```
-- Your Name: Chuqi Wang (79167724)
-- 1. Table Creation and Analysis --
-- SQL DDL statements
\i /Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW1/ZotMusicDDL.sql
DROP SCHEMA IF EXISTS ZotMusic CASCADE;
DROP SCHEMA
CREATE SCHEMA ZotMusic;
CREATE SCHEMA
SET search_path TO ZotMusic;
SET

CREATE TABLE Users (
    user_id        text NOT NULL,
    email          text NOT NULL,
    joined_date    date NOT NULL,
    nickname       text NOT NULL,
    street         text,
    city           text,
    state          text,
    zip            text,
    genres         text,
    PRIMARY KEY (user_id)
);
CREATE TABLE

CREATE TABLE Artists (
    user_id        text,
    bio            text,
    stagename      text,
    PRIMARY KEY (user_id),
    FOREIGN KEY (user_id) REFERENCES Users (user_id) ON DELETE CASCADE
);
CREATE TABLE

CREATE TABLE Listeners (
    user_id        text,
    subscription   text,
    first_name     text NOT NULL,
    last_name      text NOT NULL,
    PRIMARY KEY (user_id),
    FOREIGN KEY (user_id) REFERENCES Users (user_id) ON DELETE
CASCADE,
    CHECK (subscription IN ('free', 'monthly', 'yearly'))
);
CREATE TABLE

CREATE TABLE Records (
    record_id      text NOT NULL,
    artist_user_id text NOT NULL,
    title          text NOT NULL,
```

```
    genre          text NOT NULL,
    release_date   date NOT NULL,
    PRIMARY KEY (record_id),
    FOREIGN KEY (artist_user_id) REFERENCES Artists (user_id) ON
DELETE CASCADE
);
CREATE TABLE

CREATE TABLE Singles (
    record_id      text NOT NULL,
    video_url      text,
    PRIMARY KEY (record_id),
    FOREIGN KEY (record_id) REFERENCES Records (record_id) ON DELETE
CASCADE
);
CREATE TABLE

CREATE TABLE Albums (
    record_id      text NOT NULL,
    description    text,
    PRIMARY KEY (record_id),
    FOREIGN KEY (record_id) REFERENCES Records (record_id) ON DELETE
CASCADE
);
CREATE TABLE

CREATE TABLE Songs (
    record_id      text NOT NULL,
    track_number   int NOT NULL,
    title          text NOT NULL,
    length         int NOT NULL,
    bpm            int,
    mood           text,
    PRIMARY KEY (record_id, track_number),
    FOREIGN KEY (record_id) REFERENCES Records (record_id) ON DELETE
CASCADE
);
CREATE TABLE

CREATE TABLE Sessions (
    session_id        text NOT NULL,
    user_id           text NOT NULL,
    record_id         text NOT NULL,
    track_number      int NOT NULL,
    initiate_at       timestamp NOT NULL,
    leave_at          timestamp NOT NULL,
    music_quality     text NOT NULL,
    device            text NOT NULL,
    remaining_time    int NOT NULL,
    replay_count      int,
```

```
    PRIMARY KEY (session_id),
    FOREIGN KEY (user_id) REFERENCES Listeners(user_id) ON DELETE
CASCADE,
    FOREIGN KEY (record_id, track_number) REFERENCES Songs(record_id,
track_number) ON DELETE CASCADE
);
CREATE TABLE

CREATE TABLE Reviews (
    review_id     text NOT NULL,
    user_id       text NOT NULL,
    record_id     text NOT NULL,
    rating        int NOT NULL,
    body          text,
    posted_at     timestamp NOT NULL,
    PRIMARY KEY (review_id),
    FOREIGN KEY (user_id) REFERENCES Listeners (user_id) ON DELETE
CASCADE,
    FOREIGN KEY (record_id) REFERENCES Records (record_id) ON DELETE
CASCADE
);
CREATE TABLE

CREATE TABLE ReviewLikes(
    user_id text NOT NULL,
    review_id text NOT NULL,
    PRIMARY KEY (user_id, review_id),
    FOREIGN KEY (user_id) REFERENCES Listeners(user_id) ON DELETE
CASCADE,
    FOREIGN KEY (review_id) REFERENCES Reviews(review_id) ON DELETE
CASCADE
);
CREATE TABLE
-- Songs table field design observations
