## МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

# Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский ядерный университет «МИФИ»»

Кафедра №42 «Криптология и кибербезопасность»

ЛАБОРАТОРНАЯ РАБОТА №1-2:

«Инициализация данных. Простые запросы на выборку.»

Аверин Владислав

Группа Б19-505

# <u>Содержание</u>

| Изменения в структуре БД         | 3   |  |
|----------------------------------|-----|--|
| Создание информации и заполнение | 4   |  |
| Формирование простых запросов    | . 5 |  |
| Выводы                           | 8   |  |

# Изменения в структуре БД

Для начала необходимо рассказать об изменениях, которые я внес в мои таблицы по ходу работы над 2 лабораторной работой. Все изменения из в файле *changes.txt*, однако описать их все равно необходимо. Начнем с минорных изменений:

- 1) Изменение размеров данных VARCHAR2. Проектируя БД без ее практического заполнения, я забыл о том, что размер, который указывается в VARCHAR2, это размер в байтах, поэтому кол-во потенциально вмещаемых символов в строку != ее длине в байтах (кириллица в UTF8 не вмещается, ей нужна двухбайтовая кодировка, поэтому русские символы занимают 2 байта). В следствие чего я много где расширил строковые данные для того, чтобы в них помещалась хотя бы минимальная информация (где конкретно, см. changes.txt):
- 2) Добавлены в некоторые поля заполнение по умолчанию, чтобы при ручной вставке не прописывать их, если они почти не отличаются (например, уровень доступа у дела на 11);
- 3) Во время формирования датасета для OfficeHours у меня возникла проблема. Мне в атрибутах таблицы нужны только моменты времени, без даты. Однако я не нашел способа задать только время без дня, года и месяца. Поэтому в OfficeHours.csv у меня костыль в виде даты 01.01.2000, с которой уже идет время для прямого назначения. Неужели нет какого способа хранить время без привязки к дате? Единственное, что пришло в голову, это переводить все время дату в формат строки TO\_CHAR'ом, отбрасывая день, и хранить время в виде строки, но ведь потом для работы, например, с промежутками времени, нужно будет обратно конвертировать строку в дату, где TO\_DATE автоматом подставит «недостающую» часть формата (как я понял, это дефолтно первый день текущего месяца);

А теперь самая главная «фича»: уровни допуска.

Пытаясь продумать и сгенерировать датасет, я внезапно понял, насколько моя система уровней допуска была неудобной. И все из-за одной вещи. Ведь раньше он показывал, может ли тот или иной сотрудник работать над каким-то делом. Причем, допуск был прикручен к должности. Но это ж бред :) (Да-да, сам сделал, сам и сломал). Если, к примеру, патологоанатом имеет какой-то уровень допуска, то автоматом остальные патологоанатомы имеют такой же. Значит, нужно будет вводить патологоанатома І уровня, ІІ и т.д.. Ну и толку тогда от уровней в принципе, если они коррелируют с названиями должностей? Поэтому теперь уровень допуска индивидуален, и дает информацию сразу о двух вещах. Возьмем того же патологоанатома. Теперь он сможет провести вскрытие для какого-то дела (или его могут попросить сделать это). Он будет с ним связан (даже пусть не напрямую), но без ДОСТУПА непосредственно к материалам. Такая схема выглядит, как по мне, более логично. То есть какой-нибудь ОМОН может участвовать в расследовании (как грубая сила), но его сотрудники не должны допускаться к самому делу. Это привело меня к системе сдвоенного уровня допуска (о котором мне, вообще, уже говорили, я просто не слушал). А именно:

Всего уровней допуска все так же 4:

- I: Обычный доступ к делам, не затрагивающим гос. тайну.
- II: Секретно: доступ к делам, имеющих сведения, подверженные гос. тайне.
- III: Совершенно секретно: разглашение гос.тайны может нанести ущерб интересам МВД или отрасли экономики.

IV: Особой важности: разглашение гос.тайны может нанести ущерб интересам государства (он же полный доступ).

