OPTIMIZATION STEPS:

- PROJECTION & SELECTION: like in Relation Algebra, push projection & selection inside each table as much as possible before the JOINs. (Use WITH…AS to make new tables before the JOINs.)
- JOIN ORDER: use Smaller table on the outside.
- SET INDEX: set index on attributes in WHERE clause after the JOINs.
- RECORD TIME: use 'analyze' after 'explain'

    EXAMPLE:

        explain analyze select count(*)
        from Artists
        where name like '%A%';

naming convention for creating an index:

```
I_table_name_column_name
```

create an index:

```
CREATE INDEX index_name
    [index_type]
    ON tbl_name (key_part,...)


key_part: {col_name [(length)] | (expr)} [ASC | DESC]

index_type:
    USING {BTREE | HASH}
```

Example:

```
CREATE INDEX I_Artists_name
   USING BTREE
   ON Artists (name(255))
```

https://dev.mysql.com/doc/refman/8.0/en/create-index.html

drop an index:

```
DROP INDEX index_name ON table_name
```

https://dev.mysql.com/doc/refman/5.6/en/drop-index.html

INDICES:

We added three B+Tree indices in our database for the query optimization tasks:

```
CREATE INDEX I_Artists_name
    USING BTREE
    ON Artists (name(255));


CREATE INDEX I_Tracks_name
   USING BTREE
   ON Tracks (name(540));


CREATE INDEX I_Albums_artist_id
   USING BTREE
   ON Albums (artist_id(50));
```

ARIANNA:

Hasun Query 1: search by artist name & track name

Return 15,673 rows.

OLD QUERY:

```
SELECT T.id AS TrackId, T.name AS TrackName, T.preview_url AS Preview,
        AL.name AS AlbumName, AL.images AS Cover, A.name AS ArtistName,
        A.genres AS Genre
FROM Albums AL
JOIN Artists A ON AL.artist_id = A.id
JOIN Tracks T ON AL.id = T.album_id
WHERE LOWER(A.name) LIKE '%${ArtistName}%'
AND LOWER(T.name) LIKE '%${TrackName}%'
ORDER BY A.name, T.name
```

NEW QUERY:

```
WITH New_Artists AS (
    SELECT id, name, genres
    FROM Artists
    WHERE LOWER(name) LIKE '%A%'
), New_Albums AS (
    SELECT id, artist_id, name, images
    FROM Albums
), New_Tracks AS (
    SELECT id, album_id, name, preview_url
    FROM Tracks
    WHERE LOWER(name) LIKE '%B%'
)
SELECT T.id AS TrackId, T.name AS TrackName, T.preview_url AS Preview,
        AL.name AS AlbumName, AL.images AS Cover, A.name AS ArtistName,
        A.genres AS Genre
FROM New_Artists A
JOIN New_Albums AL ON AL.artist_id = A.id
JOIN New_Tracks T ON AL.id = T.album_id
ORDER BY A.name, T.name;
```

TIME:

OLD QUERY:

[2022-04-26 09:09:21] 15,673 rows retrieved starting from 1 in 3 s 18 ms (execution: 926 ms, fetching: 2 s 92 ms)
[2022-04-26 09:10:01] 15,673 rows retrieved starting from 1 in 2 s 926 ms (execution: 816 ms, fetching: 2 s 110 ms)
[2022-04-26 09:10:22] 15,673 rows retrieved starting from 1 in 3 s 90 ms (execution: 832 ms, fetching: 2 s 258 ms)
AVERAGE EXECUTION TIME: **858 ms**

NEW QUERY:

[2022-04-26 09:11:55] 15,673 rows retrieved starting from 1 in 2 s 643 ms (execution: 884 ms, fetching: 1 s 759 ms)
[2022-04-26 09:12:24] 15,673 rows retrieved starting from 1 in 3 s 692 ms (execution: 844 ms, fetching: 2 s 848 ms)
[2022-04-26 09:13:29] 15,673 rows retrieved starting from 1 in 2 s 847 ms (execution: 826 ms, fetching: 2 s 21 ms)
AVERAGE EXECUTION TIME: **852 ms**

