

Group 3_8

Play Archive

Queries

Group Leader :

Jay Sanghani (202101185)

Group Members :

Dhyey Ladani (202101182)

Jay Sanghani (202101185)

Utsav Maru (202101195)

Krish Rupapara (202101198)

1. Find all purchased / installed items of a given user

```
select user_id, title_id, title from
(select user_id,purchased_apps.app_id as title_id ,app_name as title
from purchased_apps
      join app on purchased_apps.app_id = app.app_id natural join
users
      union
select user_id, movie_id as title_id, movie_name as title from users
natural join purchased_movies
      natural join movie
      union
select user_id, book_id as title_id, book_name as title from users
natural join purchased_books
      natural join ebook) as w
where user_id = 104 order by title_id;
```

2. Give all wishlists of a given user.

```
Select
user_id, title_id, title from
(select user_id, app_id as title_id, app_name as title from users
natural join app_wishlist natural join app
      union
select user_id, movie_id as title_id, movie_name as title from users
natural join movie_wishlist
      natural join movie
      union
select user_id, book_id as title_id, book_name as title from users
natural join book_wishlist
      natural join ebook) as w
where user_id=104 order by title_id;
```

3. Find ratings of purchased items given by a single User.

```
select user_id, title_id, title, given_rating from
    (select user_id, pap.app_id as title_id, app_name as title,
given_rating from purchased_apps pap
    join app ap on pap.app_id = ap.app_id natural join users
    union
    select user_id, pap.movie_id as title_id, movie_name as title,
given_rating from purchased_movies pap
    join movie ap on pap.movie_id = ap.movie_id natural join
users
    union
    select user_id, pap.book_id as title_id, book_name as title,
given_rating from purchased_books pap
    join ebook ap on pap.book_id = ap.book_id natural join
users
    ) as w
where user_id=104 order by title_id;
```

4. Find transaction History for a given user.

```
select
transaction_id,recipient_id,amount,transaction_date,payment_method
from transaction_details
join users on transaction_details.source_id = users.user_id
where user_id = 104;
```

5. Find the related apps that the user has searched for.

```
select
app_id, app_name from app natural join app_category where
app_category.app_category
in (select app_category from app natural join app_category where
app_name='Spotify');
```

6. Find Apps which have pending updates for a given user.

```
select
purchased_apps.app_id, app_name from purchased_apps join app on
purchased_apps.app_id = app.app_id
where user_id = 104 and update_available = 'true';
```

7. Top 3 weekly apps.

```
select r.app_id, app_name, dev_id, dev_name, count(user_id) as
downloads
from (app natural join developer) r join purchased_apps pap on
r.app_id = pap.app_id
where installed_date between '2022-02-01' and '2022-02-07'
group by r.app_id, app_name, dev_id, dev_name
order by count(user_id) desc limit 3;
```

8. Identify the apps that have the highest user engagement, defined as the ratio of the number of reviews to the number of installs.

```

select ap.app_id, app_name,
round((count(given_rating)::numeric/downloads)::numeric, 2) as
user_engagement
from app ap join purchased_apps pap on ap.app_id = pap.app_id
where given_rating <>
group by ap.app_id, app_name
order by user_engagement desc;

```

9. Find category wise revenue of an app developer

```

Select
app_category,sum(price*downloads) categorywise_revenue from app
natural join app_category natural join developer
where dev_id=1114 group by dev_id,app_category;

```

10. Find average rating given by user on their downloaded apps

```

select user_id, round(avg(given_rating)::numeric , 2) as avg_rating
from app
join purchased_apps on app.app_id = purchased_apps.app_id
natural join users group by user_id
order by avg_rating desc;

```

11. Retrieve a list of recommended apps for a user based on their given ratings.

```

select app_id, app_name, r.app_category, rating from (

(select * from app natural join app_category) r join

```

```

        (select app_category, avg(given_rating)
        from (users natural join purchased_apps) r1 join app_category
        cat on r1.app_id = cat.app_id
        where user_id = 150 group by app_category order by
        avg(given_rating) desc limit 1) r2

        on r.app_category = r2.app_category

    ) order by rating desc limit 10;

```

12. Countrywise popularity of a given app based on downloads

```

select country, count(user_id) from app
join purchased_apps on app.app_id = purchased_apps.app_id
natural join users
where app.app_id=1101
group by country
order by count(user_id) desc;

```

13. Top 10 movies based on revenue.

```

select movie_id, movie_name, sum(price*downloads) as revenue,
studio_id, studio_name
from movie natural join studio
group by movie_id,movie_name,studio_id,studio_name
order by sum(price*downloads) desc limit 10;

```

14. Popular movies in a given country.

```
select movie_id, movie_name, count(user_id) as total_downloads
from movie natural join purchased_movies natural join users
where country = 'United Kingdom'
group by movie_id, movie_name
order by total_downloads desc;
```

15. Top 10 studio according to IMDB rating of their rating.

```
select studio_id, studio_name, round( avg(imdb)::numeric, 2 ) as
IMDB_rating
from movie natural join studio
group by studio_id, studio_name
order by avg(imdb) desc limit 10;
```

16. Find revenue of a studio based on category on each day

```
select dev_id, app_category, sum(price) as revenue, installed_date
from( app natural join app_category ) r
join purchased_apps pap on r.app_id = pap.app_id
where dev_id = 1111
group by dev_id, app_category, installed_date;
```

17. Top 10 books by Number of Downloads.

```
select
book_id, book_name, price, downloads, publisher_id, publisher_name
from ebook
natural join publisher
```

order by downloads desc limit 10;

18. Top 10 users who have downloaded the most books.

```
select user_id,user_name,count(app_id) as downloads
from purchased_apps natural join users
group by user_id,user_name
order by downloads desc limit 10;
```

19. Find revenue of a publisher on a given date.

```
select publisher_id, book_category, sum(price) from (ebook natural
join book_category) r
join purchased_books pap on r.book_id = pap.book_id
where installed_date = '2021-07-02'
group by publisher_id,book_category;
```

20. Most popular book category based on number of downloads

```
select book_category, count(book_category) as no_of_downloads
from ebook natural join book_category as w
join purchased_books on w.book_id = purchased_books.book_id
group by book_category
order by count(book_category) desc;
```