\d songs
                 Table "zotmusic.songs"
    Column    |  Type   | Collation | Nullable | Default
--------------+---------+-----------+----------+---------
 record_id    | text    |           | not null |
 track_number | integer |           | not null |
 title        | text    |           | not null |
 length       | integer |           | not null |
 bpm          | integer |           |          |
 mood         | text    |           |          |
Indexes:
    "songs_pkey" PRIMARY KEY, btree (record_id, track_number)
Foreign-key constraints:
    "songs_record_id_fkey" FOREIGN KEY (record_id) REFERENCES
records(record_id) ON DELETE CASCADE
Referenced by:
```

```
    TABLE "sessions" CONSTRAINT "sessions_record_id_track_number_fkey"
FOREIGN KEY (record_id, track_number) REFERENCES songs(record_id,
track_number) ON DELETE CASCADE

/*
record_id: text NOT NULL
track_number: int NOT NULL
title: text NOT NULL
length: int NOT NULL
bpm: int
mood: text
*/
-- 2. Data Loading (COPY commands) --
\copy Users from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW1/
zot-music-dataset-assignment1/Users.csv' delimiter ',' csv header;
COPY 200
\copy Artists from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/
HW1/zot-music-dataset-assignment1/Artists.csv' delimiter ',' csv
header;
COPY 116
\copy Listeners from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/
HW1/zot-music-dataset-assignment1/Listeners.csv' delimiter ',' csv
header;
COPY 184
\copy Records from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/
HW1/zot-music-dataset-assignment1/Records.csv' delimiter ',' csv
header;
COPY 1000
\copy Singles from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/
HW1/zot-music-dataset-assignment1/Singles.csv' delimiter ',' csv
header;
COPY 300
\copy Albums from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/
HW1/zot-music-dataset-assignment1/Albums.csv' delimiter ',' csv
header;
COPY 700
\copy Songs from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/HW1/
zot-music-dataset-assignment1/Songs.csv' delimiter ',' csv header;
COPY 6252
\copy Sessions from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/
HW1/zot-music-dataset-assignment1/Sessions.csv' delimiter ',' csv
header;
COPY 50000
\copy Reviews from '/Users/chuqiwang/Desktop/UCI/CS224P/assignments/
HW1/zot-music-dataset-assignment1/Reviews.csv' delimiter ',' csv
header;
COPY 9499
\copy ReviewLikes from '/Users/chuqiwang/Desktop/UCI/CS224P/
assignments/HW1/zot-music-dataset-assignment1/ReviewLikes.csv'
delimiter ',' csv header;
```

```
COPY 91325
-- 3. Query Answers --
set search_path to zotmusic;
SET
-- Problem A --
select 'Users' as entity, count(*) from Users union all
select 'Records' as entity, count(*) from Records union all
select 'Reviews' as entity, count(*) from Reviews;
 entity  | count
---------+-------
 Users   |   200
 Records |  1000
 Reviews |  9499
(3 rows)

-- Problem B --
select u.user_id, u.email, u.nickname, u.zip from users as u
join artists on artists.user_id = u.user_id
join listeners on listeners.user_id = u.user_id
where email like '%@icloud.com';
                  user_id                  |           email
|    nickname    |  zip
-----------------------------------------
+--------------------------+----------------+-------
 user_10dfa3b6-52b6-43a9-835a-ad110ad50ff2 | roberthammond@icloud.com
| roberthammond  | 54869
 user_4a9ffbf6-5430-45a4-b68e-0ae6f6737d8b | william09@icloud.com
| william09      | 55308
 user_457ad608-9661-4384-9919-1d89c52fd0de | danielharrison@icloud.com
| danielharrison | 23703