EXPLAIN:

OLD QUERY:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	AL		ALL	PRIMARY,artist_id				66524	100	Using temporary; Using filesort
1	SIMPLE	A		eq_ref	PRIMARY	PRIMARY	202	music_app.AL.artist_id	1	100	Using where
1	SIMPLE	T		ref	album_id	album_id	202	music_app.AL.id	1	100	Using where

NEW QUERY:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Albums		ALL	PRIMARY,artist_id,I_Albums_artist_id				66524	100	Using temporary; Using filesort
1	SIMPLE	Artists		eq_ref	PRIMARY	PRIMARY	202	music_app.Albums.artist_id	1	100	Using where
1	SIMPLE	Tracks		ref	album_id	album_id	202	music_app.Albums.id	1	100	Using where

ANALYZE:

OLD QUERY:

-> Sort: A.`name`, T.`name` (actual time=2016.704..2044.932 rows=15673 loops=1)
    -> Stream results (cost=61971.42 rows=88895) (actual time=0.140..1915.244 rows=15673 loops=1)
        -> Nested loop inner join (cost=61971.42 rows=88895) (actual time=0.135..1873.680 rows=15673 loops=1)
            -> Nested loop inner join (cost=30858.05 rows=66524) (actual time=0.079..1020.786 rows=53089 loops=1)
                -> Table scan on AL (cost=7574.65 rows=66524) (actual time=0.029..147.453 rows=74991 loops=1)
                -> Filter: (lower(A.`name`) like '%A%') (cost=0.25 rows=1) (actual time=0.007..0.009 rows=1 loops=74991)
                    -> Single-row index lookup on A using PRIMARY (id=AL.artist_id) (cost=0.25 rows=1) (actual time=0.004..0.004 rows=1 loops=74991)
            -> Filter: (lower(T.`name`) like '%B%') (cost=0.33 rows=1) (actual time=0.012..0.014 rows=0 loops=53089)
                -> Index lookup on T using album_id (album_id=AL.id) (cost=0.33 rows=1) (actual time=0.007..0.009 rows=1 loops=53089)

NEW QUERY:

-> Sort: A.ArtistName, T.TrackName (actual time=2016.454..2045.409 rows=15673 loops=1)
    -> Stream results (cost=61971.42 rows=88895) (actual time=0.189..1906.423 rows=15673 loops=1)
        -> Nested loop inner join (cost=61971.42 rows=88895) (actual time=0.183..1854.189 rows=15673 loops=1)
            -> Nested loop inner join (cost=30858.05 rows=66524) (actual time=0.126..1011.676 rows=53089 loops=1)
                -> Table scan on Albums (cost=7574.65 rows=66524) (actual time=0.029..144.330 rows=74991 loops=1)
                -> Filter: (lower(Artists.`name`) like '%A%') (cost=0.25 rows=1) (actual time=0.007..0.008 rows=1 loops=74991)
                    -> Single-row index lookup on Artists using PRIMARY (id=Albums.artist_id) (cost=0.25 rows=1) (actual time=0.003..0.004 rows=1 loops=74991)
            -> Filter: (lower(Tracks.`name`) like '%B%') (cost=0.33 rows=1) (actual time=0.012..0.013 rows=0 loops=53089)
                -> Index lookup on Tracks using album_id (album_id=Albums.id) (cost=0.33 rows=1) (actual time=0.007..0.009 rows=1 loops=53089)

Hasun Query 2: get one song

Return 1 row.

