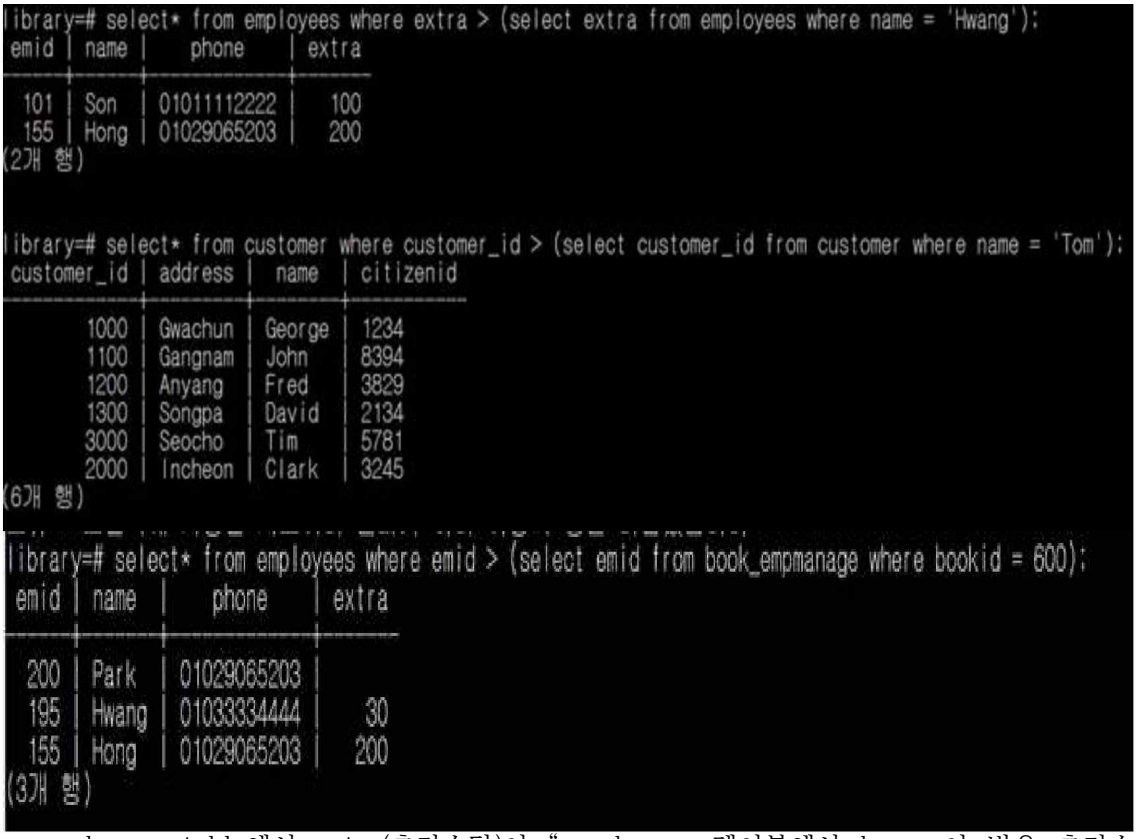
```
library=# select location, min(birth) from book_empmanage where bookid<600 group by location having min(birth) >1980;
 location | min
----------+-----
 B        | 1988
 C        | 1987
 D        | 2010
(3개 행)
```



```
library=# select rentdate, returndate from book_rent order by rentdate desc;
 rentdate | returndate
----------+------------
      615 |        815
      600 |        800
      520 |        720
      510 |        710
(4개 행)
```

->book_empmange에서 bookid가 600보다 작은 것들 추출해서 location과 emid를 추출하고, location에 의해서 grouping한후 location과 emid를 보여주는데, emid는 group에서 가장 작은 값을 min으로 보여준다
->book_empmanage에서 bookid가 600보다 작은 것들을 추출해서 location과 birth를 추출하고, location에 의해서 grouping한수 location과 birth를 보여주는데, birth는 group에서 가장 작은 값을 min으로 보여준다
->book_rent테이블에서 rentdate, returndate column을 추출하고 rendtdate의 값을 내림차순으로 정렬해서 보여준다