 user_d383327a-dd9c-4a4f-bbaf-262c7a6d90a0 | chelsealawson@icloud.com
| chelsealawson  | 10206
 user_83a40c0c-573e-44ec-8cac-b5951513f88b | turnerkayla@icloud.com
| turnerkayla    | 02182
 user_38eaa9f8-e8fc-4ce4-a8ae-ffb882c1786c | ryanmorgan@icloud.com
| ryanmorgan     | 95166
 user_3c5d30bc-0ac2-4df1-8574-892d2f666df6 | browncarrie@icloud.com
| browncarrie    | 23550
(7 rows)

-- Problem C --
select record_id, title, genre, release_date from records
where artist_user_id = (
        select a.user_id from artists as a
        join users as u on a.user_id = u.user_id
        where email = 'fwilson@outlook.com'
)
order by release_date;
                  record_id                  |           title
|  genre  | release_date
```

```
-------------------------------------------
+---------------------------+---------+-------------
 record_91c6325d-b17f-4f4c-be6d-3517b2173a9f | Statement matter
| Country | 2020-01-12
 record_822961a3-946a-49ff-8173-74d4035286b9 | Apply size
| Gospel  | 2020-01-29
 record_cbf93efd-2deb-48ae-ad73-83aa088c6f13 | Democratic what
| Soul    | 2020-03-27
 record_2406e933-23e3-4db1-acf9-3c863d48bff6 | General job heavy
| Country | 2020-05-08
 record_57061d35-de20-4bf1-9aac-a689f0db7e16 | Would determine
| Soul    | 2020-06-07
 record_3e4ed054-cf1a-4a04-8e97-e0177c6d3575 | Summer civil political
beat | Folk    | 2021-03-31
 record_116fbdd6-e706-41f7-9809-12e174e48e8f | Discover rate
| Jazz    | 2021-09-09
 record_62389a63-e95f-43d1-acea-aa1bac0e0050 | Result guess for
| Gospel  | 2021-10-17
 record_5cbf14c7-7b54-4e32-bfce-cba507c7277f | Bar talk long
| Jazz    | 2021-10-23
(9 rows)

-- Problem D --
with cte as (
        select * from records
        where artist_user_id = (
        select a.user_id from artists as a
        join users as u on a.user_id = u.user_id
        where email = 'fwilson@outlook.com'
        )
)
select artist_user_id, genre,
        count(case when a.record_id is not null then 1 end) as
album_count,
        count(case when s.record_id is not null then 1 end) as
single_count
from cte
left join singles as s on cte.record_id = s.record_id
left join albums as a on cte.record_id = a.record_id
group by artist_user_id, genre;
              artist_user_id               |  genre  | album_count |
single_count
-------------------------------------------+---------+-------------
+-------------
 user_6ac27408-a0a6-4c57-a025-7b6854f7a8e3 | Country |           1 |
1
 user_6ac27408-a0a6-4c57-a025-7b6854f7a8e3 | Folk    |           1 |
0
 user_6ac27408-a0a6-4c57-a025-7b6854f7a8e3 | Gospel  |           2 |
0
```

```
 user_6ac27408-a0a6-4c57-a025-7b6854f7a8e3 | Jazz    |           1 |
1
 user_6ac27408-a0a6-4c57-a025-7b6854f7a8e3 | Soul    |           1 |
1
(5 rows)

-- Problem E --
select coalesce(a.stagename, u.nickname) as name,
u.email from users as u
join artists as a on u.user_id = a.user_id
join records as r on u.user_id = r.artist_user_id
group by a.user_id, a.stagename, u.email, u.nickname
having count(distinct r.genre) >= 9;
      name      |             email
----------------+-------------------------------
 blakeshannon   | william09@icloud.com
 elizabeth55    | powerschristopher@foxmail.com
 ehester        | william57@mail.com
 richardsbilly  | robert92@outlook.com
 sandersallison | keith65@university.edu
 khall          | lthompson@college.edu
(6 rows)

-- Problem F --
with cte as (
        select artist_user_id from records as r
        where r.genre in ('R&B', 'Hip-Hop')
        group by artist_user_id
        having count(distinct case when r.genre = 'R&B' then 1 end) >
0 and
        count(distinct case when r.genre = 'Hip-Hop' then 1 end) > 0