OLD QUERY:

```
SELECT T.id AS TrackId, T.name AS TrackName, T.preview_url AS Preview,
            AL.name AS AlbumName, AL.images AS Cover, A.name AS ArtistName,
            A.genres AS Genre
FROM Albums AL
JOIN Artists A ON AL.artist_id = A.id
JOIN Tracks T ON AL.id = T.album_id
WHERE T.id = '7nSDE1ceSRoUryWMiCnBXZ';
```

NEW QUERY:

```
WITH New_Artists AS (
  SELECT id, name, genres
  FROM Artists
), New_Albums AS (
  SELECT id, artist_id, name, images
  FROM Albums
), New_Tracks AS (
  SELECT id, album_id, name, preview_url
  FROM Tracks
  WHERE id = '7nSDE1ceSRoUryWMiCnBXZ'
)
SELECT T.id AS TrackId, T.name AS TrackName, T.preview_url AS Preview,
     AL.name AS AlbumName, AL.images AS Cover, A.name AS ArtistName,
     A.genres AS Genre
FROM New_Artists A
JOIN New_Albums AL ON AL.artist_id = A.id
JOIN New_Tracks T ON AL.id = T.album_id;
```

TIME:

OLD QUERY:

[2022-04-26 09:22:10] 1 row retrieved starting from 1 in 111 ms (execution: 82 ms, fetching: 29 ms)
[2022-04-26 09:22:33] 1 row retrieved starting from 1 in 112 ms (execution: 87 ms, fetching: 25 ms)
[2022-04-26 09:22:52] 1 row retrieved starting from 1 in 120 ms (execution: 90 ms, fetching: 30 ms)
AVERAGE EXECUTION TIME: **87ms**

NEW QUERY:

[2022-04-26 09:24:18] 1 row retrieved starting from 1 in 117 ms (execution: 84 ms, fetching: 33 ms)
[2022-04-26 09:24:52] 1 row retrieved starting from 1 in 109 ms (execution: 90 ms, fetching: 19 ms)
[2022-04-26 09:25:13] 1 row retrieved starting from 1 in 99 ms (execution: 87 ms, fetching: 12 ms)
AVERAGE EXECUTION TIME: **87 ms**

EXPLAIN:

OLD QUERY:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	T		const	PRIMARY,album_id	PRIMARY	202	const	1	100	
1	SIMPLE	AL		const	PRIMARY,artist_id	PRIMARY	202	const	1	100	
1	SIMPLE	A		const	PRIMARY	PRIMARY	202	const	1	100	

NEW QUERY:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Tracks		const	PRIMARY,album_id	PRIMARY	202	const	1	100	
1	SIMPLE	Albums		const	PRIMARY,artist_id,I_Albums_artist_id	PRIMARY	202	const	1	100	
1	SIMPLE	Artists		const	PRIMARY	PRIMARY	202	const	1	100	

ANALYZE:

OLD QUERY:

> Rows fetched before execution (cost=0.00 rows=1) (actual time=0.002..0.003 rows=1 loops=1)

NEW QUERY:

-> Rows fetched before execution (cost=0.00 rows=1) (actual time=0.002..0.002 rows=1 loops=1)

Spencer Query 1: word cloud

Return 49 rows.

OLD QUERY:

```
WITH Content AS (
    SELECT id, name, album_type
    FROM Albums
    WHERE artist_id IN
       ( SELECT id
         FROM Artists A
         WHERE LOWER(name) LIKE '%$A%')
)
SELECT C.name AS AlbumName,
     C.album_type AS AlbumType,
     T.name AS TrackName,
     T.popularity AS Popularity
FROM Content C
JOIN Tracks T ON T.album_id = C.id
ORDER BY T.popularity DESC;
```

NEW QUERY:

```
WITH Content AS (
    SELECT id, name, album_type
    FROM Albums
    WHERE artist_id IN
       ( SELECT id
         FROM Artists A
         WHERE LOWER(name) LIKE '%$A%')
), New_Tracks AS (
  SELECT album_id, name, popularity
  FROM Tracks
)
SELECT C.name AS AlbumName,
     C.album_type AS AlbumType,
     T.name AS TrackName,
     T.popularity AS Popularity
FROM Content C
JOIN New_Tracks T ON T.album_id = C.id
ORDER BY T.popularity DESC;
```

TIME:

OLD QUERY:

