

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ  
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА  
ЭКОНОМИКИ»

Высшая школа бизнеса  
Образовательная программа  
«Бизнес-Информатика»

Контрольное домашнее задание по дисциплине  
Базы данных

Выполнили:  
Газизова Анита, ББИ2006  
Овечкина Мария, ББИ2006  
Овчарук Полина, ББИ2006  
Тюльпакова Марина, ББИ2006  
Щербакова Виктория, ББИ2006

Москва, 2022

## Оглавление

1. Роли в команде	4
2.Ссылка на видео	5
3.Анализ предметной области	6
3.1.Общие сведения о предметной области	6
3.2.Цель создания базы данных	7
3.3.Возможные пользователи и сценарии взаимодействия пользователей и БД	7
3.4.Список основных сущностей вашей предметной области	8
3.5.Атрибуты каждой сущности с указанием первичных ключей	8
3.6. Описание связей между сущностями	9
3.7.Алгоритмы обработки данных, используемые в сценариях	10
3.8.Планируемый перечень отчетных форм	11
3.9.Возможная архитектура программного продукта на основе вашей базы данных	11
4.1.Инфологическая модель предметной области	12
4.2.Даталогическая модель	12
4.3.Диаграмма с типами данных	14
4.4.Нормализация	14
5.Заполнение таблиц данными	15
6.Разработка СУБД	23
6.1.Запросы	23
6.2.Триггер	34
6.3.Функции	34
6.6.Представления	37
6.6.Хранимые процедуры	39
6.7.Настройка индексов	45

7.Отчетные формы	47
7.1.Основной отчёт Tableau (Power BI) – агрегированные данные со ссылками на детализацию	47
7.2.Детальный отчёт Tableau (Power BI) для drill down	47
7.3.Информационная панель в Excel с графиком и "срезами"	49

## 1. Роли в команде

Анита:

- описание предметной области (описание сущностей, атрибутов и связей)
- разработка даталогической и инфологической моделей
- создание запросов
  - Простой запрос с условием и формулами в SELECT(2 запроса)
- создание триггера
- разработка отчета в Excel
- создание процедур: Процедура №3

Мария:

- описание предметной области
- создание запросов:
  - Запрос с EXISTS
  - Запрос с SELECT INTO
  - Запросы с подзапросом во FROM(2 запроса)
  - Запрос с Запрос, использующий оконную функцию LAG или LEAD для выполнения сравнения данных в разных периодах
  - Запрос, использующий манипуляции с множествами
- создание таблиц в СУБД
- настройка индексов
- подготовка видео

Марина:

- заполнение таблиц данными с помощью парсинга сайта и Python
- создание запросов
  - Запрос с подзапросом в FROM, агрегированием, группировкой и сортировкой(2 запроса)
  - Запрос с внешним соединением и проверкой на наличие NULL
  - Запрос с CASE (IIF) и агрегированием
  - Запрос с HAVING и агрегированием
- создания одной функции(№1)
- создание представлений (оба)
- создание процедур: Процедура №4

Виктория:

- заполнение таблиц данными с помощью Python
- создание запросов:

- Запрос с коррелированным подзапросом в SELECT(2 запроса)
  - Запрос с коррелированным подзапросом в WHERE(2 запроса)
  - Запрос с агрегированием и выражением JOIN, включающим не менее 2 таблиц(3 запроса)
  - Запрос с агрегированием и выражением JOIN, включающим не менее 3 таблиц/выражений
- создания одной функции(№2)
  - создания процедур: Процедуры №1, №2
  - Отчет в Power BI

## 2.Ссылка на видео

[https://www.youtube.com/watch?v=qwNqLnoe8\\_g](https://www.youtube.com/watch?v=qwNqLnoe8_g)

### 3. Анализ предметной области

#### 3.1. Общие сведения о предметной области

Государственный музей изобразительных искусств имени А.С. Пушкина - музейный комплекс, обладающий одним из крупнейших в России художественных собраний зарубежного искусства, хранящий артефакты, созданные мастерами разных эпох - от Древнего Египта и античной Греции до наших дней. Только одна постоянная экспозиция насчитывает более двух тысяч произведений искусства художников со всего мира.

В наше время ГМИИ им. А.С. Пушкина нисколько не теряет свою популярность благодаря обширному выбору экскурсионных программ, специально подготовленных как для самых маленьких слушателей, так и для взрослых посетителей, а также для преподавателей и искусствоведов. В ходе экскурсии посетители в небольших группах могут выбрать одно или несколько зданий музея для обхода. В настоящий момент можно посетить 7 зданий музея с различной направленностью. В нашем проекте мы остановимся на создании базы данных для постоянной экспозиции одного из действующих зданий, а именно для Галереи искусства стран Европы и Америки XIX–XX веков. По проходимости данная галерея не уступает главному зданию музея, что обусловлено их территориальной близостью.

Основу постоянной экспозиции Галереи искусства стран Европы и Америки XIX–XX веков составляют шедевры французского импрессионизма, постимпрессионизма и авангардных течений начала XX столетия из коллекций московских собирателей Сергея Щукина и Ивана Морозова. Это всемирно известные произведения Клода Моне, Пьера Огюста Ренуара, Эдгара Дега, Поля Сезанна, Винсента ван Гога, Поля Гогена, Анри Матисса, Пабло Пикассо. Здесь также представлены живопись и скульптура романтизма, реализма, академизма, искусство символизма, абстракционизма и других художественных направлений XX века.

### 3.2.Цель создания базы данных

Создание базы данных не только с информацией об экспонатах выставки, а также с информацией о билетах и посетителях выставки может способствовать деятельности музеев в анализе целевой аудитории посетителей, а также оптимизировать работу заведения, например при подсчете выручки от галереи за определенный период.

Более того, хранение данных в базе гарантирует ускорение процесса обработки данных, в отличие от хранения, например, в Excel, и предоставляет новые возможности в области использования аналитических инструментов для их обработки.

### 3.3.Возможные пользователи и сценарии взаимодействия пользователей и БД

Предположительно базой данных могут пользоваться следующие сотрудники ГМИИ им. А.С. Пушкина:

- Аналитики
- Маркетологи
- Бухгалтеры
- Работники архива
- Смотрители залов
- Экскурсоводы
- Продавцы билетов
- Разработчики приложения/сайта музея

С помощью базы данных можно выполнять следующие функции:

- вывод данных о выручке в целом или за определенный временной промежуток для планирования бюджета;
- вывод данных о посещении как в целом, так и отдельных категорий гостей для проведения анализа целевой аудитории;
- сбор и хранение электронных почт посетителей для осуществления новостной рассылки музея;
- планирование экскурсий: с помощью базы данных можно создавать новые экскурсии или дополнять/упрощать уже созданные;
- информирование об экспонатах выставки

### 3.4.Список основных сущностей вашей предметной области

1. Автор: данная сущность хранит информацию об авторах произведений, выставляющихся в галерее.
2. Произведение: данная сущность хранит информацию о различных произведениях, которые выставляются в галерее. Хранится такая информация, как музейный номер, название самого произведения, год и страна его создания.
3. Коллекция: сущность хранит справочную информацию о коллекциях музея.
4. Материал: сущность хранит справочную информацию о материалах, которые используются при создании произведений искусства.
5. Экскурсия — сущность, хранящая в себе информацию о проводимых в галерее экскурсиях: их названия, время проведения и продолжительность.
6. Экскурсовод — сущность, описывающая работников, проводящих экскурсии по экспозиции галереи. Для них хранится информация о их полном имени, а также их ученая степень(или отсутствие таковой).
7. Билет — сущность, хранящая в себе информацию о дате и типе купленного билета.
8. Посетитель — сущность содержит в себе информацию о посетителях галереи: их фамилии, имена, отчества и почты.
9. Отзыв: при желании, посетитель может оставить отзыв о галерее. В отзыв входит как оценка, так и комментарий посетителя.

### 3.5.Атрибуты каждой сущности с указанием первичных ключей

Сущность: Автор

Атрибуты: AID, Фамилия, Им

Первичный ключ: AID

Сущность: Произведение

Атрибуты: Музейный номер, Название, Страна, Год.

Первичный ключ: Музейный номер

Сущность: Коллекция

Атрибуты: CID, Название



Первичный ключ: CID

Сущность: Материал

Атрибуты: MID, Название

Первичный ключ: MID

Сущность: Экскурсия

Атрибуты: EID, Название, Время, Продолжительность

Первичный ключ: EID

Сущность: Экскурсовод

Атрибуты: GID, Фамилия, Имя, Отчество, Ученая степень

Первичный ключ: GID

Сущность: Билет

Атрибуты: TID, Дата, Тип

Первичный ключ: TID

Сущность: Посетитель

Атрибуты: VID, Фамилия, Имя, Отчество, Почта

Первичный ключ: VID

Сущность: Отзыв

Атрибуты: Оценка, текст

Первичный ключ: TID

### 3.6. Описание связей между сущностями

#### 1. Автор\_Произведение

Данная связь имеет тип Один ко многим, так как у автора может быть несколько произведений, а у произведения — только один автор.

#### 2. Коллекция\_Произведение

Данная связь имеет тип Один ко многим, так как произведение входит только в одну коллекцию, в коллекции содержится множество произведений.

#### 3. Материал\_Произведение

Связь имеет тип Многие ко многим, так как произведение может быть сделано из нескольких материалов, а один материал может быть использован при создании множества произведений

4. Произведение\_Экскурсия

Связь имеет тип Многие ко многим, ведь одно произведение может входить в несколько экскурсий, в экскурсию входит множество произведений.

5. Экскурсия\_Экскурсовод

Связь имеет тип Один ко многим, так как один экскурсовод может вести несколько экскурсий, у экскурсии только один экскурсовод

6. Экскурсия\_Билет

Это связь типа Один ко многим, так как экскурсия может быть включена во множество билетов, но в одном билете может быть максимум одна экскурсия.

7. Билет\_Посетитель

Данная связь имеет тип Один ко многим, так как одному билету соответствует строго один посетитель, у посетителя может быть много билетов.

8. Билет\_Отзыв

Данная связь имеет тип Один к одному, так как отзыв соответствует купенному билету, и при этом у билета может быть максимум один отзыв.

### 3.7.Алгоритмы обработки данных, используемые в сценариях

Первый алгоритм - линейный поиск нужного значения в таблице. Он является наиболее простым, но долгим по выполнению, так как имеет сложность  $O(n)$ , где  $n$  — это количество строк.

Второй алгоритм - бинарный поиск. Он работает существенно быстрее, так как его сложность равна  $\log_2(n)$ , где  $n$  — это количество строк. Поиск с использованием индексов возможен при помощи данного алгоритма, т.е. происходит постоянное уменьшение диапазона поиска в 2 раза, пока не дойдёт до минимума(обычно несколько строк/совсем небольшой диапазон)

### 3.8.Планируемый перечень отчетных форм

Прежде всего - финансовые отчёты по выручке/прибыли. А также отчёты, связанные с показателем эффективности экскурсоводов и занятости галереи по различным периодам, так как по ним можно оптимизировать работу персонала и назначать бонусы/премии для лучших сотрудников-экскурсоводов.

### 3.9.Возможная архитектура программного продукта на основе вашей базы данных

База данных позволит создать десктопное приложение, которым смогут пользоваться сотрудники музея. На главной странице будет располагаться окошки для ввода данных, например, окошко №1 - название экспоната, окошко №2 - автор, окошко №3 - музейный номер. Это позволит сотрудникам с соответствующими правами и обязанностями быстро искать интересующие экспонаты и просматривать и изменять информацию по ним.

Во-первых, им будут доступны отчеты и View, которые помогут отслеживать динамику и основные характеристики работы музея, такие как посещения, прибыль, оценки посетителей в разрезе времени. Этот функционал реализуется в формате окон и фильтра по датам(бегунок или календарь) без возможности редактирования отчета.

Во-вторых, в распоряжении сотрудников будут фильтры и функции, которые помогут быстро искать и просматривать информацию.

Функции могут быть реализованы в виде кнопок (если они не требуют ничего на вход) или в виде кнопок и окошек для ввода данных.

Также базу данных можно сочетать с продуктами для ведения учета, например, поставляемыми SAP и 1С. Это позволит быстро заполнять информацию по посещениям.

## 4. Модель данных

### 4.1. Инфологическая модель предметной области

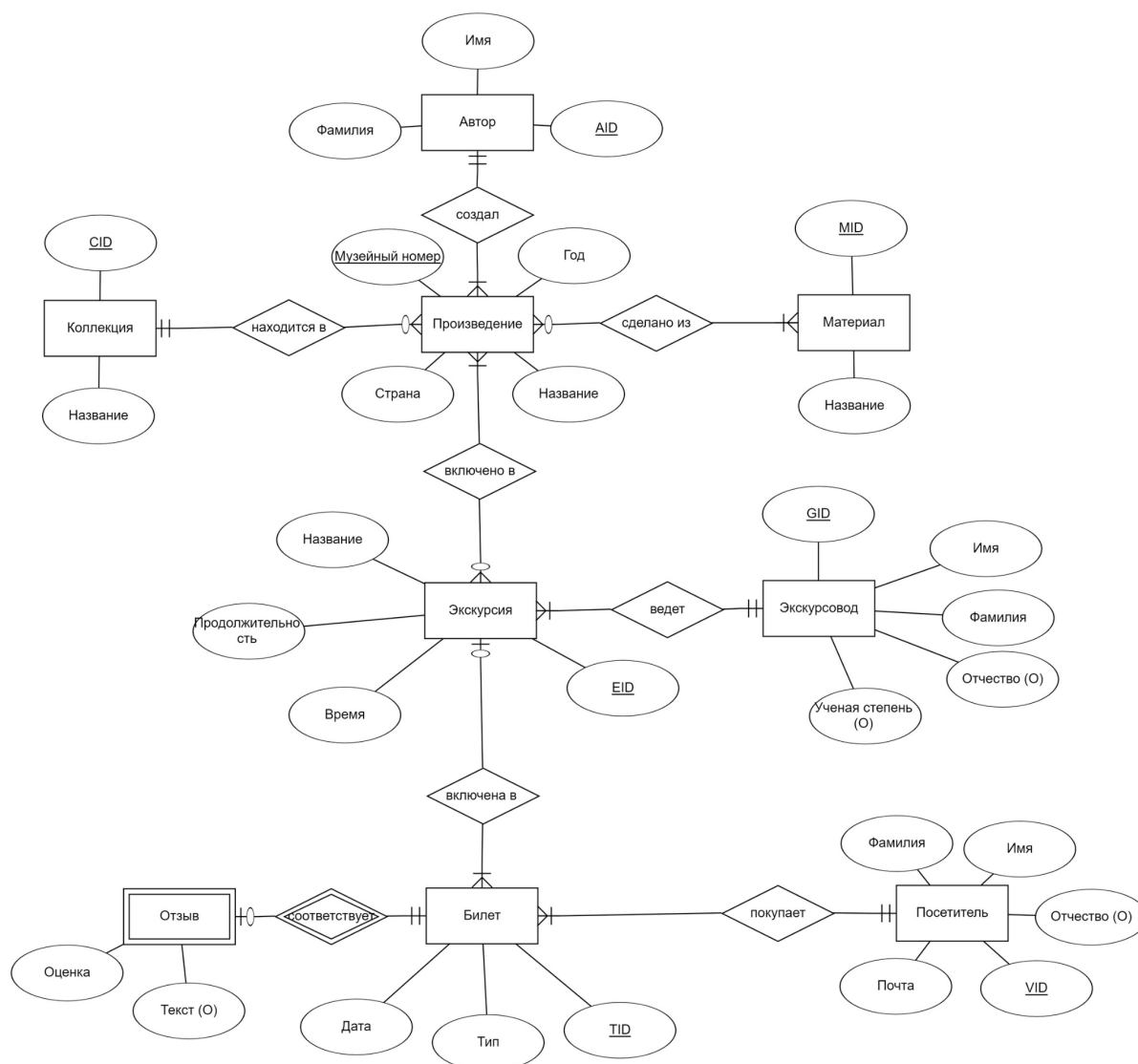


рис.1. Инфологическая модель

### 4.2. Даталогическая модель

На основе инфологической модели мы сформировали даталогическую, добавив таблицу-справочник и пререкрестные таблицы для реализации связи многие ко многим.

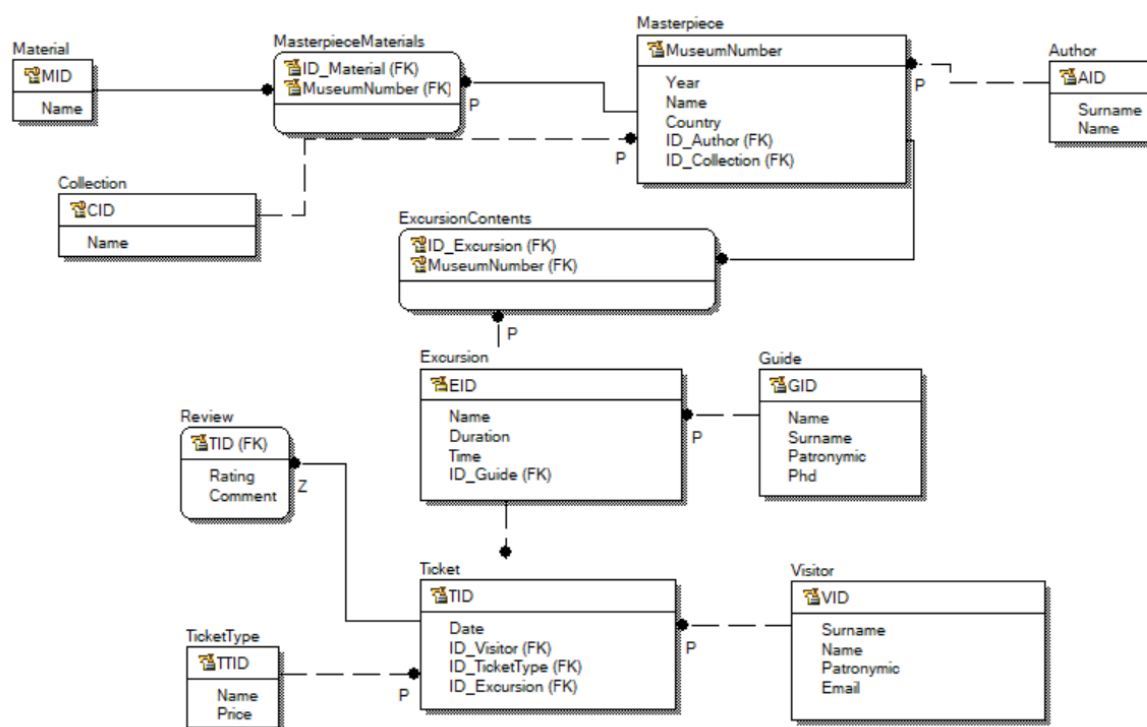


рис.2. Даталогическая модель

### 4.3. Диаграмма с типами данных

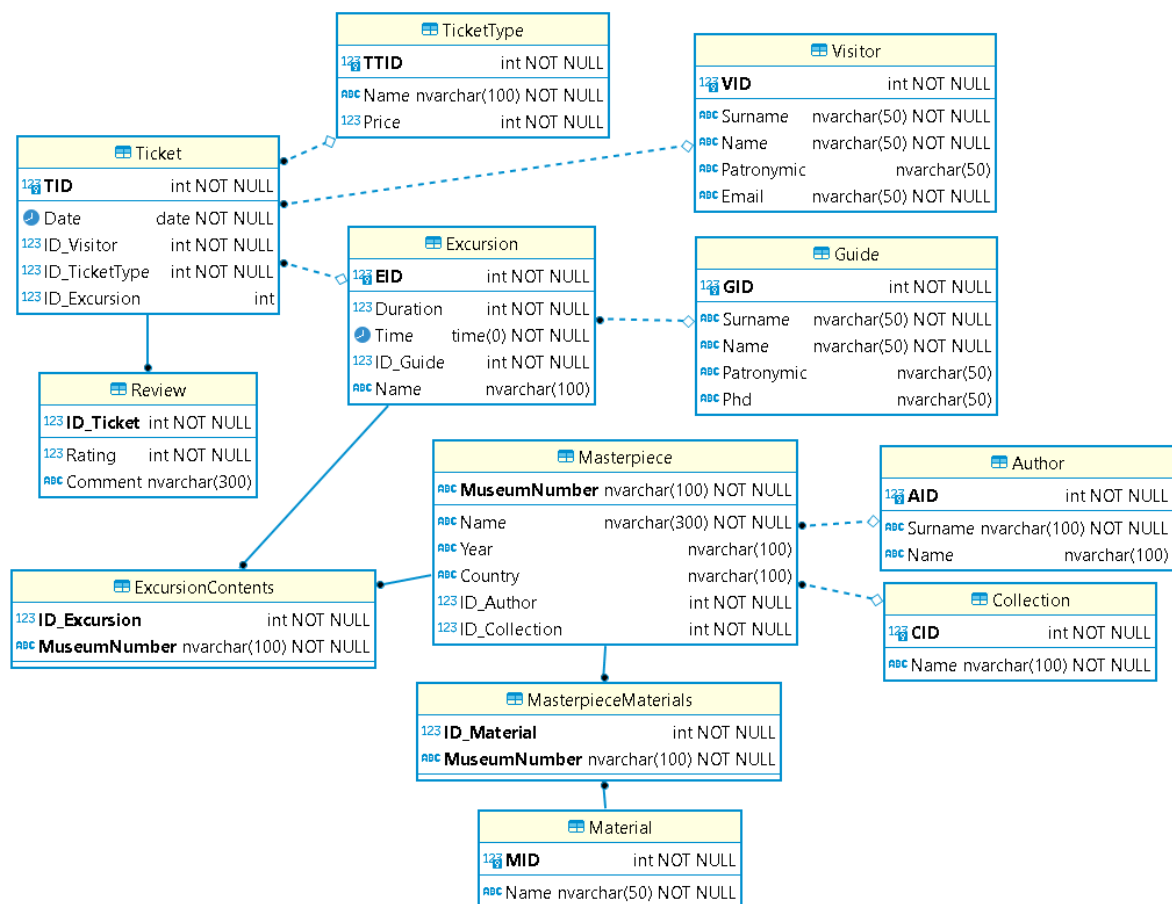


рис.3. Диаграмма с типами данных

### 4.4. Нормализация

Разработанная нами база данных находится в третьей нормальной форме, так как:

- 1) Она находится в первой нормальной форме, то есть атрибуты ее таблиц являются атомарными, отсутствуют повторяющиеся группы
- 2) Она находится во второй нормальной форме, то есть она находится в первой нормальной форме и для каждой таблицы определен первичный ключ и все атрибуты таблицы зависят от него
- 3) В таблицах отсутствуют транзитивные зависимости.