)
select u.user_id, u.email from users as u
join cte on u.user_id = cte.artist_user_id
where u.user_id not in (
        select distinct artist_user_id from records as r
        where r.genre in ('Indie', 'Jazz')
)
order by u.user_id;
                  user_id                  |            email
-------------------------------------------
+----------------------------
 user_377075cd-a1c3-4f19-816c-0fdcfc973607 | cflores@university.edu
 user_39c8d999-40e5-4fd0-9cd5-96a93239abee |
charleslewis@university.edu
 user_3fbded9d-d59a-435c-863f-55f0b086f01e | zhill@hotmail.com
 user_5a61e935-0fec-4c35-9bc4-58f9e7ecb067 | molly19@protonmail.com
 user_80b63994-e7f9-4f57-ab42-88683a70d183 | william57@mail.com
 user_81b9d067-7877-43cf-9793-6fb5ea6f4921 | melissa63@university.edu
 user_83a40c0c-573e-44ec-8cac-b5951513f88b | turnerkayla@icloud.com
```

```
 user_94b498aa-2a93-4e8e-8efa-229604beea67 | mcguiresheila@outlook.com
 user_b26fbc8f-e3c8-426f-9d4c-77eec730bacd | anthony59@college.edu
 user_bf4b7630-b9dd-40f0-b534-ef841ea43194 | linda38@outlook.com
 user_f6cb31b3-48fa-4cd1-a483-7a59189af5d1 | joseph37@yahoo.com
(11 rows)

-- Problem G --
select email, nickname, array_length(string_to_array(genres, ','), 1)
as num_genres
from users
order by num_genres desc
limit 10;
            email             |   nickname    | num_genres
------------------------------+---------------+------------
 courtney36@protonmail.com    | courtney36    |         10
 charleslewis@university.edu  | charleslewis  |         10
 ewilliams@mail.com           | ewilliams     |         10
 zmason@gmail.com             | zmason        |         10
 gomezbrittany@foxmail.com    | gomezbrittany |          9
 bknapp@icloud.com            | bknapp        |          9
 ryanmorgan@icloud.com        | ryanmorgan    |          9
 edwardscindy@foxmail.com     | edwardscindy  |          8
 joel00@gmail.com             | joel00        |          8
 gclayton@protonmail.com      | gclayton      |          8
(10 rows)

-- Problem H --
select unnest(string_to_array(genres, ',')) as genre, count(user_id)
as num_users
from users
group by genre
order by num_users desc;
   genre    | num_users
------------+-----------
 Soul       |        58
 Techno     |        56
 Indie      |        54
 Folk       |        53
 Blues      |        53
 Funk       |        49
 Country    |        49
 Classical  |        47
 R&B        |        47
 Metal      |        47
 Jazz       |        47
 Disco      |        46
 Hip-Hop    |        45
 Latin      |        45
 Pop        |        44
 Gospel     |        43
```

```
 Punk       |        42
 Electronic |        38
 Rock       |        33
 Reggae     |        27
(20 rows)

-- Problem I --
select r.title, r.record_id, s.track_number, s.title,
count(se.user_id) as num_listeners
from sessions as se
join songs as s on se.record_id = s.record_id and se.track_number =
s.track_number
join records as r on se.record_id = r.record_id
where se.remaining_time <= s.length * 0.2 or se.replay_count > 0
group by r.record_id, s.track_number, s.title
order by num_listeners desc
limit 10;
            title            |                  record_id
| track_number |        title         | num_listeners
----------------------------
+--------------------------------------+-------------
+----------------------+---------------
 Small century stop          | record_154ce2a2-1e2c-48c4-
be66-41ddf41245a7 |            8 | Local stay keep      |
16
 Popular                     | record_7e742b33-f00b-4356-83a0-
f1f0731ce72c |            5 | Brother despite      |            16
