

Show all data from table 'lv1\_tracks' that duration is greater than 200000 and duration is less than 350000 and release year is 2020 and sort by 'name' in descending order and limit to 10 rows and offset to 3 rows.

SHOW HINT

Passed

Actual result

id	spotify_id	name	album	artist	artwork_url	duration
----	------------	------	-------	--------	-------------	----------

Show all rows and all columns from table 'lv1\_tracks' that popularity is greater than or equal 40 and explicit is 1

SHOW HINT

Passed

Actual result

id	spotify_id	name	album	artist	artwork_url	dura
----	------------	------	-------	--------	-------------	------

Show name of first 10 rows from table 'lv1\_tracks' with a column named 'track\_name' instead of name.

Show name of first 10 rows from table 'lv1\_tracks' with a column named 'track\_name' instead of name.

LEVEL LV1  
DIFFICULTY MID

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR select name as "track_name" from lv1_tracks limit 10
```

Passed

Expected result

track_name
青春コンプレックス
ギターと孤独と蒼い惑星
星座になれたら
あのバンド

Actual result

track_name
青春コンプレックス
ギターと孤独と蒼い惑星
星座になれたら
あのバンド

Show all tracks that track name start with 'A'

Show all tracks that track name start with 'A' and let the result be sorted by default order.

DIFFICULTY EASY  
LEVEL LV1

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT * from lv1_tracks WHERE name LIKE 'A%'
```

Passed

Expected result

id	spotify_id	name	album
140	0iuGNjRPERARznf8iCQJ6a	And That's Why I Can't Stop!	And That's Why I Can't Stop!
239	5cBmwNnKGLPRCQWzFS0r0Y	ALIVE	ALIVE

Actual result

id	spotify_id	name
140	0iuGNjRPERARznf8iCQJ6a	And That's Why I Can't Stop!
239	5cBmwNnKGLPRCQWzFS0r0Y	ALIVE

Show all tracks that track name start with 'A' or 'J'

Show all tracks that track name start with 'A' or 'J' and let the result be sorted by default order.

DIFFICULTY **EASY**

LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT * from lv1_tracks WHERE name LIKE 'A%' or name LIKE 'J%'
```

Passed

Expected result

id	spotify_id	name	album
101	3UCbRuJbsnEZeGLXefqeow	Just a Little Step	E-SIDE 2
128	1hPFCT7wqA0TOIby783Rd9	Judgement	Judgement

Actual result

id	spotify_id	name
101	3UCbRuJbsnEZeGLXefqeow	Just a Little Step
128	1hPFCT7wqA0TOIby783Rd9	Judgement
		And That's

Show all tracks that track name contains 'A' in the track name

Show all tracks that track name contains 'A' in the track name and let the result be sorted by default order.

DIFFICULTY **EASY**

LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT * from lv1_tracks WHERE name LIKE '%A%'
```

Passed

Expected result

id	spotify_id	name	album
88	5ASNsnnOvreLu9FtvA3jpR	If I Could Draw Life	E-SIDE 2
91	6Kh3obD8LG4yeXxQDhr6Vr	Romance	E-SIDE 2
94	6oexafIxFkZaEEcfJCsx4y	The Swallow (feat. Midories)	E-SIDE 2

Actual result

id	spotify_id	name
88	5ASNsnnOvreLu9FtvA3jpR	If I Could Draw Life
91	6Kh3obD8LG4yeXxQDhr6Vr	Romance
94	6oexafIxFkZaEEcfJCsx4y	The Swallow (feat. Midories)

Query all rows with only 2 column, track name and track name in lowercase

Let's query using function LOWER() to make data be lowercase. For example, `SELECT name, LOWER(name) FROM lv1_tracks`

