Простые (2):

1) Вывести названия альбомов подкастов без пометки e ("isExplicit" = false) в отсортированном по алфавиту порядке.

SELECT title FROM "PodcastAlbum"

WHERE "isExplicit" = false

ORDER BY title

2) Изменить название 13-го жанра на “Rock” и вывести изменившуюся строчку.

UPDATE "Genre"

SET name = 'Rock'

WHERE genre\_id = 13

RETURNING \*

Средние (6):

1) Вывести пользователей, их прослушанные треки и количество прослушиваний в порядке возрастания количества для 10 пользователей с наибольшим числом прослушиваний. Учитывать треки длительностью до часа.

SELECT \* FROM (

SELECT name, title, count

FROM "User"

INNER JOIN "Track-User-Listen" ON "Track-User-Listen".user\_id = "User".user\_id

INNER JOIN "Track" ON "Track".track\_id = "Track-User-Listen".track\_id

WHERE "Track".duration < '01:00:00'

ORDER BY count DESC

LIMIT 10

) new\_table

ORDER BY count

2) Вывести первые 10 исполнителей и число их треков, осортированных в порядке убывания количества треков.

SELECT name, COUNT(track\_id) num\_tracks FROM "Track"

INNER JOIN "Track-Artist" USING (track\_id)

INNER JOIN "Artist" USING (artist\_id)

GROUP BY artist\_id, name

ORDER BY num\_tracks DESC

LIMIT 10

3) Вывести 10 артистов, выступающих в жанре ‘Rock’ и осортированных по имени, и информацию о них.

SELECT "Artist".name artist FROM "Genre-Artist"

INNER JOIN "Genre" USING (genre\_id)

INNER JOIN "Artist" USING (artist\_id)

WHERE "Genre".name = 'Rock'

ORDER BY "Artist"."name"

LIMIT 10

4) Вывести 10 записей c именем артиста и количеством его ссылок, переименованных в num\_links, упорядоченных по числу ссылок по убыванию.

SELECT "Artist".name artist, COUNT(link\_id) num\_links FROM "Link"

INNER JOIN "Artist" USING (Artist\_id)

GROUP BY "Artist".name

ORDER BY num\_links DESC

LIMIT 10

5) Найти топ-3 пользователей по количеству плейлистов подкастов. Вывести их имена, а также количество плейлистов.

SELECT name, COUNT("PodcastPlaylist".podcastplaylist\_id) num\_podcasts FROM "User-PodcastPlaylist"

INNER JOIN "User" USING (user\_id)

INNER JOIN "PodcastPlaylist" USING (podcastplaylist\_id)

GROUP BY name

ORDER BY num\_podcasts DESC

LIMIT 3

6) Вывести все альбомы подкастов, сами подкасты, их длительность, где нет марировки e, а длительность составляет от часу до 3 часов. Результаты отсортировать по названию альбома и по названию подкаста.

SELECT "PodcastAlbum".title Album, "Podcast".title Podcast, duration FROM "PodcastAlbum-Podcast"

INNER JOIN "PodcastAlbum" USING (podcastalbum\_id)

INNER JOIN "Podcast" USING (podcast\_id)

WHERE "PodcastAlbum"."isExplicit" = False AND duration BETWEEN '01:00:00' AND '03:00:00'

ORDER BY "PodcastAlbum".title, "Podcast".title

Сложные (7):

1) Найти топ-10 артистов по количеству лайков от премиум-пользователей. Вывести их ФИО, а также число лайков, отсортированные сначала по числу лайков по убыванию, потом по имени по алфавиту.

SELECT "Artist".name artist, COUNT("User"."name") num\_users FROM "Artist-User"

INNER JOIN "Artist" ON "Artist".artist\_id = "Artist-User".artist\_id

INNER JOIN

(SELECT \* FROM "User"

WHERE "User"."hasPremium" = True) "User"

ON "User".user\_id = "Artist-User".user\_id

GROUP BY "Artist".artist\_id

ORDER BY num\_users DESC, artist

LIMIT 10

2) Вывести ФИО пользователей, кто больше всего суммарно прослушал треков (таких может быть несколько). Учесть, что названия треков и ФИО пользователей могут совпадать. Отсортировать по алфавиту.