 Personal do physical        | record_3a1932fb-22f1-4834-b0fb-
c1cf3013fa42 |            2 | Identify community   |            16
 Benefit him these           | record_a120d09b-
ba72-4d34-9d81-021c01c2e3fa |            1 | Police will human    |
16
 Dog                         | record_422a9857-f805-4e65-b4f0-
bac88d34f4fb |            5 | Water                |            15
 Understand yourself suggest | record_e5260c91-
fa30-471c-8711-128a4e8ff977 |            5 | Direction air        |
15
 Western fear                | record_b6f51c72-3534-4b29-
ae62-00580338f467 |            6 | Myself letter across |
14
 Stop watch different        | record_d9120d24-b69a-41bf-
ab21-3a42beb4987b |            3 | Skill throw          |
14
 Western parent firm speech  | record_0104059c-6029-42d6-a1e5-
c6e900aa2088 |           10 | Market test could    |            14
 Such view economy           | record_489ad7e4-9a03-44fa-
b43b-409fd9fe195c |            8 | Industry when        |
14
(10 rows)
```

```
-- Problem J --
  -- View DDL:
create view RatedRecords as
with cte as (
        select rv.review_id, rv.record_id, rv.rating,
        coalesce(count(rl.user_id), 0) + 1 as weight
        from reviews as rv
        left join reviewlikes as rl on rv.review_id = rl.review_id
        group by rv.review_id, rv.record_id, rv.rating
)
select r.title, r.record_id,
coalesce(sum(cte.rating * cte.weight) / nullif(sum(cte.weight), 0), 0)
as rating,
count(cte.review_id) as num_reviews
from records as r
join cte on r.record_id = cte.record_id
group by r.title, r.record_id;
CREATE VIEW
  -- View test query:
select * from RatedRecords
where num_reviews >= 5
order by rating desc
limit 10;
          title          |                  record_id
|       rating       | num_reviews
------------------------
+--------------------------------------+-------------------
+-------------
 Second concern star     | record_ab8ecafc-cae3-4394-824b-45f860d63c8a
| 4.3287671232876712 |           9
 Real growth             | record_dd3f83b0-f0ec-41a0-b048-83a7dd515191
| 4.2375000000000000 |           7
 Crime                   | record_602a6db2-1784-45d2-8645-1c141f25a049
| 4.2142857142857143 |           6
 News explain might turn | record_cbf50912-23ca-4de0-b94d-41b2076fd9d3
| 4.2121212121212121 |           6
 Black movement          | record_acb14a89-3854-45de-9df7-ef1be87c7e47
| 4.1724137931034483 |           5
 Water film              | record_0e4952cc-3445-4fe6-80c9-867e98ae4233
| 4.1630434782608696 |           8
 Be sign hair            | record_e8fa438d-3e01-40df-bd1c-5d7dadf98015
| 4.1532258064516129 |          10
 Music religious charge  | record_3cd5b91e-f6c5-469a-8f92-98f85a1a7af7
| 4.1392405063291139 |           7
 Change adult            | record_9388a7fd-5828-4438-82b1-46c13e63c6c2
| 4.1241830065359477 |          17
 Sit pay political       | record_66f81619-1bf3-4784-bd94-9c3adeeb5785
| 4.1086956521739130 |           6
(10 rows)
```

```
-- Problem K --
  -- Table alteration DDL:
alter table records
add rating decimal(3, 2);
ALTER TABLE
  -- Table update query:
update records as r
set rating = RatedRecords.rating
from RatedRecords
where r.record_id = RatedRecords.record_id;
UPDATE 1000
  -- Change verification query:
select r.title, r.record_id, r.rating, RatedRecords.num_reviews
from records as r
join RatedRecords on r.record_id = RatedRecords.record_id
where RatedRecords.num_reviews >= 5
order by r.rating desc
limit 10;
          title          |                  record_id
| rating | num_reviews
------------------------
+--------------------------------------+--------+-------------