LEVEL LV1

DIFFICULTY MID

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT name, LOWER(name) FROM lv1_tracks
```

Passed

Expected result

name	LOWER(name)
青春コンプレックス	青春コンプレックス
ギターと孤独と蒼い惑星	ギターと孤独と蒼い惑星
星座になればたら	星座になればたら
あのバンド	あのバンド

Actual result

name	LOWER(name)
青春コンプレックス	青春コンプレックス
ギターと孤独と蒼い惑星	ギターと孤独と蒼い惑星
星座になればたら	星座になればたら
あのバンド	あのバンド

Show all tracks that track name contains 'A' in the track name with case insensitive (contains 'a' or 'A')

Try using function LOWER() to make data be lowercase. So character 'A' and 'a' will be treated as the same and we can use LIKE condition to query data.

DIFFICULTY MID

LEVEL LV1

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT * from lv1_tracks WHERE name LIKE '%A%'
```

Passed

Expected result

id	spotify_id	name	album
88	5ASNsnnOvreLu9FtvA3jpR	If I Could Draw Life	E-SIDE 2
91	6Kh3obD8LG4yeXxQDhr6Vr	Romance	E-SIDE 2
		The Swallow	

Actual result

id	spotify_id	name
88	5ASNsnnOvreLu9FtvA3jpR	If I Could Draw Life
91	6Kh3obD8LG4yeXxQDhr6Vr	Romance
		The Swallow

Show data that contains 'THE' (case-insensitive) in the track name or album name or artist name

Try using function LOWER() to make data be lowercase. Then, use LIKE condition to filter data and use OR operator to combine multiple conditions.

DIFFICULTY **MID**  
LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT * FROM lv1_tracks WHERE LOWER(name) LIKE '%the%' OR LOWER(album) LIKE '%the%' OR LOWER(artist) LIKE '%the%'
```

Passed

Expected result

id	spotify_id	name	album	artist
47	6MCjmGYlw6mQVWRFVgBRvB	夜に駆ける	THE BOOK	YOASOBI
58	1zd35Y44Blc1CwwVbW3Qnk	群青	THE BOOK	YOASOBI

Actual result

id	spotify_id	name
47	6MCjmGYlw6mQVWRFVgBRvB	夜に駆ける
58	1zd35Y44Blc1CwwVbW3Qnk	群青
62	2vjY3jGKElvqXoaGNEuYef	怪物

Show data of first 10 rows from table 'lv1\_tracks' with a column named 'track\_name' instead of name and a column named 'track\_duration' instead of duration.

Show name of first 10 rows from table 'lv1\_tracks' with a column named 'track\_name' instead of name and a column named 'track\_duration' instead of duration.

DIFFICULTY **MID**  
LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT name as track_name, duration as track_duration from lv1_tracks LIMIT 10
```

Passed

Expected result

track_name	track_duration
青春コンプレックス	203626
ギターと孤独と蒼い惑星	228960
星座になれたら	258173

Actual result

track_name	track_duration
青春コンプレックス	203626
ギターと孤独と蒼い惑星	228960
星座になれたら	258173

Show all columns from table 'lv1\_tracks' but change only column named 'name' to 'track\_name'.

Show all columns from table 'lv1\_tracks' but change only column named 'name' to 'track\_name'.

DIFFICULTY MID  
LEVEL LV1

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR select name as track_name, id ,spotify_id, name, album, artist, artwork_url, duration, popularity, explicit, year, count from lv1_tracks
```

Passed

Expected result

track_name	id	spotify_id	name
青春コンプレックス	1	60nwK1iMgnFCznF6FINfts	青春コンプレックス
ギターと孤独と蒼い惑星	4	1rgncjmlpHMP3DBhpagyVb	ギターと孤独と蒼い惑星
星座になれたら	8	1iNhNmEwrd2TP4Xrv7pQBI	星座になれたら

Actual result

track_name	id	spotify_id
青春コンプレックス	1	60nwK1iMgnFCznF6FINfts
ギターと孤独と蒼い惑星	4	1rgncjmlpHMP3DBhpagyVb
星座になれたら	8	1iNhNmEwrd2TP4Xrv7pQBI

Find out the year of the oldest song.

Find out the year of the oldest song.

DIFFICULTY MID  
LEVEL LV1

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT MIN(year) FROM lv1_tracks
```

Passed

Expected result

MIN(year)
2012

Actual result

MIN(year)
2012

Find out 2 value of the oldest year and the newest year.

