Project2_group2

[기본 CRUD]

MEMBER TABLE

```sql
-- add member
insert into MEMBER values('zizon5520', '강지존', sha1('zxC1234
-- read members
select * from MEMBER;
select * from MEMBER where member_id = 'zizon5520';
-- update member
update MEMBER set member_add = '수정된 주소' where member_id =
-- delete member
delete from MEMBER where member_id = 'zizon5520';
```

GENRE TABLE

```sql
-- add genre
insert into GENRE values(110, '성인', 3000, 300, 3);
-- read genres
select * from GENRE;
select * from GENRE where genre_id = 110;
-- update genre
update GENRE set genre_rent_fee = 3500 where genre_id = 110;
-- delete genre
delete from GENRE where genre_id = 110;
```

VIDEO TABLE

```sql
-- add genre
insert into VIDEO values(102, 'Pulp Fiction', 30, '19세 이상',
```

```
-- read genres
select * from VIDEO;
select * from VIDEO where video_title = 'Pulp Fiction';
-- update genre
update VIDEO set video_reg = 0 where video_id = 102 and video
-- delete genre
delete from VIDEO where video_title = 'Pulp Fiction';
```

RENT_INFO TABLE

```
-- add rent_info
insert into RENT_INFO values(52, 'zizon5520', 23, '2024-07-19
-- read genres
select * from RENT_INFO;
select * from RENT_INFO where rent_id = 52;
-- update genre
update RENT_INFO set returned = 1 where rent_id = 52;
-- delete genre
delete from RENT_INFO where rent_id = 52;
```

[준범님문제]

```
-- 문제1. 하루마다 비디오를 연체 중인 회원의 total_fee를 업데이트 하려한
-- 단 최대 금액은 30,000원을 넘지 않는다.
update rent_info r
join video v on r.video_id = v.video_id
join genre g on v.genre_id = g.genre_id
set r.total_fee = case when r.total_fee + g.genre_late_fee >
```

```
-- 문제2. 가장많은 비디오를 배급한 회사의 비디오정보를 출력하세요
-- 단, 비디오 이름이 겹칠경우에는 1개만 출력하도록 하는 쿼리문을 작성하세요
```

```
select distinct video_title, video_id, genre_id, video_compan
where video_company = (select video_company from video group
```

video_title	video_id	genre_id	video_company
Harry Potter videoand the Sorcerer's Stone	1	10	Warner Bros.
The NeverEnding Story	8	10	Warner Bros.
The Hangover	14	20	Warner Bros.
Mad Max: Fury Road	22	30	Warner Bros.
The Dark Knight	23	30	Warner Bros.
The Matrix	25	30	Warner Bros.
Inception	28	30	Warner Bros.
Prisoners	47	50	Warner Bros.
Mystic River	50	50	Warner Bros.
The Exorcist	51	60	Warner Bros.
The Conjuring	53	60	Warner Bros.
It	57	60	Warner Bros.
Blade Runner	61	70	Warner Bros.
Inception	66	70	Warner Bros.
The Matrix	70	70	Warner Bros.
Full Metal Jacket	94	100	Warner Bros.
Dunkirk	95	100	Warner Bros.
Letters from Iwo Jima	99	100	Warner Bros.
The Dark Knight	101	30	Warner Bros.

```
-- 문제3. 2020, 2021년도 각 회원의 매출을 조회하라. (금액순으로 정렬하여
select member_id, sum(total_fee) as sales
from RENT_INFO
where actual_return_date between '2020-01-01' and '2021-12-31
group by member_id
order by sales desc;
```

member_id	sales
amy_hill	30000
cara_green	5700
rachel_w	4800
uma_lewis	3300
frankM	3300
xanderking	2400
dan_adams	2200
samh1234	2200
david123	2200
tracy1234	2200
alice2023	2200
eve_brown	2200
victorw2022	2200
bobsmith	2200
larry_thom...	2200
wendyyoung	2200
brianscott	2200
larry2021	2200
miayoung	2200
yarawright	2200
charlieB	2200
zacklopez	2200

[민성님 문제]

```
-- 문제1. 99년도 이후에 출시된 비디오를 빌린 회원의 회원id, 비디오이름, 비
-- (대여날짜는 내림차순으로 정렬)
select r.member_id, v.video_title, v.video_release_date, r.re
from RENT_INFO r join VIDEO v on r.video_id = v.video_id
where v.video_release_date >= '00/01/01';
```

member_id	video_title	video_release_date	rent_date
alice2023	The Dark Knight	2008-07-18	2020-01-04
bobsmith	Gone Girl	2014-10-03	2020-02-17
cara_green	The Incredibles	2004-11-05	2020-03-22
frankM	Paranormal Activity	2009-09-25	2020-05-07
david123	Gone Girl	2014-10-03	2020-06-20
miayoung	Anchorman: The Legend of Ron Burgundy	2004-07-09	2020-07-12
samh1234	La La Land	2016-12-09	2020-10-05
uma_lewis	Paranormal Activity	2009-09-25	2020-11-21
victorw2022	The Dark Knight	2008-07-18	2020-12-10
wendyyoung	Gone Girl	2014-10-03	2021-01-15
xanderking	Into the Woods	2014-12-25	2021-02-20
yarawright	Anchorman: The Legend of Ron Burgundy	2004-07-09	2021-03-18
eve_brown	Gone Girl	2014-10-03	2021-07-25
larry_thomp...	The Incredibles	2004-11-05	2021-08-19
tracy1234	Mad Max: Fury Road	2015-05-15	2021-09-14
cara_green	Get Out	2017-02-24	2021-10-30
rachel_w	Superbad	2007-08-17	2021-12-18
eve.davis	La La Land	2016-12-09	2022-01-12
ivy_taylor	Anchorman: The Legend of Ron Burgundy	2004-07-09	2022-04-10
kathywhite	Paranormal Activity	2009-09-25	2022-05-25
ninaharris	The Incredibles	2004-11-05	2022-06-15
grace_w	Into the Woods	2014-12-25	2022-07-07
hankmiller	The Dark Knight	2008-07-18	2022-08-21
jack_anderson	Gone Girl	2014-10-03	2022-09-10
quinn_m	Paranormal Activity	2009-09-25	2022-11-12
frank_allen	Anchorman: The Legend of Ron Burgundy	2004-07-09	2022-12-08
oscarm99	Gone Girl	2014-10-03	2023-03-15
oscar_harris	The Incredibles	2004-11-05	2023-04-22
samh1234	Mad Max: Fury Road	2015-05-15	2023-05-20
stevenp	La La Land	2016-12-09	2023-06-30
paulgar	Paranormal Activity	2009-09-25	2023-07-15
paul_clark	Superbad	2007-08-17	2023-08-09
queensam	Anchorman: The Legend of Ron Burgundy	2004-07-09	2023-11-21
miaw_123	Gone Girl	2014-10-03	2023-12-14
gracedavis	The Incredibles	2004-11-05	2024-01-12
zizon5520	The Dark Knight	2008-07-18	2024-07-19

-- 문제2. 관람등급마다 비디오가 몇개 있는지를 구해서 관람등급,개수를 출력한
-- (groub by함수를 사용할 것.)
select video_rating, count(*) from VIDEO group by video_ratin

video_rating	count(*)
전체	32
12세 이상	17
15세 이상	38
19세 이상	14

video_title	video_id	genre_id	video_company
Harry Potter videoand the Sorcerer's Stone	1	10	Warner Bros.
The NeverEnding Story	8	10	Warner Bros.
The Hangover	14	20	Warner Bros.
Mad Max: Fury Road	22	30	Warner Bros.
The Dark Knight	23	30	Warner Bros.
The Matrix	25	30	Warner Bros.
Inception	28	30	Warner Bros.
Prisoners	47	50	Warner Bros.
Mystic River	50	50	Warner Bros.
The Exorcist	51	60	Warner Bros.
The Conjuring	53	60	Warner Bros.
It	57	60	Warner Bros.
Blade Runner	61	70	Warner Bros.
Inception	66	70	Warner Bros.
The Matrix	70	70	Warner Bros.
Full Metal Jacket	94	100	Warner Bros.
Dunkirk	95	100	Warner Bros.
Letters from Iwo Jima	99	100	Warner Bros.
The Dark Knight	101	30	Warner Bros.

```
-- 문제3. 관람등급을 12세 이상, 15세 이상을 전체이용가로 변경한다
update VIDEO set video_rating='전체' where video_rating='12세 (
```

member_id	sales
amy_hill	30000
cara_green	5700
rachel_w	4800
uma_lewis	3300
frankM	3300
xanderking	2400
dan_adams	2200
samh1234	2200
david123	2200
tracy1234	2200
alice2023	2200
eve_brown	2200
victorw2022	2200
bobsmith	2200
larry_thom...	2200
wendyyoung	2200
brianscott	2200
larry2021	2200
miayoung	2200
yarawright	2200
charlieB	2200
zacklopez	2200