 Second concern star     | record_ab8ecafc-cae3-4394-824b-45f860d63c8a
|   4.33 |           9
 Real growth             | record_dd3f83b0-f0ec-41a0-b048-83a7dd515191
|   4.24 |           7
 Crime                   | record_602a6db2-1784-45d2-8645-1c141f25a049
|   4.21 |           6
 News explain might turn | record_cbf50912-23ca-4de0-b94d-41b2076fd9d3
|   4.21 |           6
 Black movement          | record_acb14a89-3854-45de-9df7-ef1be87c7e47
|   4.17 |           5
 Water film              | record_0e4952cc-3445-4fe6-80c9-867e98ae4233
|   4.16 |           8
 Be sign hair            | record_e8fa438d-3e01-40df-bd1c-5d7dadf98015
|   4.15 |          10
 Music religious charge  | record_3cd5b91e-f6c5-469a-8f92-98f85a1a7af7
|   4.14 |           7
 Change adult            | record_9388a7fd-5828-4438-82b1-46c13e63c6c2
|   4.12 |          17
 Sit pay political       | record_66f81619-1bf3-4784-bd94-9c3adeeb5785
|   4.11 |           6
(10 rows)

-- Problem L --
  -- Query against view:
select a.user_id, u.nickname, avg(rr.rating) as content_rating
from artists as a
join users as u on a.user_id = u.user_id
join records as r on a.user_id = r.artist_user_id
```

```
join RatedRecords as rr ON r.record_id = rr.record_id
group by a.user_id, u.nickname
having avg(rr.rating) >= 3.3;
                  user_id                  |    nickname    |
content_rating
-------------------------------------------+----------------
+-------------------
 user_f921401a-7991-4db2-9491-2cb32b4146db | paynedavid     |
3.4573236606157646
 user_827369a9-7b0e-4937-917e-632d1ed5620f | robertfigueroa |
3.3393895455471657
(2 rows)

EXPLAIN ANALYZE
select a.user_id, u.nickname, avg(rr.rating) as content_rating
from artists as a
join users as u on a.user_id = u.user_id
join records as r on a.user_id = r.artist_user_id
join RatedRecords as rr ON r.record_id = rr.record_id
group by a.user_id, u.nickname
having avg(rr.rating) >= 3.3;

QUERY PLAN
------------------------------------------------------------------------------
------------------------------------------------------------------------------
-------------------------------------------------------------
 HashAggregate  (cost=4643.62..4669.70 rows=580 width=96) (actual
time=27.754..27.779 rows=2 loops=1)
   Group Key: a.user_id, u.nickname
   Filter: (avg((COALESCE((sum((rv.rating *
((COALESCE(count(rl.user_id), '0'::bigint) + 1)))) /
NULLIF(sum(((COALESCE(count(rl.user_id), '0'::bigint) + 1))),
'0'::numeric)), '0'::numeric))) >= 3.3)
   Batches: 1  Memory Usage: 129kB
   Rows Removed by Filter: 114
   ->  Hash Join  (cost=4564.63..4630.58 rows=1739 width=96) (actual
time=27.044..27.587 rows=1000 loops=1)
         Hash Cond: (a.user_id = u.user_id)
         ->  Hash Join  (cost=4548.33..4609.68 rows=1739 width=96)
(actual time=27.012..27.417 rows=1000 loops=1)
               Hash Cond: (r.artist_user_id = a.user_id)
               ->  Hash Join  (cost=4523.71..4580.45 rows=1739
width=64) (actual time=26.994..27.292 rows=1000 loops=1)
                     Hash Cond: (r_1.record_id = r.record_id)
                     ->  HashAggregate  (cost=4447.58..4482.36
rows=1739 width=104) (actual time=26.869..27.044 rows=1000 loops=1)
                           Group Key: r_1.record_id
                           Batches: 1  Memory Usage: 577kB
                           ->  Hash Join  (cost=3988.90..4317.02
rows=13056 width=76) (actual time=24.060..25.728 rows=9499 loops=1)
```

```
                                  Hash Cond: (rv.record_id =
r_1.record_id)
                                  ->  HashAggregate