Find out 2 value of the oldest year and the newest year.

LEVEL **LV1**  
DIFFICULTY **MID**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT MIN(year), MAX(year) FROM lv1_tracks
```

Passed

Expected result

MIN(year)	MAX(year)
2012	2023

Actual result

MIN(year)	MAX(year)
2012	2023

Find out the oldest year and the newest year of tracks with custom column name.

Find out the oldest year and the newest year of tracks but change output column name of MIN(year) to 'oldest\_year' and change output column name of MAX(year) to 'newest\_year'.

DIFFICULTY **MID**  
LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT MIN(year) as oldest_year, MAX(year) as newest_year FROM lv1_tracks
```

Passed

Expected result

oldest_year	newest_year
2012	2023

Actual result

oldest_year	newest_year
2012	2023

Count all data from table 'lv1\_tracks' without any condition.

Count all rows and all columns from table 'lv1\_tracks' without any condition.

DIFFICULTY **MID**

LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR select COUNT(*) from lv1_tracks
```

Passed

Expected result

COUNT(\*)

95

Actual result

COUNT(\*)

95

Count all data from table 'lv1\_tracks' and change the output column name to 'total\_tracks'.

Count all rows and all columns from table 'lv1\_tracks' and change the output column name to 'total\_tracks'.

DIFFICULTY **MID**

LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR select COUNT(*) as total_tracks from lv1_tracks
```

Passed

Expected result

total\_tracks

95

Actual result

total\_tracks

95



```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT year, COUNT(*) AS total_tracks FROM lv1_tracks GROUP BY year
```

Expected result

COUNT(*)
3
1
2
3
2
1
11
50
22

Actual result

year	total_tracks
2012	3
2013	1
2014	2
2015	3
2018	2
2020	1
2021	11
2022	50
2023	22

Count all data from table 'lv1\_tracks' released in each year.

Count all rows and all columns from table 'lv1\_tracks' that released in each year into 2 columns named 'year' and 'total\_tracks'.

DIFFICULTY **MID**  
LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT COUNT(*) FROM lv1_tracks GROUP BY year
```

Passed

Expected result

COUNT(*)
3
1
2
3

Actual result

COUNT(*)
3
1
2
3

Count all data from table 'lv1\_tracks' released in each year and change the output column name to 'total\_tracks'.

Count all rows and all columns from table 'lv1\_tracks' that released in each year into 2 columns named 'year' and 'total\_tracks'.

DIFFICULTY **MID**

LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT COUNT(*) as total_tracks FROM lv1_tracks GROUP BY year
```

Passed

Expected result

total_tracks
3
1
2

Actual result

total_tracks
3
1
2

Sum the total duration of all tracks in table 'lv1\_tracks' released in each year.

Sum the total duration of all tracks in table 'lv1\_tracks' released in each year into 2 columns named 'year' and 'total\_duration'.

LEVEL **LV1**

DIFFICULTY **MID**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT SUM(duration) AS total_duration FROM lv1_tracks GROUP BY year
```

Passed

Expected result

total_duration
790265
251413
542520
808145

Actual result

total_duration
790265
251413
542520
808145

Sort the year that have the most number of tracks in table 'lv1\_tracks' from highest to lowest.

Sort the year that have the most number of tracks in table 'lv1\_tracks' from highest to lowest. If there are 2 years that have the same number of tracks, sort them by year from lowest to highest.

LEVEL **LV1**

DIFFICULTY **MID**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT year, COUNT(*) AS total_tracks FROM lv1_tracks GROUP BY year
ORDER BY total_tracks desc, year ASC
```

Passed

Expected result

year	total_tracks
2022	50
2023	22
2021	11

Actual result

year	total_tracks
2022	50
2023	22
2021	11

Sort data from both column, 'total\_tracks' and 'total\_duration' from highest to lowest.

