«Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №42 «Криптология и кибербезопасность»

Отчёт по лабораторной работе №3

"Безопасность базы данных"

Студент: Нурматов С.Р.

Группа: Б17-565

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2020

**Сложные запросы на выборку. Соединения**

Запросы составлены на основе база данных, использованной в лабораторной работе №1.

В компании BestTVS, продуктом которой является сайт для просмотра сериалов, появился новый отдел, который занимается написанием статей про сериалы и все что с ними связано. Для написания статей сотрудникам компании может понадобиться различная информация. Составлен список вопросов, ответы на которые можно использовать в той или иной статье:

1. **Подзапросы**
   1. Получить список сериалов имеющие больше одного сезона:

SELECT id, serial\_name FROM serials\_list WHERE id IN (SELECT serial\_id FROM series WHERE season\_number > 1);

1. **Соединения**
   1. Получиться количество сериалов, которые были сняты в каждой стране:

SELECT c.country\_name, COUNT (sl.serial\_name) AS serials\_quantity FROM serials\_list sl NATURAL JOIN countries c GROUP BY c.country\_name;

* 1. Получить список каналов, на которых транслируются сериалы:

SELECT ch.channel\_name, sl.serial\_name

FROM serials\_channels sl\_ch

JOIN channels ch ON sl\_ch.channel\_id = ch.channel\_id

JOIN serials\_list sl ON sl\_ch.serial\_id= sl.id;

* 1. Получить список актеров сериала «Кухня»:

SELECT sl.serial\_name, pr.person\_name

FROM serials\_persons sl\_pr

JOIN serials\_list sl ON sl\_pr.serial\_id = sl.id

JOIN persons pr ON sl\_pr.person\_id = pr.person\_id

WHERE (sl.id = 1 AND pr.It\_is\_producer = 0);

Доп задание

SELECT LISTAGG(p.person\_name, ' , ') WITHIN GROUP (ORDER BY COUNT(s\_p.person\_id) DESC) as popularity

FROM serials\_persons s\_p

JOIN persons p ON p.person\_id = s\_p.person\_id

GROUP BY p.person\_name;

1. **Иерархические запросы:**
   1. Этот запрос выводит уровень актеров и продюсеров в иерархии и соответствующего менеджера:

SELECT level, person\_id, person\_name, Prior person\_name manager\_name

FROM persons

CONNECT BY PRIOR person\_id=manager\_id

START WITH manager\_id IS null;

* 1. Создать путь назначения от корневого узла до текущего узла: SELECT person\_name, Sys\_Connect\_By\_Path (person\_id,':') Path

FROM persons

CONNECT BY PRIOR person\_id=manager\_id

START WITH manager\_id IS null;

Дополнительное задание

SELECT sl.serial\_name, LISTAGG(pr.person\_name, ' , ')

WITHIN GROUP (order by pr.popularity

DESC) AS popularity

FROM serials\_persons sl\_pr

JOIN serials\_list sl ON sl\_pr.serial\_id = sl.id

JOIN persons pr ON sl\_pr.person\_id = pr.person\_id

GROUP BY sl.serial\_name;

1. **Аналитические функции**
   1. Получить ранг series, отдельно для каждого сериала, по мере увеличения количества зрителей:

SELECT series\_id, serial\_id, series\_number, viewers,

RANK () OVER (PARTITION by serial\_id ORDER BY viewers) as rank

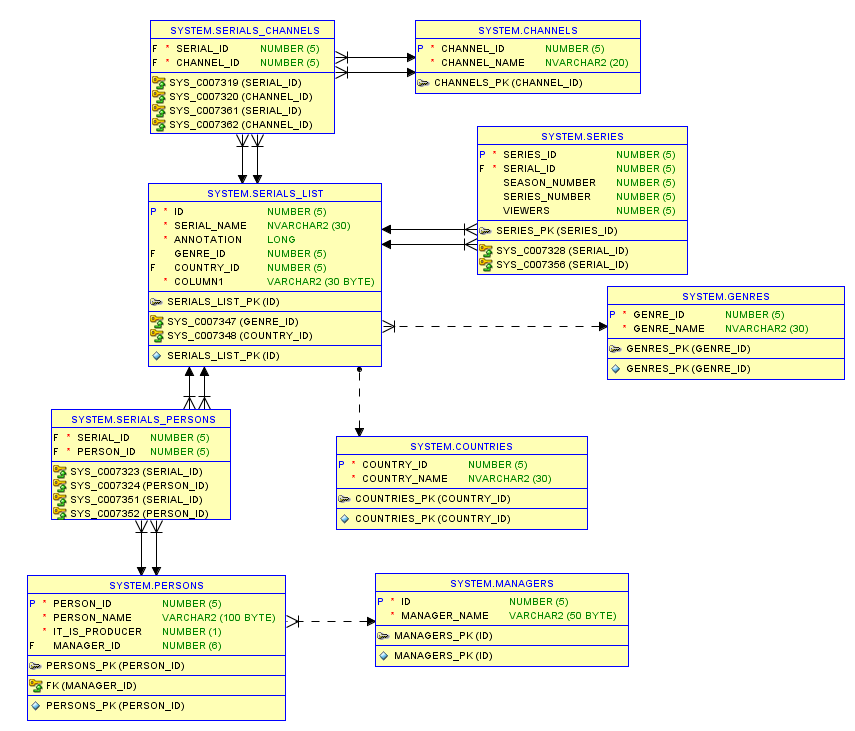
FROM Series;

* 1. Этот запрос позволяет рассчитать общее количество зрителей для каждого сериала

SELECT series\_id, serial\_id, series\_number, viewers,

SUM (Viewers) OVER (PARTITION by serial\_id ORDER BY series\_id ROWS

BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) as sum FROM Series;



**Заключение**

В ходе лабораторной работы были изучены и протестированы сложные запросы, в наибольшей степени обеспечивающие удобство использования баз данных.

Ссылка на приложение:

<https://github.com/Nurmatov-07/DataBase2020/tree/b430ab94ab84f8b89cf0df82503187afb2e137f8/Lab1_3>