SELECT name FROM "User"

WHERE user\_id IN (SELECT user\_id FROM "Track-User-Listen"

INNER JOIN "Track" USING (track\_id)

INNER JOIN "User" USING (user\_id)

GROUP BY user\_id

HAVING SUM(count) = (SELECT SUM(count) FROM "Track-User-Listen"

INNER JOIN "Track" USING (track\_id)

INNER JOIN "User" USING (user\_id)

GROUP BY user\_id

ORDER BY sum DESC

LIMIT 1))

ORDER BY name

3) Найти треки, которые заблокировало наибольшее число пользователей. Учесть, что названия треков, ФИО пользователей могут сопадать. Отсортировать по алфавиту.

SELECT title FROM "Track"

WHERE track\_id IN (SELECT track\_id FROM "Track-User-Block"

INNER JOIN "Track" USING (track\_id)

INNER JOIN "User" USING (user\_id)

GROUP BY track\_id

HAVING COUNT(user\_id) = (SELECT COUNT(user\_id) FROM "Track-User-Block"

INNER JOIN "Track" USING (track\_id)

INNER JOIN "User" USING (user\_id)

GROUP BY track\_id

ORDER BY count DESC

LIMIT 1))

ORDER BY title

4) Найти топ-3 самых популярных жанра у треков с пометкой e (isExplicit). Суммарная продолжительность треков одного жанра должна быть более 30 часов. Вывести название жанров, количество их треков.

SELECT name, COUNT(track\_id) FROM "Genre-Track"

INNER JOIN (SELECT \* FROM "Track" WHERE "Track"."isExplicit" = true) "Track" USING (track\_id)

INNER JOIN "Genre" USING (genre\_id)

GROUP BY name

HAVING SUM(duration) > '30:00:00'

ORDER BY count DESC

LIMIT 3

5) Найти подкасты, которые прослушало наименьшее число премиум пользователей. Учитывать только подкасты, длящиеся менее часа. Также учесть, что названия подкастов и ФИО пользователей могут повторяться. Вывести единственную строчку – их количество.

SELECT COUNT(title) FROM "Podcast"

WHERE podcast\_id IN (SELECT podcast\_id FROM "User-Podcast-Listen"

INNER JOIN (SELECT \* FROM "User" WHERE "User"."hasPremium" = TRUE) "User" USING (user\_id)

INNER JOIN (SELECT \* FROM "Podcast" WHERE "Podcast"."duration" < '01:00:00') "Podcast" USING (podcast\_id)

GROUP BY podcast\_id

HAVING COUNT(user\_id) = (SELECT COUNT(user\_id) FROM "User-Podcast-Listen"

INNER JOIN (SELECT \* FROM "User" WHERE "User"."hasPremium" = TRUE) "User" USING (user\_id)

INNER JOIN (SELECT \* FROM "Podcast" WHERE "Podcast"."duration" < '01:00:00') "Podcast" USING (podcast\_id)

GROUP BY podcast\_id

ORDER BY count

LIMIT 1))

6) Вывести любимый жанр у пользователя с id = 3, который суммарно прослушал треки этого жанра менее чем на 10 часов. Также вывести количество его треков в этом жанре.

SELECT "Genre".name, COUNT(track\_id) FROM "Track-User-Like"

INNER JOIN "Track" USING (track\_id)

INNER JOIN (SELECT \* FROM "User" WHERE user\_id = 3) "User" USING (user\_id)

INNER JOIN "Genre-Track" USING (track\_id)

INNER JOIN "Genre" USING (genre\_id)

GROUP BY user\_id, "Genre".name

HAVING SUM("Track".duration) < '10:00:00'

ORDER BY count DESC

LIMIT 1

7) Вывести не премиум пользователя, который лайкнул более 3 подкастов на наибольшее суммарное время. Учесть, что названия подкастов могут совпадать. Вывести пользователя и это время.

SELECT name, sum FROM "User"

RIGHT JOIN (SELECT user\_id, SUM(duration) FROM "User-Podcast-Like"

INNER JOIN (SELECT \* FROM "User" WHERE "User"."hasPremium" = FALSE) "User" USING (user\_id)

INNER JOIN "Podcast" USING (podcast\_id)

GROUP BY user\_id

HAVING COUNT(podcast\_id) > 3

ORDER BY sum DESC

LIMIT 1) t1 USING (user\_id)