Show all rows from table 'lv1\_tracks' by sort all columns from table 'lv1\_tracks' by release year from highest to lowest, if year is the same, sort by duration from shortest to longest.

DIFFICULTY **MID**

LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT year, COUNT(*) AS total_tracks, SUM(duration) AS total_duration
FROM lv1_tracks
GROUP BY year
ORDER BY total_tracks DESC,
```

Passed

Expected result

year	total_tracks	total_duration
2022	50	11079420
2023	22	4800530
2021	11	2334915
2012	3	790265

Actual result

year	total_tracks	total_duration
2022	50	11079420
2023	22	4800530
2021	11	2334915
2012	3	790265

Display the data information into the given format 1.

Display the data information into the given format (try query anything for system to show the expected output).

DIFFICULTY **HIGH**  
LEVEL **LV1**

SHOW

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT artist, COUNT(*) AS total_tracks, SUM(duration) AS total_duration FROM lv1_tracks GROUP BY artist ORDER BY total_trac
```

Passed

Expected result

artist	total_tracks	total_duration
KANA-BOON	1	197944
PEOPLE 1	1	216093
ASH DA HERO	1	226240
BiSH	1	211911
サトシ(CV:松本梨香)	1	248738
ASCA	1	255813

Actual result

artist	total_tracks	total_duration
KANA-BOON	1	197944
PEOPLE 1	1	216093
ASH DA HERO	1	226240
BiSH	1	211911
サトシ(CV:松本梨香)	1	248738
ASCA	1	255813

Display the data information into the given format 2.

Display the data information into the given format (try query anything for system to show the expected output).

DIFFICULTY **HIGH**  
LEVEL **LV1**

SHOW HIT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT artist, album, COUNT(*) AS total_tracks, SUM(duration) AS total_duration FROM lv1_tracks GROUP BY artist, album ORDER BY total_tracks
```

Passed

Expected result

artist	album	total_tracks	total_duration
サトシ(CV:松本梨香)	めざせポケモンマスター -with my friends - from THE FIRST TAKE	1	248738
YOASOBI	アドベンチャー	1	199964
ClariS	SECOND STORY	1	251413
TUYU	It Might Be Painful, but I Still Love It.	1	204999
結束バンド	星座になればたら	1	258173
結束バンド	ギターと孤独と蒼い惑星	1	228960

Actual result

artist	album	total_tracks	total_duration
サトシ(CV:松本梨香)	めざせポケモンマスター -with my friends - from THE FIRST TAKE	1	248738
YOASOBI	アドベンチャー	1	199964
ClariS	SECOND STORY	1	251413
TUYU	It Might Be Painful, but I Still Love It.	1	204999
結束バンド	星座になればたら	1	258173
結束バンド	ギターと孤独と蒼い惑星	1	228960

Display the data information into the given format 3.

Display the data information into the given format (try query anything for system to show the expected output).

DIFFICULTY **HIGH**  
LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT album, COUNT(*) AS total_tracks, AVG(duration) AS average_duration FROM lv1_tracks GROUP BY album ORDER BY average_duration
```

Passed

Expected result

album	total_tracks	average_duration
我武者羅 (TV size)	1	90000.0000
アイタリナイ	1	162366.0000
And That's Why I Can't Stop!	1	165454.0000
ALIVE (Lycoreco Version)	1	166332.0000
TVアニメ『SPY×FAMILY』オリジナル・サウンドトラック	2	178316.5000
TVアニメ『SPY×FAMILY』オリジナル・サウンドトラック (Vol.1)	2	190416.5000

Actual result

album	total_tracks	average_duration
我武者羅 (TV size)	1	90000.0000
アイタリナイ	1	162366.0000
And That's Why I Can't Stop!	1	165454.0000
ALIVE (Lycoreco Version)	1	166332.0000
TVアニメ『SPY×FAMILY』オリジナル・サウンドトラック	2	178316.5000
TVアニメ『SPY×FAMILY』オリジナル・サウンドトラック (Vol.1)	2	190416.5000

Display the data information into the given format 4.