## 5. Заполнение таблиц данными

Заполнение таблиц Masterpiece, Material, Author, AuthorMaterial:

Для парсинга и обращения с данными использовались библиотеки BeautifulSoup, requests, pandas

С сайта <https://collection.pushkinmuseum.art/entity/OBJECT?otdel=14> с помощью инструментов разработчика брались ссылки на каждую из страниц, содержащих информацию об определенных выставках. Затем с каждой страницы вытаскивались ссылки на каждый экспонат.

```
def get_links(collection_link):
    s = collection_link
    base = "https://collection.pushkinmuseum.art"

    page = requests.get(s, verify=False).text
    soup = BeautifulSoup(page)
    table = soup.find("iss-masonry", {"class": "cards-search__content__list"})
    links_list = [base+w["href"] for w in table.find_all("a")]
    return links_list

def interate_links(fund, pages_cnt): #https://collection.pushkinmuseum.art/entity/OBJECT?page=" + "&otdel=14&fund=13"
    art_list = []
    link_part1 = "https://collection.pushkinmuseum.art/entity/OBJECT?page="
    link_part2 = "&otdel=14&fund="
    for i in range(1, pages_cnt+1):
        link = link_part1 + str(i) + link_part2 + str(fund)
        art_list = art_list + get_links(link)
    return art_list

[ ] #Живопись https://collection.pushkinmuseum.art/entity/OBJECT?page=2&otdel=14&fund=13
drawing_list = interate_links(13, 30)

#Скульптура https://collection.pushkinmuseum.art/entity/OBJECT?page=2&otdel=14&fund=35
sculpture_list = interate_links(35, 7)

#Декоративно прикладное https://collection.pushkinmuseum.art/entity/OBJECT?otdel=14&fund=36
decoration_list = interate_links(36, 3)

#Миниатюра https://collection.pushkinmuseum.art/entity/OBJECT?otdel=14&fund=51
miniature_list = interate_links(51, 2)
```

рис.4. Заполнение таблиц данными часть 1

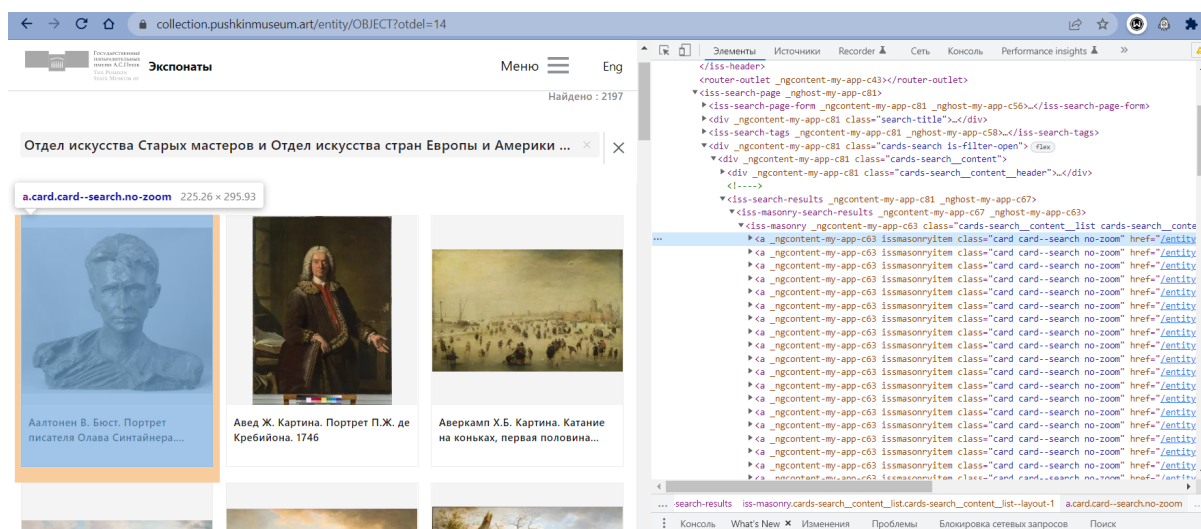


рис.5. Заполнение таблиц данными часть 2

Для каждой ссылки с сайта вытаскивались поля с информацией.

```
def get_dataframe(art_piece):
    s = art_piece
    page = requests.get(s, verify=False).text

    soup = BeautifulSoup(page)
    info_text, info_entity = "iss-info-text-field", "iss-info-entity-field"
    elems_text, elems_entity = soup.find_all(info_text), soup.find_all(info_entity)
    info_text, info_entity = [w.text for w in elems_text], [w.text for w in elems_entity]
    info = [str(part).replace("Читайте далее", '') for part in info_text] + [str(part).replace(" (автор)", '') for part in info_entity]
    dirt = ["(?)", "(круп)", "(автор оригинала)", "(автор модели)", "(последователь)", "(приписывается)", "(часовщик)"]
    for dirty in dirt:
        info = [str(part).replace(dirty, '') for part in info]
    attribute_dict = {"Музейный номер": [], "Страна": [], "Название": [], "Датировка /Время создания": [], "Материал": [], "Техника изготовления": [], "Автор": [], "Коллекция": []}
    for string in info:
        for attribute in attribute_dict:
            if attribute in string:
                s = string.split(attribute)[1]
                attribute_dict[attribute] = s
    if len(attribute_dict["Автор"]) > 0:
        attribute_dict["Фамилия"] = attribute_dict["Автор"].split()[0]
        attribute_dict["Имя"] = ' '.join(attribute_dict["Автор"].split()[1:])
    else:
        attribute_dict["Фамилия"] = ""
        attribute_dict["Имя"] = ""
    attribute_dict

    df = pd.DataFrame.from_dict(attribute_dict, orient='index')
    df = df.transpose()
    return df

[ ] df = pd.DataFrame()

[ ] for item in decoration_list:
    df_new = get_dataframe(item)
    df = df.append(df_new, ignore_index = True)

for item in miniature_list:
    df_new = get_dataframe(item)
    df = df.append(df_new, ignore_index = True)

for item in vosproizvedeniya_list:
    df_new = get_dataframe(item)
    df = df.append(df_new, ignore_index = True)
```

рис.6. Заполнение таблиц данными часть 3

Данные со списком материалов были получены с помощью соединения двух столбцов: техника и материал

```
[ ] df = pd.read_excel('tech.xlsx')
mylist = df['tech'].tolist()

[ ] mystring = str(mylist)
mystring = mystring.replace("'", "")
mystring = mystring.replace(" ", "")
mystring

'[фанеровка, литье, чеканка, золочение, литье, чеканка, патини
е, литье, золочение, роспись, литье, чеканка, золочение, патин
ние, литье, золочение, чеканка, золочение, золочение, литье, ч
ение, литье, чеканка, золочение, золочение, выемчатая эмаль, р
т...'
```

```
[ ] mystring = str(mylist)
mystring = mystring.replace("'", "")
mystring = mystring.replace(" ", "")
mynewlist = mystring.split(',')
mynewlist = list(map(lambda x: x.strip(' '), mynewlist))
clearlist = list(set(mynewlist))
clearlist[1]

'терракота (темно-коричневая)'
```

```
df = pd.DataFrame(clearlist)
df
```

рис.7. Заполнение таблиц данными часть 4



Вся информация загружалась в “сыром” виде в базу данных, после чего каждая из таблиц заполнялась при помощи SQL-запросов

Column1	Номер	Страна	Название	Дата	Материал	Техника	Автор	Коллекция	Фамилия	Имя
178	213 ОИ-14		Дон Жуан.	1981	оргалит, ткань, шёлк, ж	ассамбляж	Хасиор Владисла	Объекты и инс	Хасиор	Владислав
179	214 ОИ-11		Штандарт Святого Хубер	1982	ткань, фанера	ассамбляж	Хасиор Владисла	Объекты и инс	Хасиор	Владислав
180	215 ОИ-15		Русалка	1985	папье-маше, стекло, ж	ассамбляж	Хасиор Владисла	Объекты и инс	Хасиор	Владислав
181	216 ОИ-16		Двуликий наблюдатель.	1985	ткань, дерево, картон	ассамбляж	Хасиор Владисла	Объекты и инс	Хасиор	Владислав
182	217 ОИ-10		Штандарт догматика	1987	ткань	ассамбляж	Хасиор Владисла	Объекты и инс	Хасиор	Владислав
183	218 ОИ-12		Штандарт Архангела	1987	ткань, металл	ассамбляж	Хасиор Владисла	Объекты и инс	Хасиор	Владислав
184	219 ОИ-13		Сельский лектор	1987	металл, дерево, папье	ассамбляж	Хасиор Владисла	Объекты и инс	Хасиор	Владислав
185	220 ОИ-27		Штандарт делегата	1987	шёлк, металл, картон,	ассамбляж	Хасиор Владисла	Объекты и инс	Хасиор	Владислав
186	225 ОИ-17		Драма III-026	1962	ткань на картоне, гипс	ассамбляж	Шайна Юзеф	Объекты и инс	Шайна	Юзеф
187	228 ОИ-22			1981	холст (на фанере), тка	ассамбляж	Штейн Йонаш	Объекты и инс	Штейн	Йонаш
188	67 М-486		Постамент под скульптур кон. XVIII в.		мрамор (темно-красн	гравировка		Декоративно-г		
189	140 Ж-3241	Франция	Сабиянки	вторая половина XIX	веки	кость	гуашь	Жерар Луи Гасто	Миниатюра	Жерар Луи Гастон
190	141 Ж-3139	Франция	Портрет Наполеона Бона	1798 - 1806	кость	гуашь	Изабе Жан-Батист	Миниатюра	Изабе Жан-Батист	Жан-Батист
191	142 Ж-2757	Франция	Портрет генерала Жана	1800-1804	кость	гуашь	Изабе Жан-Батист	Миниатюра	Изабе Жан-Батист	Жан-Батист
192	129 Ж-2329	Франция	Портрет дамы в черном	1790	кость	гуашь	Бозиль Жан Жак	Миниатюра	Бозиль Жан Жак	Гийом
193	130 Ж-2331	Италия	Портрет дамы в открыто	1805	кость	гуашь	Босси Доменико	Миниатюра	Босси Доменико	Доменико
194	131 Ж-2230	Франция	Портрет графа Ю.А. Голс	последняя четверть XVIII	кость	гуашь	Герен Жан Урбен	Миниатюра	Герен Жан Урбен	Жан Урбен
195	132 Ж-2231	Франция	Портрет графини Е.Л.Гол	последняя четверть XVIII	кость	гуашь	Герен Жан Урбен	Миниатюра	Герен Жан Урбен	Жан Урбен
196	133 Ж-2232	Франция	Портрет дамы в черном	1818 - 1820	кость	гуашь	Герен Жан Урбен	Миниатюра	Герен Жан Урбен	Жан Урбен
197	134 Ж-2791	Австрия	Портрет Марии де Ла Ро	первая половина XIX	века	кость	гуашь	Даффингер Мори	Миниатюра	Даффингер Мориц
198	209 ОИ-33		Три женщины в троллейб	1970-е гг.	бумага (на шите оргал	гуашь	Турецкий Борис	Объекты и инс	Турецкий Борис	Захарович
199	210 ОИ-34		Две девушки в троллейб	1970-е гг.	бумага	гуашь	Турецкий Борис	Объекты и инс	Турецкий Борис	Захарович
200	153 Ж-2464	Франция	Портрет французского а	1780-е	кость	гуашь	Неизвестный мас	Миниатюра	Неизвестный мас	Неизвестный мастер
201	155 Ж-2291	Франция	Портрет А.-Ж. Теруань д	Начало 1790-х	кость	гуашь	Неизвестный мас	Миниатюра	Неизвестный мас	Неизвестный мастер
202	156 Ж-3595	Франция	Портрет князя Дмитрия	1824	кость	гуашь	Неизвестный мас	Миниатюра	Неизвестный мас	Неизвестный мастер
203	144 Ж-4135	Франция	Портрет графа Карла Ла	1814	кость	гуашь	Лаби Огюст Фран	Миниатюра	Лаби Огюст Франсуа	Огюст Франсуа
204	145 Ж-2442	Франция	Портрет К.А. Нарышкина	1817-1825	кость	гуашь	Лагрене Антельм	Миниатюра	Лагрене Антельм	Франсуа
205	146 Ж-2244	Франция	Портрет дамы в оранже	Около 1796	кость	гуашь	Ларивьер-Куанс	Миниатюра	Ларивьер-Куанс	Жан-Батист Виктор

рис.8. Заполнение таблиц данными часть 5

Пример заполнения таблицы Masterpiece.

```

INSERT into Masterpiece (MuseumNumber, Name, Year, Country, ID_Collection, ID_Author)
SELECT DISTINCT Номер, Название, Дата, Страна, Collection.CID, Author.AID
FROM NewTable_2_PAMAGITE as pom
INNER JOIN Collection
ON Collection.Name = пом.Коллекция
INNER JOIN Author
ON Author.Surname = пом.Фамилия
AND Author.Name = пом.Имя
WHERE Название is NOT NULL;

```

рис.9. Заполнение таблиц данными часть 6

Пример заполнения таблицы MasterpieceMaterials

```

INSERT into MasterpieceMaterials (MuseumNumber, ID_Material)
SELECT DISTINCT MuseumNumber, MID
FROM NewTable_2_PAMAGITE as pom
INNER JOIN Material
ON pom.Техника like '%' + Material.Name + '%'
OR pom.Материал like '%' + Material.Name + '%'
INNER JOIN Masterpiece
ON Masterpiece.MuseumNumber = пом.Номер;

```

рис.10. Заполнение таблиц данными часть 7

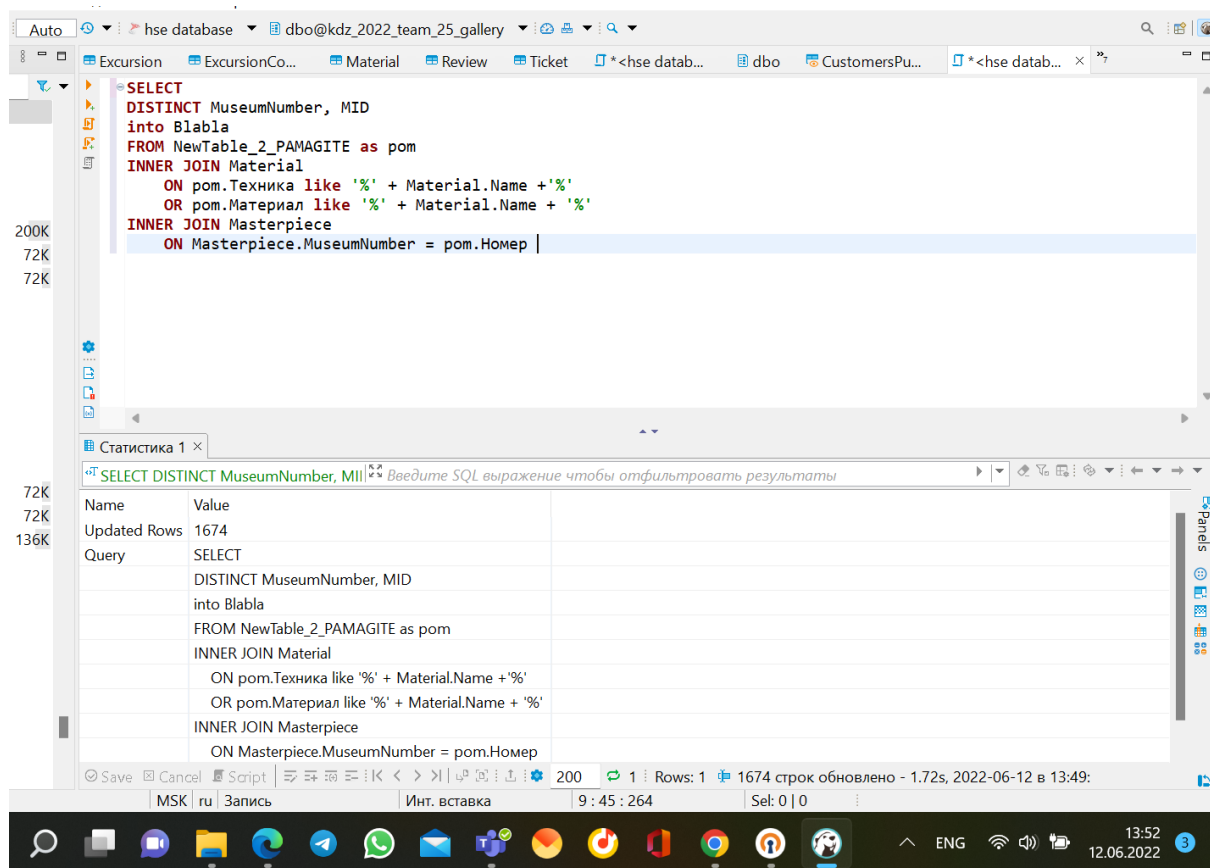


рис.11. Заполнение таблиц данными часть 8

Заполнение таблицы Visitor:

Из открытого источника были взяты имена и фамилии, email был создан по принципу “имя+фамилия+@gmail.com”. Сохранённый файл был импортирован в таблицу Visitor. Ниже представлен код на языке Python.

```

Visitor

In [3]: data_names_us = pd.read_csv('Data/us_names.txt', delimiter = "\t")

In [4]: data_surnames_us = pd.read_csv('Data/us_surnames.txt', delimiter = "\t")

In [5]: df_elements_names = data_names_us.sample(n=30)

In [6]: df_elements_surnames = data_surnames_us.sample(n=30)

In [7]: df_elements_names['key'] = 0
df_elements_surnames['key'] = 0
surnames_names_us = df_elements_surnames.merge(df_elements_names, on='key', how='outer')

In [8]: surnames_names_us = surnames_names_us.rename(columns = {'Aaberg': 'Surname'})

In [9]: surnames_names_us = surnames_names_us.rename(columns = {'Aaron': 'Name'})

In [10]: surnames_names_us = surnames_names_us.drop(columns=['key'], axis=1)

In [11]: surnames_names_us['NameSurname'] = surnames_names_us['Name'].str.cat(surnames_names_us['Surname'], sep = "")

In [12]: surnames_names_us['Email'] = surnames_names_us['NameSurname'].apply(lambda x: x + '@gmail.com')

In [13]: from sklearn.utils import shuffle

In [14]: df_copy = shuffle(surnames_names_us)

In [15]: df_copy.reset_index(inplace=True, drop=True)

In [16]: df_copy=df_copy.drop(columns=['NameSurname'], axis=1)

In [17]: df_copy.head(3)
Out[17]:
   Surname  Name  Email
0  Sherlin  Bernarda  BernardaSherlin@gmail.com
1  Wesberry  Margorie  MargorieWesberry@gmail.com
2    Bible    Arica    AricaBible@gmail.com

In [18]: len(df_copy['Email'].unique())
Out[18]: 900

In [30]: df_copy.to_csv('USA_Names_Surnames_Emails.csv')

```

рис.12. Заполнение таблицы Visitor

## Заполнение таблицы Ticket:

Данная таблица была заполнена с помощью ранее созданных таблиц Экскурсия и Посетитель. С помощью функции рандом были созданы билеты. Полученный файл был импортирован в таблицу Ticket. Ниже представлен код на языке Python.

#### Ticket

```
In [19]: import random
        from random import randint

In [20]: ticket_types_not_ex = [0, 1, 2]

In [21]: ticket_types_ex = [2, 3, 4]

In [22]: excursions = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]

In [23]: df_tickets_n = pd.DataFrame(columns=['Date', 'ID_Visitor', 'ID_TicketType', 'ID_Excursion'])

In [24]: dates = pd.read_csv('new_cases_RUS.csv', sep=',')

In [25]: dates = dates['date']

In [26]: dates_excursions = dates.sample(n=300)

In [27]: dates_excursions = dates_excursions.tolist()

In [28]: dates_not_ex = set(dates.tolist()).intersection(dates_excursions)

In [29]: for i in dates_excursions:
        df_tickets_n = df_tickets_n.append({'Date': i, 'ID_Visitor': randint(0, 900), 'ID_TicketType': random.choice(ticket_types_ex), 'ID_Excursion': random.choice(excursions)})

In [31]: for i in dates_not_ex:
        df_tickets_n = df_tickets_n.append({'Date': i, 'ID_Visitor': randint(0, 900), 'ID_TicketType': random.choice(ticket_types_not_ex), 'ID_Excursion': random.choice(excursions)})

In [32]: df_tickets_n = df_tickets_n.sort_values('Date')

In [33]: df_tickets_n = df_tickets_n.reset_index()

In [69]: df_tickets_n.to_csv('Excursion.csv')
```

рис.13. Заполнение таблицы Ticket

#### Заполнение таблицы ExcursionContents:

Эта таблица была заполнена с помощью распределения произведений между сделанными ранее экскурсиями и экскурсоводами. Полученный файл был импортирован в таблицу ExcursionContents. Продолжительность некоторых экскурсий была изменена позднее. Ниже представлен код на языке Python.

#### ExcursionContents

```
In [72]: masterpieces = pd.read_csv('Data/Masterpiece_202206100017.csv', sep=',')

In [79]: masterpieces_id = masterpieces['MuseumNumber'].values.tolist()

In [77]: excursions = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]

In [110]: df_ExcursionContent = pd.DataFrame(columns=['ID_Excursion', 'MuseumNumber'])

In [112]: for i in range(2, 5):
           num_of_creations = random.sample(masterpieces_id, 15)
           for k in num_of_creations:
               df_ExcursionContent = df_ExcursionContent.append({'ID_Excursion': i, 'MuseumNumber': k}, ignore_index=True)
           ...

In [115]: for i in range(5, 8):
           num_of_creations = random.sample(masterpieces_id, 22)
           for k in num_of_creations:
               df_ExcursionContent = df_ExcursionContent.append({'ID_Excursion': i, 'MuseumNumber': k}, ignore_index=True)
           ...

In [117]: for i in range(8, 11):
           num_of_creations = random.sample(masterpieces_id, 30)
           for k in num_of_creations:
               df_ExcursionContent = df_ExcursionContent.append({'ID_Excursion': i, 'MuseumNumber': k}, ignore_index=True)
           ...

In [118]: for i in range(11, 14):
           num_of_creations = random.sample(masterpieces_id, 50)
           for k in num_of_creations:
               df_ExcursionContent = df_ExcursionContent.append({'ID_Excursion': i, 'MuseumNumber': k}, ignore_index=True)
           ...

In [119]: for i in range(14, 17):
           num_of_creations = random.sample(masterpieces_id, 100)
           for k in num_of_creations:
               df_ExcursionContent = df_ExcursionContent.append({'ID_Excursion': i, 'MuseumNumber': k}, ignore_index=True)
           ...

In [120]: len(df_ExcursionContent)
Out[120]: 651

In [122]: df_ExcursionContent.to_csv('ExcursionContent.csv')
```

рис.14. Заполнение таблицы ExcursionContents

#### Заполнение таблицы Review:

Для данной таблицы были скачаны из DBeaver ID билетов и с помощью функции random были сделаны оценки от 1 до 10. Полученный файл был импортирован в таблицу Review. Ниже приведен код на языке Python.

```

In [5]: list_tic_id = pd.read_csv('Data/Ticket_202206111554.csv')

In [11]: rev_350 = list_tic_id.sample(n=350)

In [18]: id_1 = rev_350['TID'].values

In [20]: print(len(set(id_1)))

350

In [22]: df_ratings = pd.DataFrame(columns=['ID_Ticket', 'Rating'])

In [24]: for i in id_1:
          df_ratings = df_ratings.append({'ID_Ticket': i, 'Rating': randint(1, 10)}, ignore_index=True)

...

In [28]: df_ratings = df_ratings.sort_values('ID_Ticket')

In [29]: df_ratings
Out[29]:
   ID_Ticket  Rating
309         3       3
80          7       2
22          9       3
259        11       3
96         14       5
...         ...     ...
180       1045       3
313       1051       1
311       1055       7
129       1058       8
64        1062       8

350 rows x 2 columns

In [30]: df_ratings.to_csv('Rating_sort.csv')

```

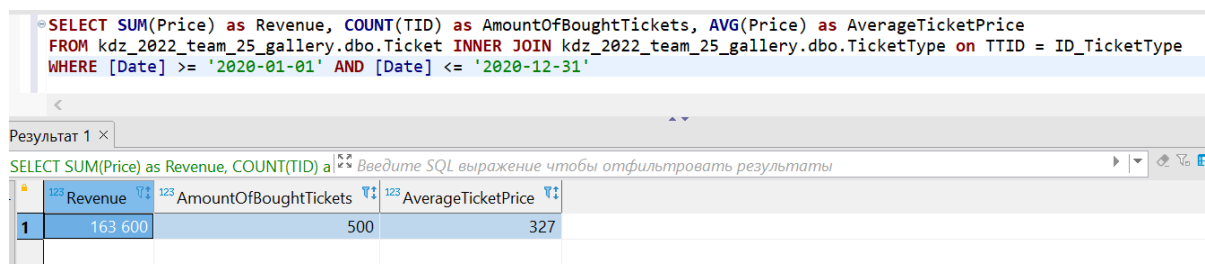
рис.15. Заполнение таблицы Review

## 6. Разработка СУБД

### 6.1. Запросы

#### 1) Простой запрос с условием и формулами в SELECT(1)

Выводит выручку, количество купленных билетов и их среднюю стоимость за 2020 год



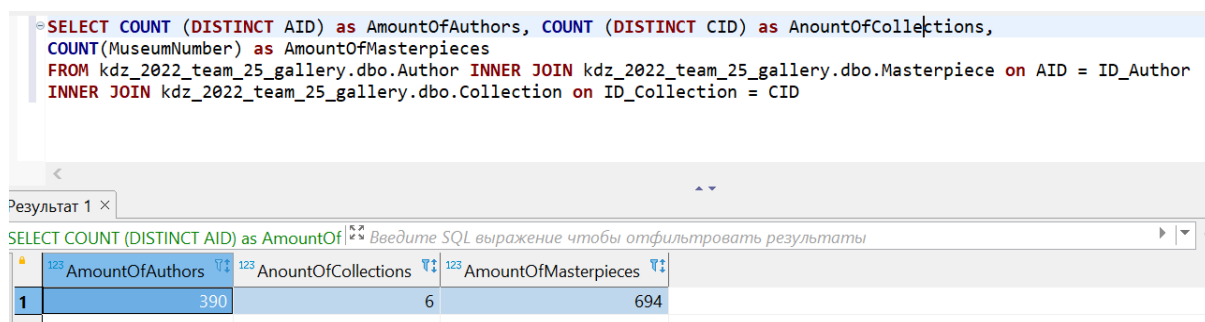
```
SELECT SUM(Price) as Revenue, COUNT(TID) as AmountOfBoughtTickets, AVG(Price) as AverageTicketPrice
FROM kdz_2022_team_25_gallery.dbo.Ticket INNER JOIN kdz_2022_team_25_gallery.dbo.TicketType on TTID = ID_TicketType
WHERE [Date] >= '2020-01-01' AND [Date] <= '2020-12-31'
```

	Revenue	AmountOfBoughtTickets	AverageTicketPrice
1	163 600	500	327

рис.16. Запрос №1

#### 2) Простой запрос с условием и формулами в SELECT(2)

Выводит краткую информацию о постоянной экспозиции.



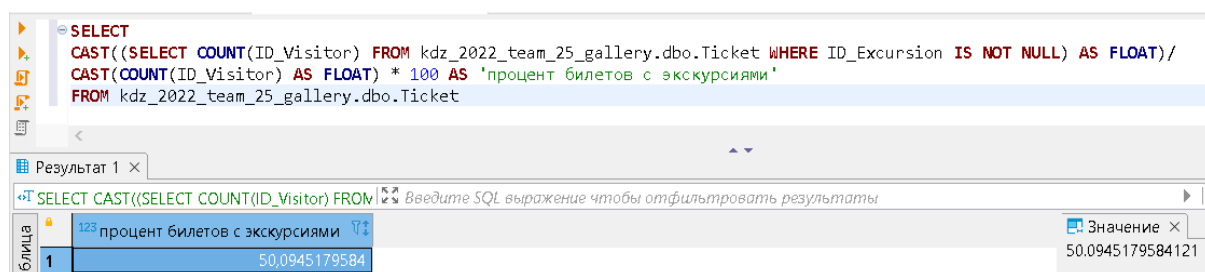
```
SELECT COUNT (DISTINCT AID) as AmountOfAuthors, COUNT (DISTINCT CID) as AmountOfCollections,
COUNT(MuseumNumber) as AmountOfMasterpieces
FROM kdz_2022_team_25_gallery.dbo.Author INNER JOIN kdz_2022_team_25_gallery.dbo.Masterpiece on AID = ID_Author
INNER JOIN kdz_2022_team_25_gallery.dbo.Collection on ID_Collection = CID
```

	AmountOfAuthors	AmountOfCollections	AmountOfMasterpieces
1	390	6	694

рис.17. Запрос №2

#### 3) Запрос с коррелированным подзапросом в SELECT(1)

Выводит процент билетов, которые были куплены с экскурсией



```
SELECT
CAST((SELECT COUNT(ID_Visitor) FROM kdz_2022_team_25_gallery.dbo.Ticket WHERE ID_Excursion IS NOT NULL) AS FLOAT)/
CAST(COUNT(ID_Visitor) AS FLOAT) * 100 AS 'процент билетов с экскурсиями'
FROM kdz_2022_team_25_gallery.dbo.Ticket
```

	процент билетов с экскурсиями
1	50.0945179584

рис.18. Запрос №3

#### 4) Запрос с коррелированным подзапросом в SELECT(2)

Выводит количество купленных билетов для конкретного клиента(в данном случае для клиента с номером 14)

```

SELECT v.Name, v.Surname, (SELECT COUNT(t.ID) FROM kdz_2022_team_25_gallery.dbo.Ticket t WHERE t.ID_Visitor = 14) AS 'количество купленных билетов'
FROM kdz_2022_team_25_gallery.dbo.Visitor v
WHERE v.VID = 14

```

Имя	Фамилия	количество купленных билетов
Palmer	Macphail	4

рис.19. Запрос №4

## 5) Запрос с подзапросом в FROM(1)

Выводит среднее количество произведений в коллекции.

```

SELECT AVG(MasterpiecesPerCollection) AS AvgMasterpiecesPerCollection
FROM kdz_2022_team_25_gallery.dbo.Collection C
INNER JOIN
(SELECT ID_Collection, COUNT(MuseumNumber) AS MasterpiecesPerCollection
FROM kdz_2022_team_25_gallery.dbo.Masterpiece
GROUP BY ID_Collection) AS MPC
ON C.CID=MPC.ID_Collection

```

AvgMasterpiecesPerCollection
99

рис.20. Запрос №5

## 6) Запрос с подзапросом в FROM(2)

Выводит среднее количество произведений в экскурсии

```

SELECT AVG(MasterpiecesPerExcursion) AS AvgMasterpiecesPerExcursion
FROM kdz_2022_team_25_gallery.dbo.Excursion E
INNER JOIN
(SELECT ID_Excursion, COUNT(MuseumNumber) AS MasterpiecesPerExcursion
FROM kdz_2022_team_25_gallery.dbo.ExcursionContents EC
GROUP BY ID_Excursion) AS MPE
ON E.EID=MPE.ID_Excursion

```

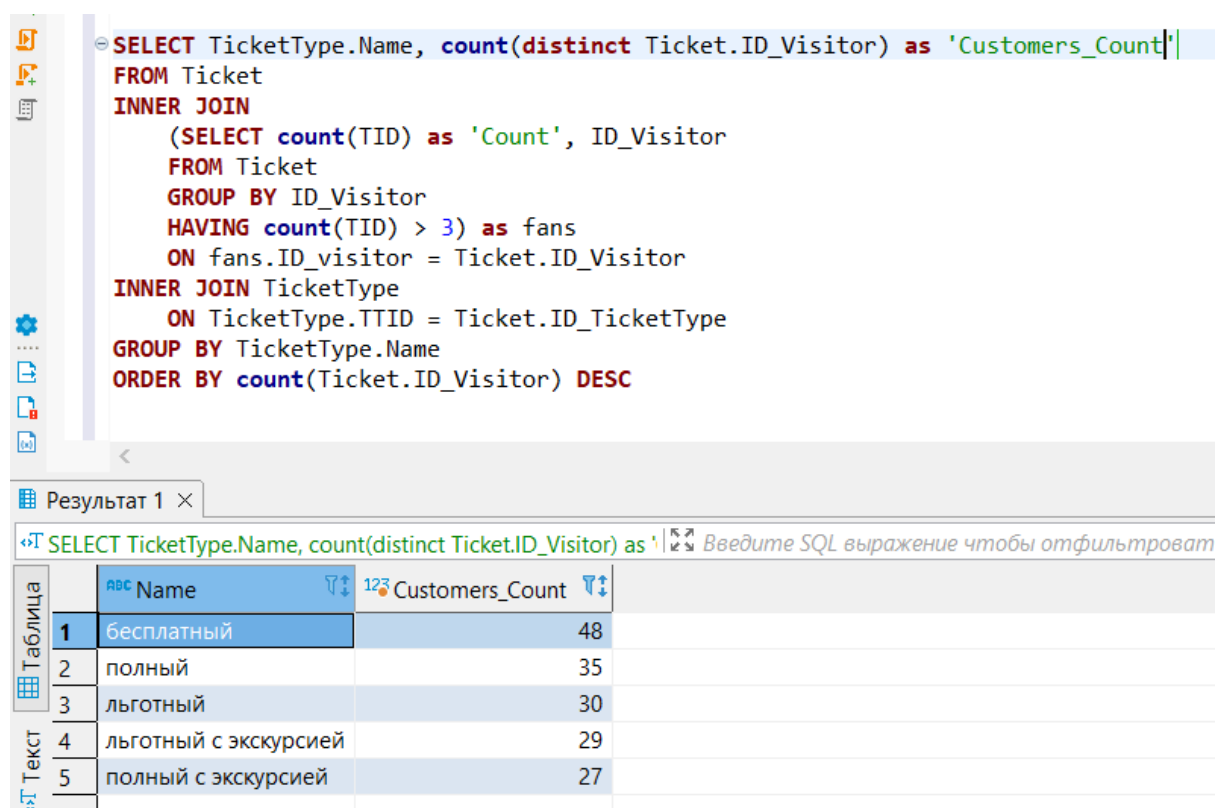
AvgMasterpiecesPerExcursion
43

рис.21. Запрос №6



7) Запрос с подзапросом в FROM, агрегированием, группировкой и сортировкой

Выводит категории и количество лояльных (у них более 3 посещений) клиентов, которые пользуются ими, в порядке приоритетности категории. Может быть полезно для оценки лояльного сегмента посетителей.



```
SELECT TicketType.Name, count(distinct Ticket.ID_Visitor) as 'Customers_Count'
FROM Ticket
INNER JOIN
  (SELECT count(TID) as 'Count', ID_Visitor
   FROM Ticket
   GROUP BY ID_Visitor
   HAVING count(TID) > 3) as fans
ON fans.ID_visitor = Ticket.ID_Visitor
INNER JOIN TicketType
ON TicketType.TTID = Ticket.ID_TicketType
GROUP BY TicketType.Name
ORDER BY count(Ticket.ID_Visitor) DESC
```

Результат 1 ×

SELECT TicketType.Name, count(distinct Ticket.ID\_Visitor) as 'Customers\_Count'

	ABC Name	123 Customers_Count
1	бесплатный	48
2	полный	35
3	льготный	30
4	льготный с экскурсией	29
5	полный с экскурсией	27

рис.22. Запрос №7

8) Запрос с подзапросом в FROM, агрегированием, группировкой и сортировкой

Выводит, какие экскурсии покрывают топ-5(по количеству произведений) авторов, и на сколько %. Может быть полезно для оценки насыщенности экскурсий и свободных посещений. Например, если Франческо Мессина - один из ключевых скульпторов 20 века, а экскурсии покрывают только 40% его произведений, возможно, стоит добавить больше его скульптур в экскурсии.

```

SELECT
CASE WHEN ExcursionContents.ID_Excursion is NULL
THEN 'Свободный осмотр'
ELSE CAST(ExcursionContents.ID_Excursion as VARCHAR(5))
END as 'Покрытие осмотрами, ID экскурсии',
Author.Name + ' ' + Author.Surname as Author,
ROUND(count(Masterpiece.MuseumNumber)*100./top_authors.Count, 2) as 'Процент покрытия'
FROM Masterpiece
INNER JOIN
(SELECT Top(5) count(Masterpiece.MuseumNumber) as 'Count', ID_Author
FROM Masterpiece
GROUP BY Masterpiece.ID_Author
ORDER BY count(Masterpiece.MuseumNumber) DESC
) as top_authors
ON top_authors.ID_Author = Masterpiece.ID_Author
LEFT JOIN ExcursionContents
ON ExcursionContents.MuseumNumber = Masterpiece.MuseumNumber
INNER JOIN Author
ON Author.AID = top_authors.ID_Author
GROUP BY ExcursionContents.ID_Excursion, Author.Name, Author.Surname, top_authors.Count

```

рис.23. Запрос №8 часть 1

	ABC Покрытие осмотрами, ID экскурсии	ABC Author	123 Процент покрытия
1	Свободный осмотр	Антуан Луи Бари	55
2	Свободный осмотр	Аристид Майоль	45,45
3	Свободный осмотр	Владислав Хасиор	40
4	Свободный осмотр	Огюст Роден	58,33
5	Свободный осмотр	Франческо Мессина	60
6	2	Антуан Луи Бари	15
7	2	Франческо Мессина	10
8	3	Франческо Мессина	5
9	4	Владислав Хасиор	10
10	4	Франческо Мессина	5
11	5	Антуан Луи Бари	5
12	5	Франческо Мессина	10
13	6	Антуан Луи Бари	5
14	6	Аристид Майоль	9,09
15	7	Антуан Луи Бари	5
16	8	Аристид Майоль	9,09
17	8	Огюст Роден	8,33
18	9	Аристид Майоль	9,09
19	10	Антуан Луи Бари	10
20	10	Аристид Майоль	9,09
21	10	Огюст Роден	16,67
22	11	Владислав Хасиор	10
23	11	Огюст Роден	8,33
24	12	Франческо Мессина	5
25	13	Антуан Луи Бари	10
26	13	Аристид Майоль	9,09
27	14	Антуан Луи Бари	15
28	14	Аристид Майоль	9,09

рис.24. Запрос №8 часть 2

## 9) Запрос с коррелированным подзапросом в WHERE(1)

Вывод произведений с непустым названием, фамилии авторов которых начинаются с конкретной буквы (здесь - Л).

SQL Query:

```
SELECT m.MuseumNumber, m.Name
FROM kdz_2022_team_25_gallery.dbo.Masterpiece m
WHERE m.ID_Author IN (
SELECT a.AID
FROM kdz_2022_team_25_gallery.dbo.Author a
WHERE a.Surname LIKE 'л%') AND DATALENGTH(m.Name) > 0
```

Результат 1 ×

SQL Query:

```
SELECT m.MuseumNumber, m.Name FROM
```

Введите SQL выражение чтобы отфильтровать результаты

Таблица	МuseumNumber	Name
1	Ж-2244	Портрет дамы в оранжевой шали в парке
2	Ж-2245	Портрет старухи в лиловом платье и палевой шляпке
3	Ж-2248	Портрет дамы в красном бархатном платье
4	Ж-2442	Портрет К.А. Нарышкина в виде камеи
5	Ж-2453	Сельская ярмарка
6	Ж-4135	Портрет графа Карла Ласерда де Кастро
7	ОИ-23	Арка
8	ОИ-24	Башня
9	П-621	Лампа в виде снопа колосьев с сидящими в нем фигурками детей
10	П-931	Каминные часы с двумя обелисками
11	Ск-28	Женский портрет
12	Ск-283	Две женщины.
13	Ск-394	Голова мальчика.
14	Ск-432	Портрет Ф.Э.Дзержинского.
15	Ск-453	Бегство в Египет.
16	Ск-454	Сидящая женщина
17	Ск-455	Прачка
18	Ск-494	Голова бульдога
19	Ск-501	Грузчики
20	Ск-586	Собаки.
21	Ск-605	Голова молодой девушки.
22	Ск-627	Сеятель

рис.25. Запрос №9

#### 10) Запрос с коррелированным подзапросом в WHERE(2)

Выводит количество билетов с экскурсиями, продолжительность которых была более 90 минут

SQL Query:

```
SELECT COUNT(t.TID) AS 'количество билетов'
FROM kdz_2022_team_25_gallery.dbo.Ticket t
WHERE t.ID_Excursion IN (
SELECT e.EID
FROM kdz_2022_team_25_gallery.dbo.Excursion e
WHERE e.Duration > 90)
```

Результат 1 ×

SQL Query:

```
SELECT COUNT(t.TID) AS 'количество бил
```

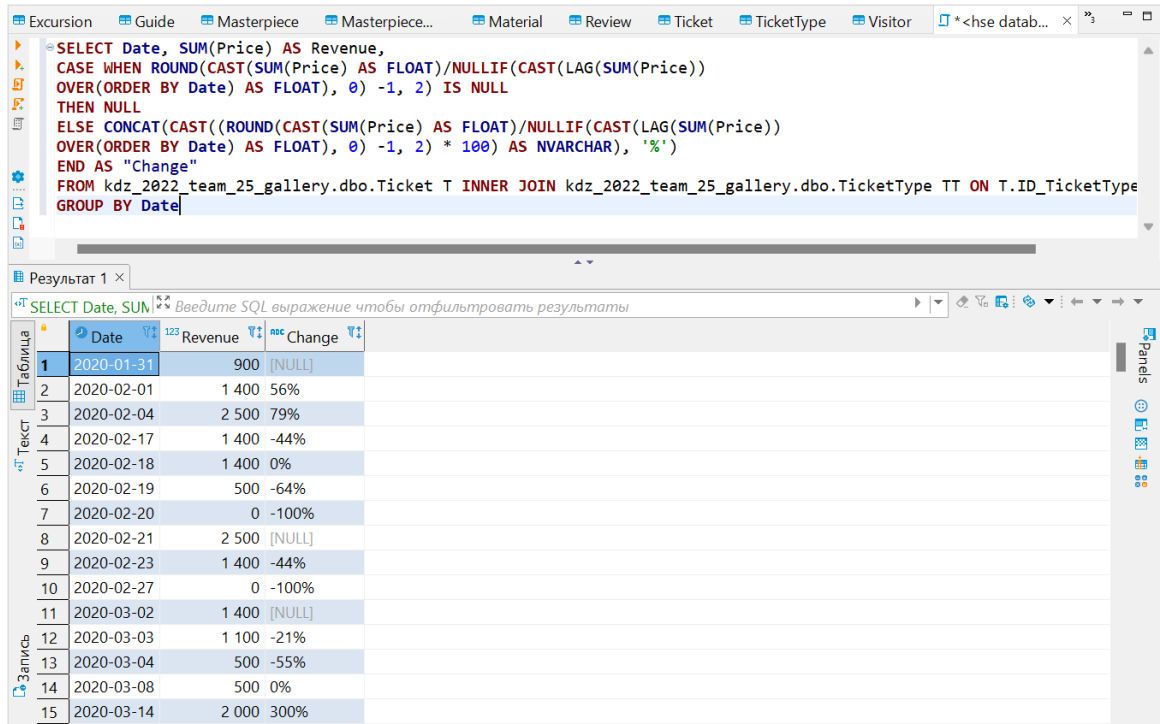
Введите SQL выражение чтобы

Таблица	количество билетов
1	180

рис.26. Запрос №10

- 11) Запрос, использующий оконную функцию LAG или LEAD для выполнения сравнения данных в разных периодах

Выводит процент изменения выручки по сравнению с предыдущим днем.



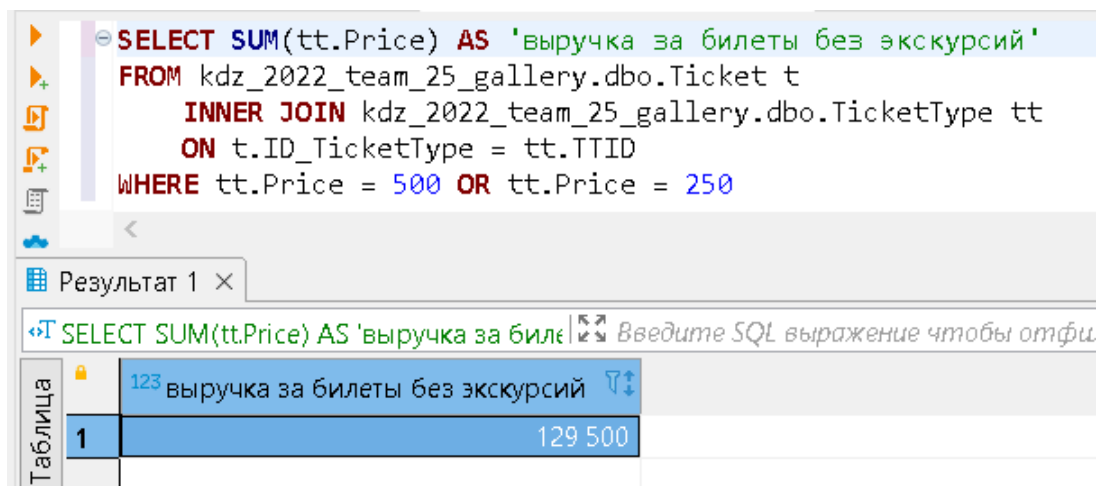
```
SELECT Date, SUM(Price) AS Revenue,
CASE WHEN ROUND(CAST(SUM(Price) AS FLOAT)/NULLIF(CAST(LAG(SUM(Price))
OVER(ORDER BY Date) AS FLOAT), 0) -1, 2) IS NULL
THEN NULL
ELSE CONCAT(CAST((ROUND(CAST(SUM(Price) AS FLOAT)/NULLIF(CAST(LAG(SUM(Price))
OVER(ORDER BY Date) AS FLOAT), 0) -1, 2) * 100) AS NVARCHAR), '%')
END AS "Change"
FROM kdz_2022_team_25_gallery.dbo.Ticket T INNER JOIN kdz_2022_team_25_gallery.dbo.TicketType TT ON T.ID_TicketType
GROUP BY Date
```

	Date	Revenue	Change
1	2020-01-31	900	[NULL]
2	2020-02-01	1 400	56%
3	2020-02-04	2 500	79%
4	2020-02-17	1 400	-44%
5	2020-02-18	1 400	0%
6	2020-02-19	500	-64%
7	2020-02-20	0	-100%
8	2020-02-21	2 500	[NULL]
9	2020-02-23	1 400	-44%
10	2020-02-27	0	-100%
11	2020-03-02	1 400	[NULL]
12	2020-03-03	1 100	-21%
13	2020-03-04	500	-55%
14	2020-03-08	500	0%
15	2020-03-14	2 000	300%

рис.27. Запрос №11

- 12) Запрос с агрегированием и выражением JOIN, включающим не менее 2 таблиц(1)

Выводит выручку за продажи билетов без экскурсий



```
SELECT SUM(tt.Price) AS 'выручка за билеты без экскурсий'
FROM kdz_2022_team_25_gallery.dbo.Ticket t
INNER JOIN kdz_2022_team_25_gallery.dbo.TicketType tt
ON t.ID_TicketType = tt.TTID
WHERE tt.Price = 500 OR tt.Price = 250
```

	выручка за билеты без экскурсий
1	129 500

рис.28. Запрос №12

- 13) Запрос с агрегированием и выражением JOIN, включающим не менее 2 таблиц(2)

Выводит количество мастеров из конкретной страны(здесь -Швейцарии)

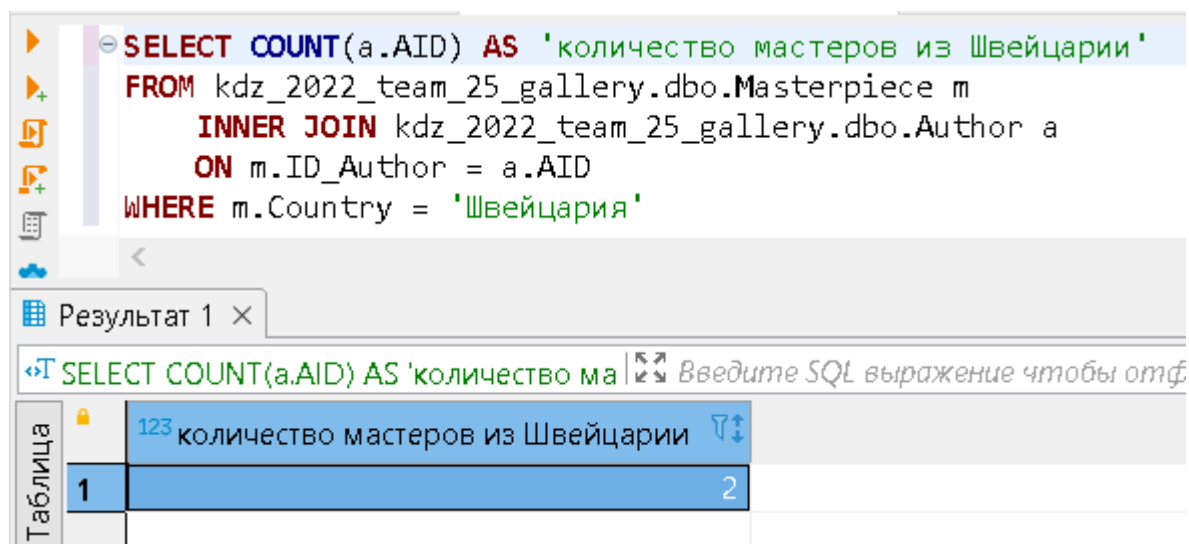


рис.29. Запрос №13

- 14) Запрос с агрегированием и выражением JOIN, включающим не менее 2 таблиц(3)

Выводит количество произведений конкретной коллекции (здесь - “Живопись”)

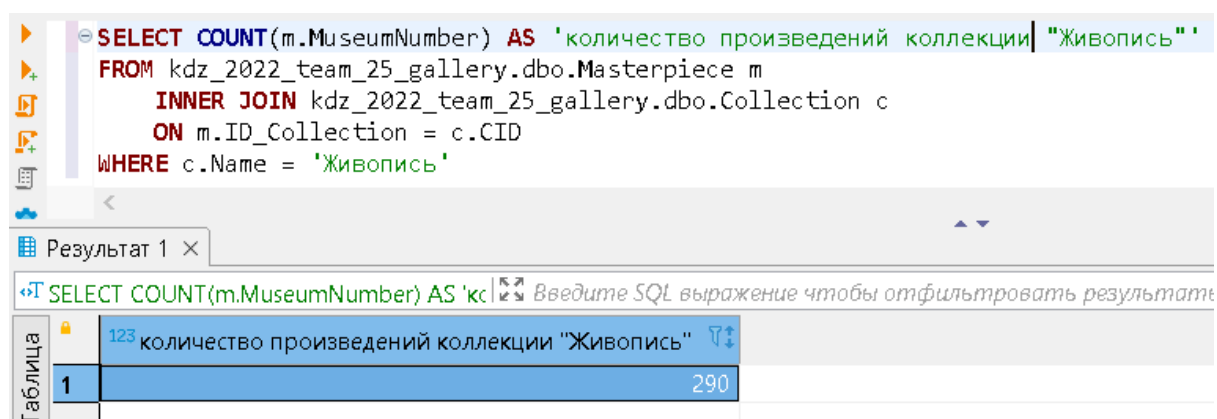


рис.30. Запрос №14

- 15) Запрос, использующий манипуляции с множествами

Выводит выставки, которые посетили в 2021 году и оценили на 10.

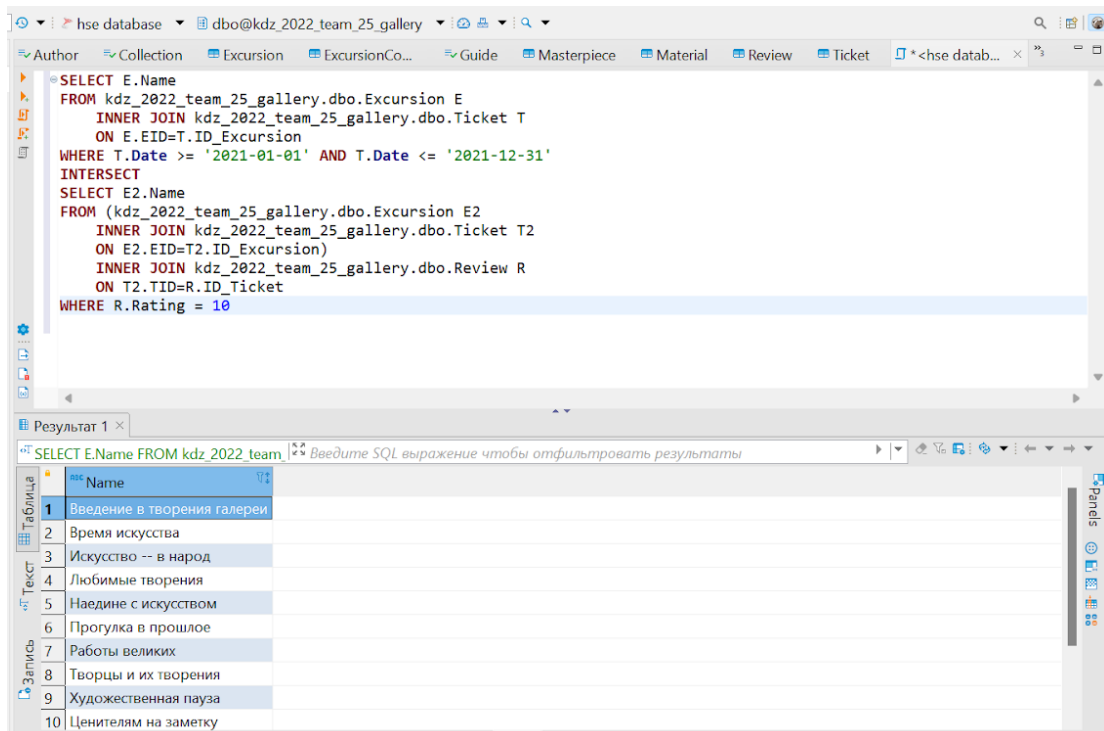


рис.31. Запрос №15

## 16) Запрос с EXISTS

Запрос выводит информацию по клиентам, которые оставляли негативные отзывы. Может быть полезен для последующей связи с клиентами для обратной связи

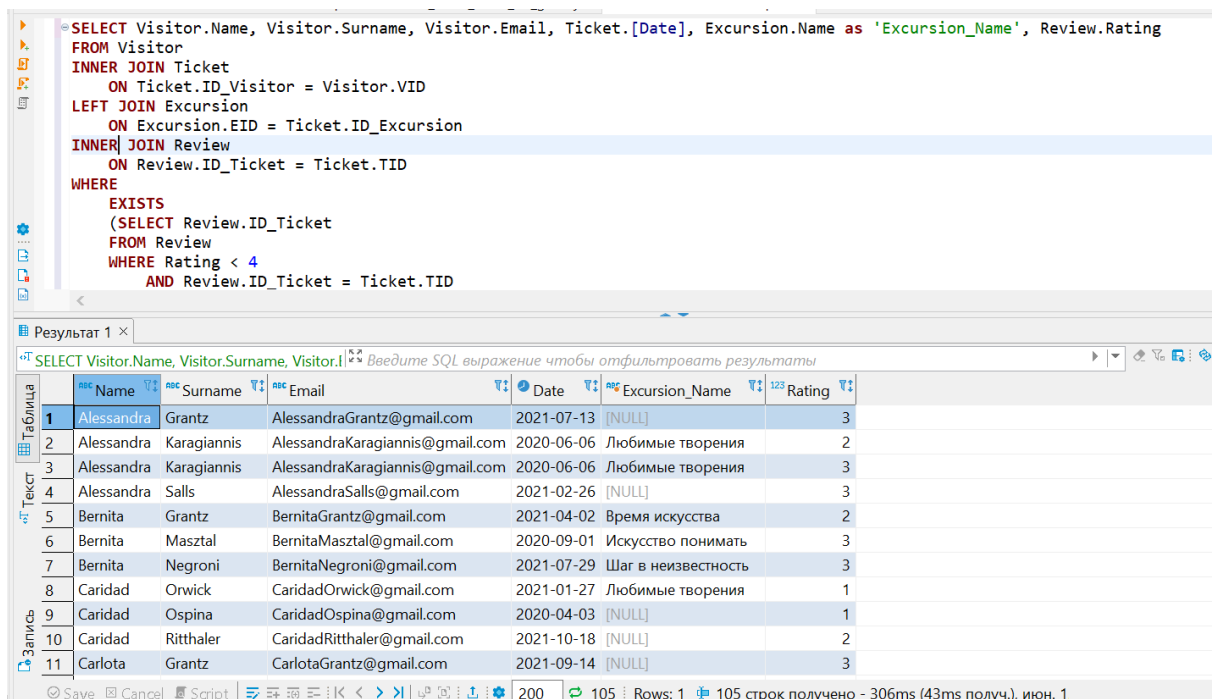
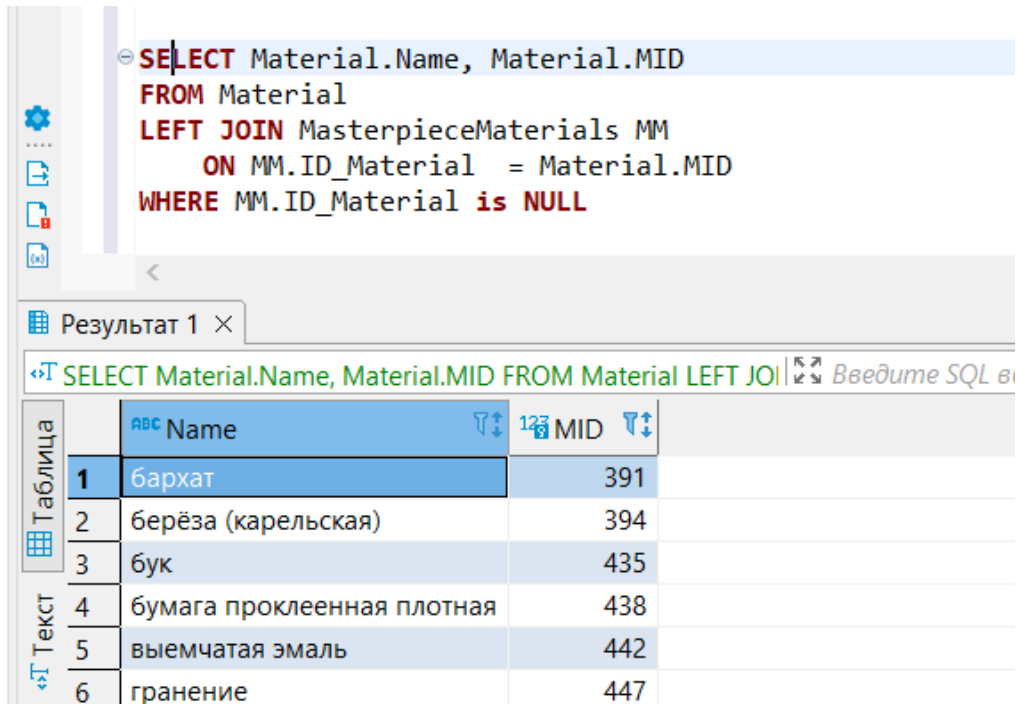


рис.32. Запрос №16

17) Запрос с внешним соединением и проверкой на наличие NULL

Выводит материалы, которые не задействованы ни в одном производстве, но есть в базе данных. Может быть полезно для редактирования или чистки базы данных.