(cost=3912.77..4075.97 rows=13056 width=76) (actual
time=23.928..24.547 rows=9499 loops=1)
                                        Group Key: rv.review_id
                                        Batches: 1  Memory Usage:
1937kB
                                        ->  Hash Right Join
(cost=565.76..3335.49 rows=115456 width=100) (actual
time=1.383..14.768 rows=91523 loops=1)
                                              Hash Cond: (rl.review_id
= rv.review_id)
                                              ->  Seq Scan on
reviewlikes rl  (cost=0.00..2466.56 rows=115456 width=64) (actual
time=0.002..2.931 rows=91325 loops=1)
                                              ->  Hash
(cost=402.56..402.56 rows=13056 width=68) (actual time=1.372..1.372
rows=9499 loops=1)
                                                    Buckets: 16384
Batches: 1  Memory Usage: 1279kB
                                                    ->  Seq Scan on
reviews rv  (cost=0.00..402.56 rows=13056 width=68) (actual
time=0.002..0.602 rows=9499 loops=1)
                                  ->  Hash  (cost=54.39..54.39
rows=1739 width=64) (actual time=0.129..0.129 rows=1000 loops=1)
                                        Buckets: 2048  Batches: 1
Memory Usage: 107kB
                                        ->  Seq Scan on records r_1
(cost=0.00..54.39 rows=1739 width=64) (actual time=0.003..0.050
rows=1000 loops=1)
                  ->  Hash  (cost=54.39..54.39 rows=1739 width=64)
(actual time=0.123..0.123 rows=1000 loops=1)
                        Buckets: 2048  Batches: 1  Memory Usage:
132kB
                        ->  Seq Scan on records r
(cost=0.00..54.39 rows=1739 width=64) (actual time=0.007..0.058
rows=1000 loops=1)
            ->  Hash  (cost=16.50..16.50 rows=650 width=32) (actual
time=0.015..0.015 rows=116 loops=1)
                  Buckets: 1024  Batches: 1  Memory Usage: 17kB
                  ->  Seq Scan on artists a  (cost=0.00..16.50
rows=650 width=32) (actual time=0.002..0.007 rows=116 loops=1)
         ->  Hash  (cost=12.80..12.80 rows=280 width=64) (actual
time=0.031..0.031 rows=200 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 25kB
               ->  Seq Scan on users u  (cost=0.00..12.80 rows=280
width=64) (actual time=0.003..0.017 rows=200 loops=1)
 Planning Time: 0.172 ms
 Execution Time: 27.916 ms
```

```
(39 rows)

  -- Index DDL:
create index idx_records_rating on records (rating);
CREATE INDEX
  -- Query against materialized data:
select a.user_id, u.nickname, avg(r.rating) as content_rating
from artists as a
join users as u on a.user_id = u.user_id
join records as r on a.user_id = r.artist_user_id
group by a.user_id, u.nickname
having avg(r.rating) >= 3.3;
                  user_id                  |    nickname    |
content_rating
-------------------------------------------+----------------
+------------------
 user_f921401a-7991-4db2-9491-2cb32b4146db | paynedavid     |
3.4566666666666667
 user_827369a9-7b0e-4937-917e-632d1ed5620f | robertfigueroa |
3.3400000000000000
(2 rows)

EXPLAIN ANALYZE
select a.user_id, u.nickname, avg(r.rating) as content_rating
from artists as a
join users as u on a.user_id = u.user_id
join records as r on a.user_id = r.artist_user_id
group by a.user_id, u.nickname
having avg(r.rating) >= 3.3;
                                                                    QUERY PLAN
-------------------------------------------------------------------------------
-------------------------------------------------------------
 HashAggregate  (cost=100.72..115.72 rows=333 width=96) (actual
time=0.451..0.473 rows=2 loops=1)
   Group Key: a.user_id, u.nickname
   Filter: (avg(r.rating) >= 3.3)
   Batches: 1  Memory Usage: 129kB
   Rows Removed by Filter: 114
   ->  Hash Join  (cost=40.92..93.22 rows=1000 width=76) (actual
time=0.045..0.297 rows=1000 loops=1)
         Hash Cond: (a.user_id = u.user_id)
         ->  Hash Join  (cost=24.62..74.27 rows=1000 width=76) (actual
time=0.017..0.165 rows=1000 loops=1)
               Hash Cond: (r.artist_user_id = a.user_id)
               ->  Seq Scan on records r  (cost=0.00..47.00 rows=1000
width=44) (actual time=0.003..0.031 rows=1000 loops=1)
               ->  Hash  (cost=16.50..16.50 rows=650 width=32) (actual
time=0.013..0.013 rows=116 loops=1)
                     Buckets: 1024  Batches: 1  Memory Usage: 17kB
                     ->  Seq Scan on artists a  (cost=0.00..16.50
```

```
rows=650 width=32) (actual time=0.001..0.006 rows=116 loops=1)
         ->  Hash  (cost=12.80..12.80 rows=280 width=64) (actual
time=0.027..0.028 rows=200 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 25kB
               ->  Seq Scan on users u  (cost=0.00..12.80 rows=280
width=64) (actual time=0.002..0.014 rows=200 loops=1)
 Planning Time: 0.042 ms
 Execution Time: 0.481 ms
(18 rows)

  /*
     … brief performance discussion …
         Without Materialized Data (Using RatedRecords View), the
planning time is 2.533 ms and
         execution time is 52.630 ms.

         With Materialized Data and Index (Using Records Table
Directly), the planning time is
         1.907 ms and the execution time is only 1.526 ms.

         In conclusion, Using materialized data (with an indexed
rating column in Records) provides
         significantly faster and more scalable performance than
dynamically calculating ratings
         through the RatedRecords view.
  */
-- Problem M --
select coalesce(music_quality, 'ALL') as music_quality,
coalesce(device, 'ALL') as device,
count(*) as num_session
from sessions
group by rollup(music_quality, device)
order by num_session desc;
 music_quality |     device      | num_session
---------------+-----------------+-------------
 ALL           | ALL             |       50000
 lowest        | ALL             |        8432
 lossless      | ALL             |        8400
 normal        | ALL             |        8380
 low           | ALL             |        8358
 high          | ALL             |        8327
 Hi-Fi         | ALL             |        8103
 lowest        | mobile-app      |        2188
 normal        | desktop-browser |        2156
 high          | mobile-browser  |        2153
 normal        | mobile-app      |        2147
 lowest        | desktop-browser |        2126
 lossless      | desktop-app     |        2115
 lossless      | desktop-browser |        2112
 lossless      | mobile-app      |        2106
```

```
 low           | desktop-app     |        2102
 low           | mobile-app      |        2102
 Hi-Fi         | mobile-app      |        2100
 low           | desktop-browser |        2091
 normal        | mobile-browser  |        2087
 lowest        | mobile-browser  |        2075
 high          | desktop-app     |        2075
 lossless      | mobile-browser  |        2067
 low           | mobile-browser  |        2063
 high          | mobile-app      |        2051
 high          | desktop-browser |        2048
 lowest        | desktop-app     |        2043
 Hi-Fi         | mobile-browser  |        2026
 Hi-Fi         | desktop-app     |        2003
 normal        | desktop-app     |        1990
 Hi-Fi         | desktop-browser |        1974
(31 rows)
```