Display the data information into the given format (try query anything for system to show the expected output) with explicit track percentage in each album.

DIFFICULTY **HIGH**  
LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT album, (SELECT COUNT(*) FROM lv1_tracks t1 WHERE t1.album = lv1_tracks.album AND t1.explicit = 1) AS explicit_tracks_count, (SELECT COUNT(*) FROM lv1_tracks t2 WHERE t2.album = lv1_tracks.album) AS total_tracks, (SELECT COUNT(*) FROM lv1_tracks t3 WHERE t3.album = lv1_tracks.album AND t3.explicit = 1) / (SELECT COUNT(*) FROM lv1_tracks t4 WHERE t4.album = lv1_tracks.album) AS explicit_percentage FROM lv1_tracks GROUP BY album ORDER BY explicit_percentage
```

Passed

Expected result

album	explicit_tracks_count	total_tracks	explicit_percentage
祝福	0	1	0.0000
slash	0	1	0.0000
It Might Be Painful, but I Still Love It.	0	1	0.0000
Fairy Party	0	2	0.0000
星座になれたら	0	1	0.0000
THE BOOK 2	0	6	0.0000

Actual result

album	explicit_tracks_count	total_tracks	explicit_percentage
祝福	0	1	0.0000
slash	0	1	0.0000
It Might Be Painful, but I Still Love It.	0	1	0.0000
Fairy Party	0	2	0.0000
星座になれたら	0	1	0.0000
THE BOOK 2	0	6	0.0000

Display the data information into the given format 5.

Select artist that have most number of released tracks in each year with limit only 1 artist per year.

DIFFICULTY **HIGH**  
LEVEL **LV1**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT year, (SELECT artist FROM lv1_tracks WHERE year = lv1_tracks.year GROUP BY artist ORDER BY COUNT(*) DESC LIMIT 1) AS top_artist_name_by_track_count FROM lv1_tracks GROUP BY year ORDER BY 'year'
```

Passed

Expected result

year	top_artist_name_by_track_count
2022	結束バンド
2021	結束バンド
2023	結束バンド
2015	結束バンド
2020	結束バンド
2012	結束バンド
2018	結束バンド

Actual result

year	top_artist_name_by_track_count
2022	結束バンド
2021	結束バンド
2023	結束バンド
2015	結束バンド
2020	結束バンド
2012	結束バンド
2018	結束バンド

Show album information from the track previously queried.

From the last task, you should have a query result with album\_id of the first track in table 'lv2\_tracks'. Use that album\_id to query album with that specific id from table 'lv2\_albums'.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT * FROM lv2_albums WHERE id = 1
```

Passed

Expected result

id	spotify_id	name	artist_id	artwork_url
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	https://i.scdn.co/image/ab67616d0000t

Actual result

id	spotify_id	name	artist_id	artwork
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	https://

Show all track information with linked album information.

Use MySQL JOIN statement to query the track information with linked album information by using ON lv2\_tracks.album\_id = lv2\_albums.id to join the two tables.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT * FROM lv2_tracks WHERE album_id = (SELECT album_id FROM lv2_tracks LIMIT 1)
```

Passed

Expected result

id	spotify_id	name	album_id	duration	popularity	expli
1	60nwK1lMgnFCznF6fINfts	青春コンプレックス	1	203626	66	0
4	1rgncjmlpHMP3DBhpagyVb	ギターと孤独と甘い惑星	1	228960	66	0
8	1iNhNmEwrd2TP4Xrv7pQBI	星座になればたら	1	258173	67	0
12	4Ji17AjAdjf83FslDcXe0J	あのバンド	1	213786	63	0