Однако сам допуск теперь состоит из двух цифр, где каждая цифра принадлежит [0, 4]. Первая цифра отвечает за то, к каким делам сотрудник имеет доступ (Т.е. может получить информацию о нем). Вторая же показывает, к делам какого уровня он может привлекаться как эксперт. Это значит, что теперь в БД будут храниться записи обо всех связях сотрудников и дела (А вот какую именно роль он исполнял в деле, можно посмотреть в самих материалах дела). Соотвественно, уровни допуска могут быть: 01 (1), 02 (2), 03 (3), 04 (4), 11, 12, 13, 14, 22, 23, 24, 33, 34, 44. Такая система еще и удобна тем, что уровни допуска строго упорядочены, а значит, можно сразу проверять уровень допуска сотрудника по этой иерархии. Более подробное описание каждого из уровней можно найти в AccessLevels.csv.

Р.S. Немного уточненный механизм работы уровней допусков: эти допуски есть как у сотрудников, так и у всех расследований. Но запись сотрудника уровня 04 (напр. оперативник ОМОНа), работающего / работавшего над делом уровня 44 - нормальное явление. Это значит, что он был просто связан с ним (участвовал в облаве по поимке подозреваемых, например), но он НЕ МОЖЕТ запросить доступ на это дело и получить удовлетворительный ответ. То есть теперь уровни допусков разграничивают то, какие дела могут просматривать те или иные сотрудники.

Вот, вроде бы, и все изменения в ERD. Новую версию диаграммы я залил на git, а старая схема осталась там же (чтобы можно было чекнуть изменения).

## Создание информации и заполнение

Я использовал для импорта в БД SQL Developer (надеюсь, это не возбранялось), поэтому тут особо нечего комментировать: в каждом .csv файле есть header title (или как оно там называется), которое в настройках импорта надо поставить игнорироваться, а так же раздельный символ - «;». Все файлы я успешно импортировал (единственное, мне не понравилось, если нужно подправить один атрибут в процессе работы, менять его ручками в Developer, а потом его же в .csv файле ради синхронизации, поэтому в какой-то момент мне хотелось чисто заполнить БД изнутри, а потом экспортировать готовый файл). Собственно говоря, надеюсь, меня не покарают за то, что я использовал такой простой способ в виде Developer'а и не юзал SQL Loader.

## Формирование простых запросов

(Все данные запросы лежат в скрипте requests.sql)

Для начала попробуем добавить новую запись, например, в таблицу Employees:

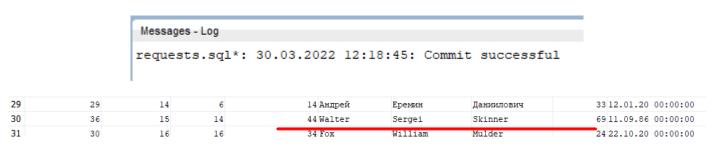
#### INSERT INTO employees

(employee\_id, department\_id, post\_id, access\_level\_id, first\_name, second\_name, patronymic, age, employment\_date)

#### **VALUES**

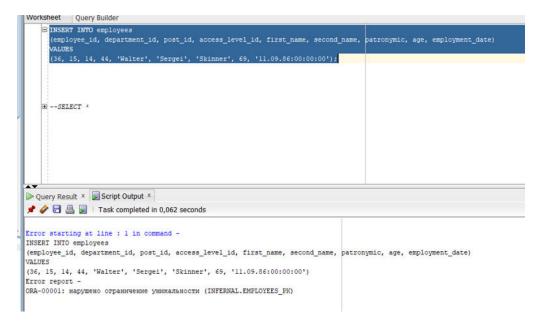
(36, 15, 14, 44, 'Walter', 'Sergei', 'Skinner', 69, '11.09.86:00:00:00');

Как видим, оно успешно добавилось:



(Правда, вручную оно добавляется как-то криво и непонятно куда: она отображается не в конце, как должно по идее быть, в где-то между employee id 29 и 30)

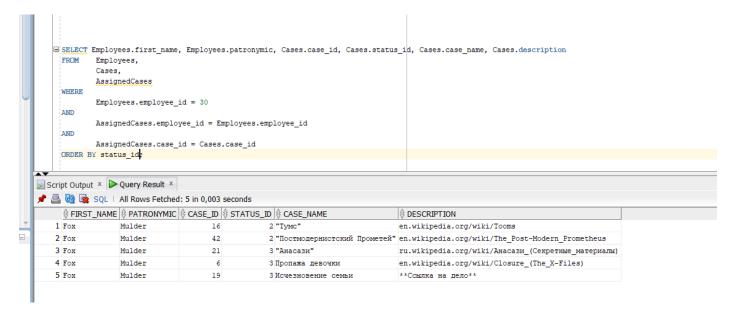
Ну и естественно, если мы попытаемся добавить абсолютно такое же поле еще раз, БД откажется это делать, ругаясь, что такой ключ уже использается.



#### SELECT + WHERE + ORDER:

Пример с полной выборкой SELECT, условной выборкой WHERE и упорядоченной выборкой ORDER BY я объединил в один (так как по отдельности они в принципе понятные).

«Есть сотрудник, нам известен его номер (employee\_id). Требуется узнать, в каких делах он принимал участие.»



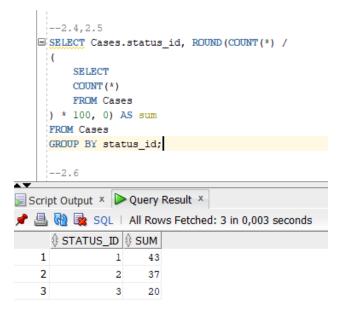
Здесь мы объединили 2 таблицы Employees и Cases в одну через таблицу AssignedCases.

P.S. Ура, мы изобрели JOIN :)

## Использование простых и агрегатных функций:

Не будем далеко ходить, например:

«Посчитаем статистику дел: сколько из них в процессе, сколько раскрыто, сколько заморожено (не раскрыто, но закрыто).»

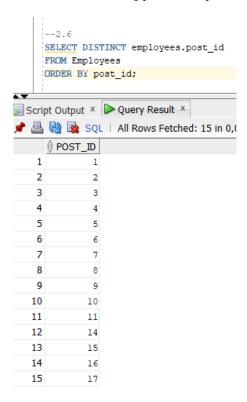


Для этого ищем подзапросом кол-во строк в Cases, и отдельно кол-во строк со значением каждого из статусов. Округлим для того, чтобы получить целое значение (типа, красивее)

(20% не раскрытых дел, гм... Че-то много, как-то плохо я сгенерил для идеального полицейского участка)

## Выборка уникальных записей:

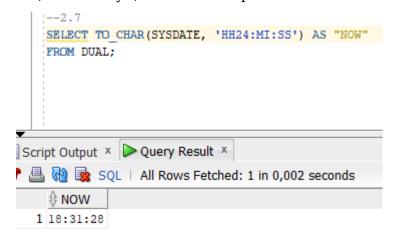
Тут немного сложнее, ибо не используя JOIN'ов (см. след. лабу) в моей схеме придумать что-то уникальное, что можно было бы логично посчитать, оказалось немного сложно. Ну давайте, например, посчитаем, сотрудники каких отделов трудятся в участке на данный момент:



Как мы видим, в участке нет ни одного сотрудника с должностями 12 и 13 (Вообще, можно было бы и соединить таблицы, чтобы они выдавали названия отделов и чтобы запрос показывал сразу отсутствующие должности, но для примера, думаю, этого достаточно)

## Выборка из вырожденной таблицы:

Тут я тоже далеко не лез, вывел текущее системное время:



# Выводы

В результате работы были сконфигурированы данные для импорта в БД, изменена ее структура на более логичную, а так же проведены самые примитивные запросы для проверки функциональности и валидности наших данных.