[연화님 문제]

```
-- 문제1. 회원중 최고령자와 최연소자를 찾아서 아이디와 이름과 생년월일을 조
select member_id, member_name, member_birth
from MEMBER
where member_birth = (select min(member_birth) from MEMBER)
or member_birth = (select max(member_birth) from MEMBER);
```

member_id	member_name	member_birth
oscarm99	이도현	1970-03-25
yarawright	김미래	2009-01-15

```
-- 문제2. 고객 아이디, 이름, 이용횟수를 이용횟수 내림차순으로 정렬하여 출력
select m.member_id, m.member_name, (select count(*) from rent_
from MEMBER m
order by count desc;
```

member_id	member_name	count
stevenp	강민수	1
zacklopez	정지훈	1
dan_adams	정민호	1
hankmiller	조성현	1
larry_thom...	민하영	1
paul_clark	민하영	1
tinaclark	이현정	1
zizon5520	강지존	1
david123	김지훈	1
hankmoore	한승훈	1
larry2021	김하늘	1
paulgar	홍길동	1
tracy1234	이승철	1
alice2023	김민수	1
eve_brown	김진아	1
ivy_taylor	이서연	1
miaw_123	이진아	1
queensam	홍길동	1
uma_lewis	서준희	1
amy_hill	홍진수	1
eve.davis	이소영	1
quinn_m	오수민	1
victorw2022	김유진	1
bobsmith	이영희	1
frank_allen	박성현	1
jack_ande...	서진호	1
ninaharris	정유진	1
rachel_lee	김수정	1
wendyyoung	김하연	1
brianscott	최수현	1
frankM	장민호	1
jackA99	오진수	1
ninataylor	유진아	1
xanderking	이진희	1
grace_w	최지연	1
kathy_85	박서진	1
oscar_harris	이주영	1

[다훈님 문제]

```
-- 문제1. 비디오 가게에 가장 큰 지출을 낸 사용자의 회원id와 이름, 총금액을
select m.member_id, m.member_name, sum(r.total_fee) as salse
from MEMBER m join RENT_INFO r on m.member_id = r.member_id
group by member_id
order by salse desc
limit 1;
```

member_id	member_name	salse
amy_hill	홍진수	30000

```
-- 문제2. 고객들이 가장 많이 대여한 비디오를 검색하여 그 비디오의 장르의 빈
SELECT video_id, genre_id
FROM VIDEO
WHERE video_id = (
SELECT video_id
FROM RENT_INFO
GROUP BY video_id
ORDER BY COUNT(video_id) DESC
LIMIT 1
);
```

video_id	genre_id
45	50

```
-- 문제3. 블랙리스트 컬럼을 member 테이블에 추가하세요
alter table MEMBER
add is_blacked boolean;
```

[진구]

```
-- 문제1. 사장님이 이벤트를 위해 출시일이 30년이 넘은 비디오를 조회하려 한
-- 1-1. 비디오 정보 조회
select video_id, video_title from VIDEO where video_release_d

-- 1-2. 개수 조회
select count(*) from VIDEO where video_release_date < date_su
```

video_id	video_title
7	The Princess Bride
8	The NeverEnding Story
10	Willow
16	Ferris Bueller's Day Off
19	Airplane!
21	Die Hard
42	The Silence of the Lambs
43	Psycho
51	The Exorcist
52	A Nightmare on Elm Street
61	Blade Runner
62	Star Wars: Episode IV - A...
64	The Terminator
68	E.T. the Extra-Terrestrial
74	The Lion King
81	The Sound of Music
87	Beauty and the Beast
88	Grease
89	West Side Story
93	Apocalypse Now
94	Full Metal Jacket
98	Platoon

count(*)
22

```
-- 문제2. 오픈 이후 세 번 이상 대여된 비디오의 정보를 횟수 기준으로 내림차
-- (video_id, video_title, 총 대여 횟수 포함)
select v.video_id, v.video_title, count(*) as count
from RENT_INFO r
join VIDEO v on r.video_id = v.video_id
group by v.video_id
having count >= 3
order by count desc;
```

video_id	video_title	count
45	Gone Girl	7
67	The Fifth Element	6
12	Anchorman: The Legend of Ron Burgundy	5
78	The Incredibles	5
89	West Side Story	5
56	Paranormal Activity	5
23	The Dark Knight	4
33	Titanic	4
34	La La Land	3

```sql
-- 문제3. 하루마다 반납이 되지 않은(연체중인) 대여에 대해 total_fee를 업
-- total_fee가 30,000원을 넘기는 날 actual_return_date에 해당 날짜
-- 3-1. total_fee update
update RENT_INFO r
join VIDEO v on r.video_id = v.video_id
join GENRE g on v.genre_id = g.genre_id
set r.total_fee = g.genre_rent_fee + (g.genre_late_fee * TIME
where r.returned = 0 and r.actual_return_date is null;

-- 3-2. total_fee >= 30,000 인 경우
-- actual_return_date를 오늘 날짜로 수정
update RENT_INFO
set actual_return_date = now()
where actual_return_date is null and total_fee >= 30000;
```