Actual result

id	spotify_id	name	album_id	duration
1	60nwK1lMgnFCznF6fINfts	青春コンプレックス	1	203626
4	1rgncjmlpHMP3DBhpagyVb	ギターと孤独と甘い惑星	1	228960
8	1iNhNmEwrd2TP4Xrv7pQBI	星座になればたら	1	258173
12	4Ji17AjAdjf83FslDcXe0J	あのバンド	1	213786
16	5ISHFvPLUqKz2JfDRtwnb2	忘れてやらない	1	223360

Show track name and album name of all data.

Use MySQL JOIN statement to query the track name and album name of all data by using ON lv2\_tracks.album\_id = lv2\_albums.id to join the two tables.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT lv2_tracks.name, lv2_albums.name  
FROM lv2_tracks  
JOIN lv2_albums ON lv2_tracks.album_id = lv2_albums.id
```

Passed

Expected result

name	name
青春コンプレックス	結束バンド
ギターと孤独と甘い惑星	結束バンド
星座になればたら	結束バンド
あのバンド	結束バンド
忘れてやらない	結束バンド
カラカラ	結束バンド

Actual result

name	name
青春コンプレックス	結束バンド
ギターと孤独と甘い惑星	結束バンド
星座になればたら	結束バンド
あのバンド	結束バンド
忘れてやらない	結束バンド
カラカラ	結束バンド

Show track name and album name of all data with track name starting with 'A'.

Use MySQL JOIN statement to query the track name and album name of all data with track name starting with 'A' by using ON lv2\_tracks.album\_id = lv2\_albums.id to join the two tables. And mix it with WHERE statement to filter the result.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT lv2_tracks.name, lv2_albums.name
FROM lv2_tracks
JOIN lv2_albums ON lv2_tracks.album_id = lv2_albums.id
WHERE lv2_tracks.name LIKE 'A%'
```

Passed

Expected result

name	name
And That's Why I Can't Stop!	And That's Why I Can't Stop!
ALIVE	ALIVE
ALIVE - LycoReco Version	ALIVE (LycoReco Version)
again (ClariS 1st 武道館コンサート) [Live]	ClariS 1st 武道館コンサート~2つの仮面と失われた太陽 ~

Actual result

name	name
And That's Why I Can't Stop!	And That's Why I Can't Stop!
ALIVE	ALIVE
ALIVE - LycoReco Version	ALIVE (LycoReco Version)
again (ClariS 1st 武道館コンサート) [Live]	ClariS 1st 武道館コンサート~2つの仮面と失われた太陽 ~