[2022-04-26 09:43:43] 49 rows retrieved starting from 1 in 286 ms (execution: 250 ms, fetching: 36 ms)
[2022-04-26 09:43:58] 49 rows retrieved starting from 1 in 275 ms (execution: 258 ms, fetching: 17 ms)
[2022-04-26 09:44:13] 49 rows retrieved starting from 1 in 271 ms (execution: 253 ms, fetching: 18 ms)
AVERAGE EXECUTION TIME: **254 ms**

NEW QUERY:

[2022-04-26 09:44:47] 49 rows retrieved starting from 1 in 288 ms (execution: 254 ms, fetching: 34 ms)
[2022-04-26 09:45:01] 49 rows retrieved starting from 1 in 273 ms (execution: 255 ms, fetching: 18 ms)
[2022-04-26 10:09:52] 49 rows retrieved starting from 1 in 278 ms (execution: 250 ms, fetching: 28 ms)
AVERAGE EXECUTION TIME: **253 ms**

EXPLAIN:

OLD QUERY:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Albums		ALL	PRIMARY,artist_id				66524	100	Using temporary; Using filesort
1	SIMPLE	A		eq_ref	PRIMARY	PRIMARY	202	music_app.Albums.artist_id	1	100	Using where
1	SIMPLE	T		ref	album_id	album_id	202	music_app.Albums.id	1	100	

NEW QUERY:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Albums		ALL	PRIMARY,artist_id,I_Albums_artist_id				66524	100	Using temporary; Using filesort
1	SIMPLE	A		eq_ref	PRIMARY	PRIMARY	202	music_app.Albums.artist_id	1	100	Using where
1	SIMPLE	Tracks		ref	album_id	album_id	202	music_app.Albums.id	1	100	

ANALYZE:

OLD QUERY:

-> Sort: T.popularity DESC (actual time=798.155..798.191 rows=49 loops=1)
  -> Stream results (cost=61971.42 rows=88895) (actual time=22.347..798.028 rows=49 loops=1)
    -> Nested loop inner join (cost=61971.42 rows=88895) (actual time=22.340..797.851 rows=49 loops=1)
      -> Nested loop inner join (cost=30858.05 rows=66524) (actual time=22.308..797.059 rows=30 loops=1)
        -> Table scan on Albums (cost=7574.65 rows=66524) (actual time=0.054..85.029 rows=74991 loops=1)
        -> Filter: (lower(A.`name`) like '%$A%') (cost=0.25 rows=1) (actual time=0.007..0.007 rows=0 loops=74991)
          -> Single-row index lookup on A using PRIMARY (id=Albums.artist_id) (cost=0.25 rows=1) (actual time=0.003..0.004 rows=1 loops=74991)
      -> Index lookup on T using album_id (album_id=Albums.id) (cost=0.33 rows=1) (actual time=0.015..0.018 rows=2 loops=30)

NEW QUERY:

-> Sort: T.Popularity DESC (actual time=786.388..786.423 rows=49 loops=1)
  -> Stream results (cost=61971.42 rows=88895) (actual time=22.287..786.266 rows=49 loops=1)
    -> Nested loop inner join (cost=61971.42 rows=88895) (actual time=22.280..786.068 rows=49 loops=1)
      -> Nested loop inner join (cost=30858.05 rows=66524) (actual time=22.249..785.397 rows=30 loops=1)
        -> Table scan on Albums (cost=7574.65 rows=66524) (actual time=0.068..84.133 rows=74991 loops=1)
        -> Filter: (lower(A.`name`) like '%$A%') (cost=0.25 rows=1) (actual time=0.007..0.007 rows=0 loops=74991)
          -> Single-row index lookup on A using PRIMARY (id=Albums.artist_id) (cost=0.25 rows=1) (actual time=0.003..0.004 rows=1 loops=74991)
      -> Index lookup on Tracks using album_id (album_id=Albums.id) (cost=0.33 rows=1) (actual time=0.014..0.017 rows=2 loops=30)

NIDHI: