|  |  |  |
| --- | --- | --- |
| Лабораторная работа № 4  Реализация SQL-запросов базы данных | Ф.И.О. | Белоусов Дмитрий Вадимович |
| Группа | ИВТ-260 |
| Преподаватель | Соколов Александр Александрович |
| Дата сдачи | 11.05.2022 |

# Задание на лабораторную работу

Необходимо написать для своей базы данных **как минимум 100 SQL-запросов.** Обязательно в них должны входить указанные ниже запросы (число в скобках носит рекомендательный характер). Приветствуется добавление собственных запросов с теми операторами, которые ниже не указаны, но о которых рассказывалось на лекциях.

1. Запросы, которые вы указали в **функциональных** требованиях (в заголовке указать, что за требование) (**10 шт. +**)
2. **UPDATE** в **разных** таблицах, с **WHERE**, можно условно, например, изменить заранее созданные некорректные данные (**5 шт.**)
3. **DELETE** в **разных** таблицах, с **WHERE**, можно условно, например, удалить заранее созданные некорректные данные (**5 шт.**)
4. **SELECT, DISTINCT, WHERE, AND/OR/NOT, IN, BETWEEN**, различная работа с **датами** и **числами**, преобразование данных, **IS NULL**, **AS** для таблиц и столбцов и др. в различных вариациях (**15 шт. +**)
5. **LIKE** и другая работа со строками (**5-7 шт.+**)
6. **SELECT INTO** или **INSERT SELECT**, что поддерживается СУБД (**2-3 шт.**). Для использования запроса **INSERT SELECT** вначале можно создать новую тестовую таблицу или несколько, в которые будут скопированы данные из существующих таблиц с помощью данного запроса. Код создания таблиц также приложить в лабораторную работу
7. **JOIN: INNER, OUTER (LEFT, RIGHT, FULL), CROSS, NATURAL**, разных, в различных вариациях, несколько запросов с более, чем одним **JOIN** (**15 шт.+**)
8. **GROUP BY** (некоторые с **HAVING**), **LIMIT, ORDER BY** (**ASC**|**DESC**) вместе с **COUNT**, **MAX**, **MIN, SUM, AVG** в различных вариациях, можно по отдельности (**15 шт.+**)
9. **UNION**, **EXCEPT**, **INTERSECT**, что поддерживается СУБД (**3-5 шт.**)
10. Вложенные SELECT с GROUP BY, ALL, ANY, EXISTS (3-5 шт.)
11. **GROUP\_CONCAT** и другие разнообразные функции SQL (**2-3 шт.**)
12. Запросы с **WITH** (**2-3 шт.**)
13. Запросы со строковыми функциями СУБД, с функциями работы с датами временем (форматированием дат), с арифметическими функциями (**5-7 шт.**)
14. Сложные запросы, входящие в большинство групп выше, т.е. **SELECT** ... **JOIN** ... **JOIN** ... **WHERE** ... **GROUP BY** ... **ORDER BY** ... **LIMIT** ...; (**5-7 шт. +**), можно написать больше вместо простых.

# Протокол и отчёт лабораторной работы

В протокол необходимо вставить:

1. **скриншот** реляционной схемы вашей БД (из 3-й лабораторной)
2. текст всех запросов, **отформатированный** для читаемости (с отступами), а не "простынёй текста". Запросы должны быть **сгруппированы** по видам (см. ниже) и **пронумерованы**, к каждому запросу нужно короткое **описание** (заголовок), **что делается в этом запросе** (что ожидаем получить с его помощью)
3. к сложным запросам (с **JOIN, GROUP BY,** большим блоком **WHERE** и т.п.) необходимо приложить **скриншоты** из СУБД с результатами выполнения

# Выполнение и отчёт

## Функциональные требования:

**Справочные (оперативные запросы):**

1. Показать «любимые» треки, релизы, исполнителей пользователя.
2. Найти трек по части названия
3. Просмотреть дату выпуска релиза
4. Просмотреть длительность трека
5. Показать длительность релиза

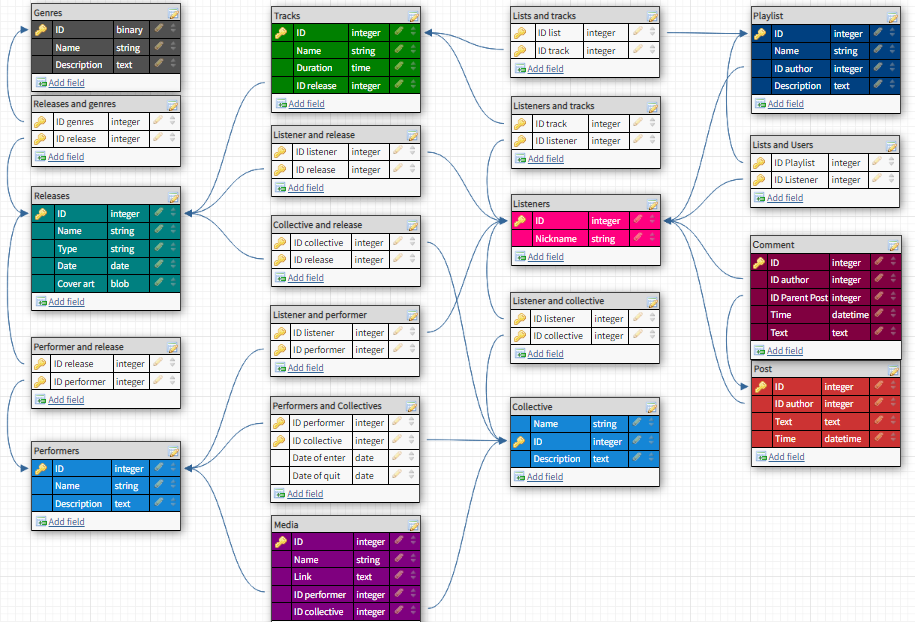
**Аналитические запросы:**

1. Показать топ-3 наиболее популярных (добавленных) треков на релизе
2. Показать релизы, выпущенные за некоторый заданный период времени
3. Определить исполнителя и релиз по треку
4. Показать количество слушателей исполнителя/релиза/трека
5. Показать топ-3 самых популярных треков жанра

## Модификация:

1. Вывести все посты и их комментарии за заданный промежуток времени и приджойнить их авторов
2. Топ 3 жанра через релизы по количеству лайков
3. Для всех коллективов вывести всех участников, через group concat, которые на текущий момент состоят в группе

## Реляционная схема:



## Код:

[Ссылка на проект в Github, где можно найти файл с sql-скриптами](https://github.com/coldousedbird/DataBasesProject.git)