Show track name, album name and artist name of all data

Use MySQL JOIN statement to query the track name, album name and artist name of all data by using ON lv2\_tracks.album\_id = lv2\_albums.id to join the track and album table. And then use another JOIN statement to join the artist table with the previous joined table by using ON lv2\_albums.artist\_id = lv2\_artists.id to join the two tables.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT lv2_tracks.name, lv2_albums.name, lv2_artists.name FROM lv2_tracks JOIN lv2_albums ON lv2_tracks.album_id =
lv2_albums.id JOIN lv2_artists ON lv2_albums.artist_id = lv2_artists.id
```

Passed

Expected result

name	name	name
青春コンプレックス	結束バンド	結束バンド
ギターと孤独と蒼い惑星	結束バンド	結束バンド
星座になれば	結束バンド	結束バンド
あのバンド	結束バンド	結束バンド
忘れてやらない	結束バンド	結束バンド
カラカラ	結束バンド	結束バンド

Actual result

name	name	name
青春コンプレックス	結束バンド	結束バンド
ギターと孤独と蒼い惑星	結束バンド	結束バンド
星座になれば	結束バンド	結束バンド
あのバンド	結束バンド	結束バンド
忘れてやらない	結束バンド	結束バンド
カラカラ	結束バンド	結束バンド



Query track and album information using base table 'lv2\_albums'

Use the same JOIN statement as the previous task to query, but this time use the base table 'lv2\_albums' to query the track and album information.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

SET STATEMENT SQL\_SELECT\_LIMIT=501 FOR SELECT \* FROM lv2\_albums JOIN lv2\_tracks ON lv2\_tracks.album\_id = lv2\_albums.id

Passed

Expected result

id	spotify_id	name	artist_id	artwork_url
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	https://i.scdn.co/image/at
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	https://i.scdn.co/image/at
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	https://i.scdn.co/image/at
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	https://i.scdn.co/image/at
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	https://i.scdn.co/image/at
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	https://i.scdn.co/image/at

Actual result

id	spotify_id	name	artist_id	artw
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	http
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	http
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	http
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	http
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	http
1	5ZGzGGNAB6U7QIKpdaMu0d	結束バンド	1	http

Concat track name and album name into one column

Use MySQL CONCAT function to concat track name and album name into one column.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

SET STATEMENT SQL\_SELECT\_LIMIT=501 FOR SELECT CONCAT(lv2\_tracks.name, ' - ', lv2\_albums.name) FROM lv2\_tracks JOIN lv2\_albums ON lv2\_tracks.album\_id = lv2\_albums.id

Passed

Expected result

CONCAT(lv2_tracks.name, ' - ', lv2_albums.name)
青春コンプレックス - 結束バンド
ギターと孤独と薔薇 - 結束バンド
星座になれたら - 結束バンド
あのバンド - 結束バンド
忘れてやらない - 結束バンド
カラカラ - 結束バンド

Actual result

CONCAT(lv2_tracks.name, ' - ', lv2_albums.name)
青春コンプレックス - 結束バンド
ギターと孤独と薔薇 - 結束バンド
星座になれたら - 結束バンド
あのバンド - 結束バンド
忘れてやらない - 結束バンド
カラカラ - 結束バンド

Show all concatenated track name in one column and album name in another column

Use MySQL CONCAT function to concat track name of each album into one column. And then use AS statement to rename the column name to 'tracks'.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT CONCAT(lv2_tracks.name) AS tracks FROM lv2_albums JOIN lv2_tracks ON lv2_tracks.album_id = lv2_albums.id
```

Passed

Expected result

tracks
青春コンプレックス
ギターと孤独と蒼い惑星
星座になれたら
あのバンド
忘れてやらない
カラカラ

Actual result

tracks
青春コンプレックス
ギターと孤独と蒼い惑星
星座になれたら
あのバンド
忘れてやらない
カラカラ

Show sum of track duration of each album

Use MySQL SUM function to sum the duration of each album.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT SUM(lv2_tracks.duration) FROM lv2_albums JOIN lv2_tracks ON lv2_tracks.album_id = lv2_albums.id
```

Passed

Expected result

SUM(lv2_tracks.duration)
21358566

Actual result

SUM(lv2_tracks.duration)
21358566

Show sum of track duration of each artist and order by the sum of duration

Use MySQL SUM function to sum the duration of each artist joined from artist to album to track. And then use ORDER BY statement to order the result by the sum of duration.

DIFFICULTY **EASY**  
LEVEL **LV2**

SHOW HINT

```
SET STATEMENT SQL_SELECT_LIMIT=501 FOR SELECT SUM(lv2_tracks.duration) FROM lv2_artists JOIN lv2_albums ON lv2_albums.artist_id = lv2_artists.id JOIN lv2_tracks ON lv2_tracks.album_id = lv2_albums.id ORDER BY SUM(lv2_tracks.duration)
```

Passed

Expected result

SUM(lv2_tracks.duration)
21358566

Actual result

SUM(lv2_tracks.duration)
21358566