8.subquery를 이용한 select문 3개

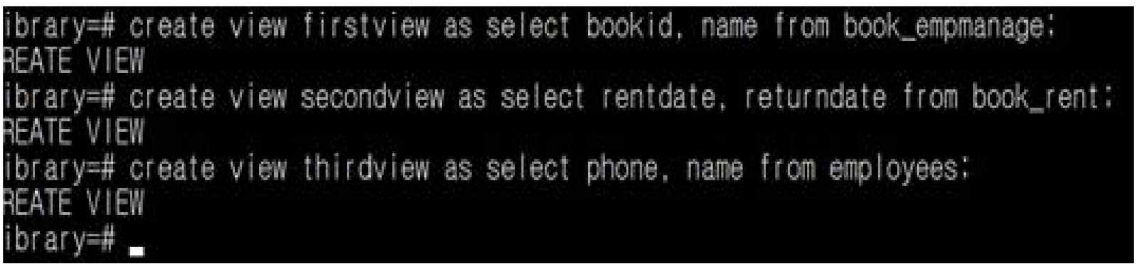


->employees table에서 extra(추가수당)이 “employees 테이블에서 hwang이 받은 추가수당”보다 높은 추가수당을 갖은 행들을 추출

->cutomer table에서 cutomer id가 “customer 테이블에서 Tom의 id"보다 높은 id를 갖은 행들을 추출

->employees table에서 emid가 “book_empmanage 테이블에서 bookid가 600인 책의 emid"보다 높은 emid를 갖은 행들을 추출

9.특정 테이블의 특정 컬럼, 특정 조건에 대해서 view의 생성 3개



10.자유로운 쿼리 생성 5개

문자열 합치기 방식으로, 내가 선택한 column들을 아래와 같이 “/“를 사이에 끼어서 합치는 방식 이다.

```
library=# select name||'/'||emid as name_employee from employees;
 name_employee
---------------
 Son/101
 Park/200
 Hwang/195
 Ryu/133
 Hong/155
(5개 행)
```

내가 선택한 column, 다음과 같이 {column 이름: column값} 으로 표현해 준다

```
library=# select array_to_json(array(select row_to_json(tmp) from(select name from book_empmange group by name)tmp));
오류:  "book_empmange" 이름의 릴레이션(relation)이 없습니다
줄 1: ...ray(select row_to_json(tmp) from(select name from book_empma...
                                                             ^

library=#  select array_to_json(array(select row_to_json(tmp) from(select name from book_empmanage group by name)tmp));
                                                array_to_json
-------------------------------------------------------------------------------------------------------------------------
 [{"name":"Interstella"},{"name":"Dream"},{"name":"Lincoln"},{"name":"Oxford"},{"name":"Magic"},{"name":"Blue"}]
(1개 행)
```

전체 열을 행 단위로 만들기

```
library=# select row_to_json(book_empmanage) from book_empmanage;
                                          row_to_json
-----------------------------------------------------------------------------------------------
 {"genre":"Fiction","birth":1995,"bookid":100,"name":"Interstella","emid":101,"location":"A"}
 {"genre":"Poem","birth":1988,"bookid":200,"name":"Magic","emid":200,"location":"B"}
 {"genre":"Autobio","birth":1978,"bookid":300,"name":"Lincoln","emid":101,"location":"A"}
 {"genre":"Poem","birth":1987,"bookid":400,"name":"Blue","emid":200,"location":"C"}
 {"genre":"Essay","birth":2010,"bookid":500,"name":"Dream","emid":195,"location":"D"}
 {"genre":"Dict","birth":2002,"bookid":600,"name":"Oxford","emid":133,"location":"B"}
(6개 행)
```

특정 열을 기준, 지정한 구분자로 문자열 만들기

```
library=# select location, array_to_string(array_agg(birth), ',')from book_empmanage group by location;
 location | array_to_string
----------+-----------------
 B        | 1988,2002
 C        | 1987
 D        | 2010
 A        | 1995,1978
(4개 행)
```

특정 열만 열 단위로 리스트 만들기