SQL Query:

```
SELECT Material.Name, Material.MID
FROM Material
LEFT JOIN MasterpieceMaterials MM
ON MM.ID_Material = Material.MID
WHERE MM.ID_Material is NULL
```

Результат 1 ×

SQL Query:

```
SELECT Material.Name, Material.MID FROM Material LEFT JOI
```

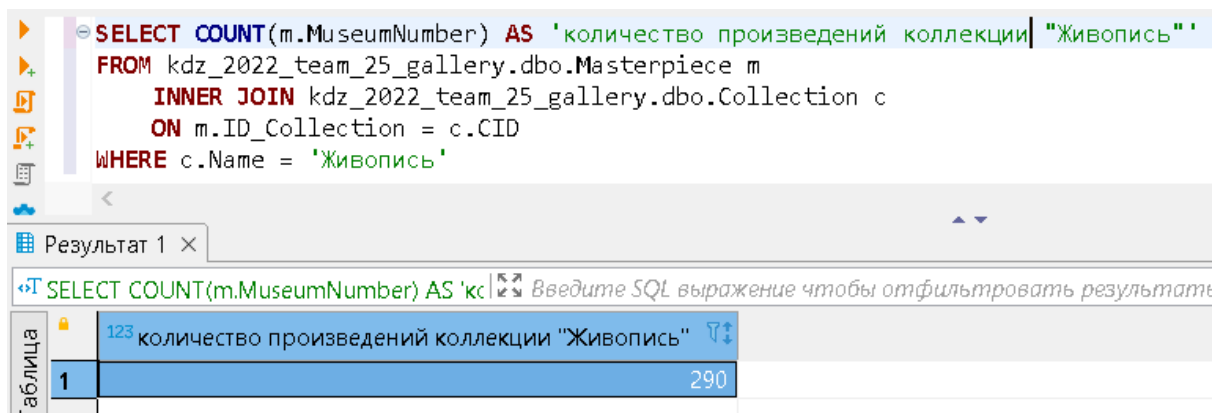
Введите SQL в

Таблица	ABC Name	123 MID
1	бархат	391
2	берёза (карельская)	394
3	бук	435
4	бумага проклеенная плотная	438
5	выемчатая эмаль	442
6	гранение	447

рис.33. Запрос №17

18) Запрос с агрегированием и выражением JOIN, включающим не менее 2 таблиц(3)

Выводит количество произведений конкретной коллекции (здесь - “Живопись”)



SQL Query:

```
SELECT COUNT(m.MuseumNumber) AS 'количество произведений коллекции "Живопись"'
FROM kdz_2022_team_25_gallery.dbo.Masterpiece m
INNER JOIN kdz_2022_team_25_gallery.dbo.Collection c
ON m.ID_Collection = c.CID
WHERE c.Name = 'Живопись'
```

Результат 1 ×

SQL Query:

```
SELECT COUNT(m.MuseumNumber) AS 'к
```

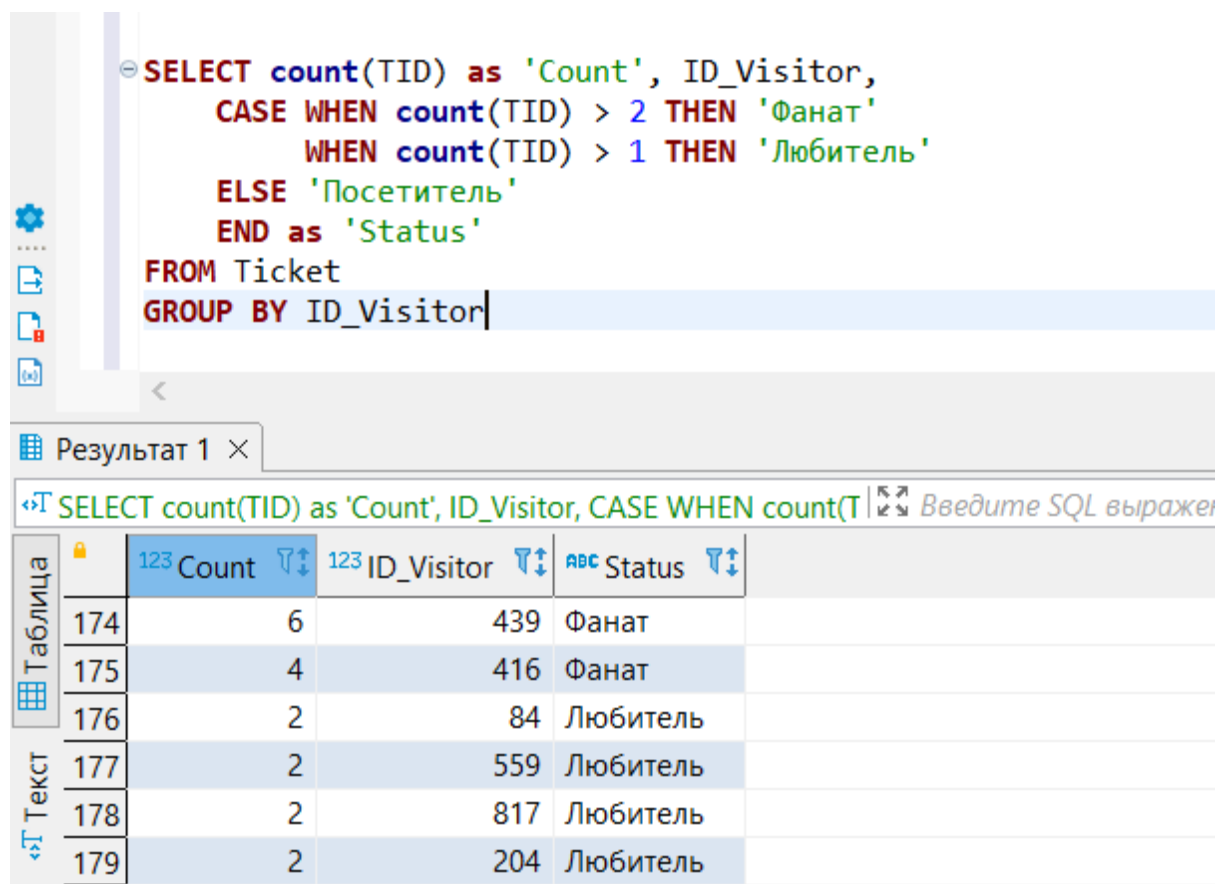
Введите SQL выражение чтобы отфильтровать результаты

Таблица	123 количество произведений коллекции "Живопись"
1	290

рис.34. Запрос №18

19) Запрос с CASE (IIF) и агрегированием

Выводит статусы посетителей в зависимости от количества посещений. Может быть полезно для создания программы лояльности музея.



```

SELECT count(TID) as 'Count', ID_Visitor,
CASE WHEN count(TID) > 2 THEN 'Фанат'
      WHEN count(TID) > 1 THEN 'Любитель'
      ELSE 'Посетитель'
END as 'Status'
FROM Ticket
GROUP BY ID_Visitor

```

Результат 1 ×

SELECT count(TID) as 'Count', ID\_Visitor, CASE WHEN count(TID) > 2 THEN 'Фанат' WHEN count(TID) > 1 THEN 'Любитель' ELSE 'Посетитель' END as 'Status' FROM Ticket GROUP BY ID\_Visitor

	Count	ID_Visitor	Status
174	6	439	Фанат
175	4	416	Фанат
176	2	84	Любитель
177	2	559	Любитель
178	2	817	Любитель
179	2	204	Любитель

рис.35. Запрос №19

## 20) Запрос с HAVING и агрегированием

Выводит экспонаты, у которых указано меньше 2 материалов. Может быть полезно для поиска пробелов в базе данных музея.





## 6.2. Триггер

Триггер NewExcursionCheck проверяет, соотносится ли с временными рамками работы музея (11:00 — 20:00) время новой экскурсии. Если нет, то вставка новой записи отменяется, и появляется сообщение “Недопустимые временные рамки для экскурсии”.

---

```
CREATE TRIGGER NewExcursionCheck
ON kdz_2022_team_25_gallery.dbo.Excursion
AFTER INSERT
AS BEGIN
    DECLARE @ExcursionEnd AS Time = DATEADD(MINUTE,
(SELECT Duration FROM inserted),(SELECT [Time] FROM inserted))
    IF (SELECT [Time] FROM inserted) < '11:00:00'
    OR @ExcursionEnd > '20:00:00'
    BEGIN
        print('Недопустимые временные рамки для экскурсии.')
        ROLLBACK
    END
END
```

---

Результат выполнения:

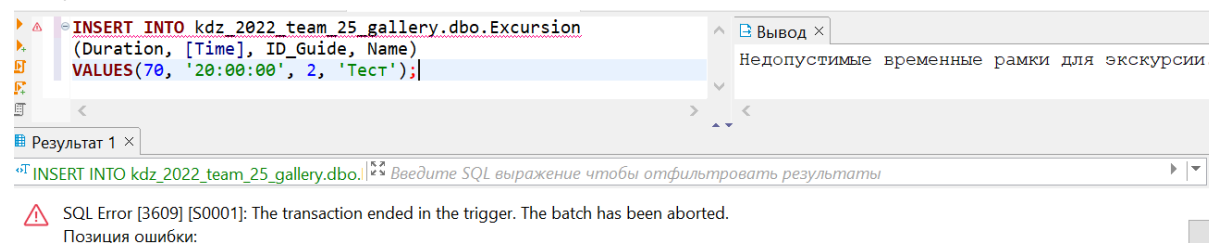


рис.38. Триггер

## 6.3. Функции

### №1

Выводит, экспонат, экскурсию, время экскурсии и экскурсовода, с которыми смотрят этот экспонат. Может быть полезно, например, реставраторам, которые хотят на время забрать картину на коррекцию. Или может быть полезно экскурсоводам, которые хотят запланировать новую экскурсию и хотят избежать столкновений с

коллегами или дублирования экскурсий. Еще один вариант использование данной функции - консультация посетителей, которым интересно, где и когда они могут послушать про какой-то экспонат.

```
CREATE FUNCTION MasterTheMasterpiece3 (@MName VARCHAR(100),
@MNumber VARCHAR(10))
RETURNS TABLE
AS RETURN
(
SELECT Masterpiece.Name, Masterpiece.MuseumNumber,
CASE WHEN Excursion.EID is NOT NULL
THEN Guide.Name + ' ' + Guide.Patronymic + ' ' +
Guide.Surname
ELSE 'Самостоятельный осмотр'
END AS 'Гид',
CASE WHEN Excursion.EID is NOT NULL
THEN Excursion.Name
ELSE 'Самостоятельный осмотр'
END as 'Название экскурсии',
CASE WHEN Excursion.EID is NOT NULL
THEN CAST(Excursion.[Time] as VARCHAR)
ELSE 'с 9:00 по 21:00 ежедневно'
END as 'Время экскурсии'
FROM Masterpiece
LEFT JOIN ExcursionContents
ON ExcursionContents.MuseumNumber =
Masterpiece.MuseumNumber
LEFT JOIN Excursion
ON ExcursionContents.ID_Excursion = Excursion.EID
LEFT JOIN Guide
ON Guide.GID = Excursion.ID_Guide
WHERE Masterpiece.Name like '%' + @MName + '%'
OR Masterpiece.MuseumNumber like @MNumber
);
```

SELECT \* FROM MasterTheMasterpiece3('девушка', 'не помню =(')

Результат 1

Введите SQL выражение чтобы отфильтровать результаты

	Name	MuseumNumber	Гид	Название экскурсии	Время экскурсии
1	Пляшущая девушка	Ж-2824	Мария Витальевна Тюльпанова	Обзор искусства 19-20 века	11:00:00
2	Девушка с кувшином.	Ск-233	Екатерина Михайловна Шульман	Время искусства	16:15:00
3	Девушка с кувшином.	Ск-233	Мария Витальевна Тюльпанова	Наедине с искусством	16:00:00
4	Девушка с корзиной.	Ск-444	Мария Витальевна Тюльпанова	Обзор искусства 19-20 века	11:00:00
5	Девушка с корзиной.	Ск-444	Мария Витальевна Тюльпанова	Наедине с искусством	16:00:00
6	Девушка.	Ск-473	Анна Андреевна Ольшанская	Введение в творения галереи	15:30:00
7	Девушка.	Ск-473	Мария Витальевна Тюльпанова	Ценителям на заметку	13:30:00
8	Девушка.	Ск-473	Мария Витальевна Тюльпанова	Наедине с искусством	16:00:00
9	Нагнувшаяся девушка.	Ск-603	Анна Андреевна Ольшанская	Творцы и их творения	17:00:00

**№2**

Функция даёт информацию о том, сколько экскурсий провел каждый экскурсовод в каком-то определенном временном периоде. Может быть использовано для аналитики по заработной плате или количеству отработанных часов-экскурсий по каждому сотруднику.

---

```

CREATE FUNCTION dbo.CountExcursionsGuides
(@startDate NVARCHAR(10), @finishDate NVARCHAR(10))
RETURNS TABLE
AS RETURN
(
SELECT  tableNew.GID,    SUM(tableNew.numExc)    AS    'количество
проведённых экскурсий'
FROM
(
SELECT g.GID, t.[Date], COUNT(DISTINCT e.EID) AS 'numExc'
FROM kdz_2022_team_25_gallery.dbo.Ticket t
      INNER JOIN kdz_2022_team_25_gallery.dbo.Excursion e
      ON t.ID_Excursion = e.EID
      INNER JOIN kdz_2022_team_25_gallery.dbo.Guide g
      ON e.ID_Guide = g.GID
WHERE t.ID_Excursion IS NOT NULL AND (t.[Date] >= @startDate AND
t.[Date] <= @finishDate)
GROUP BY g.GID, t.[Date]
) AS tableNew
GROUP BY tableNew.GID
)

```

---

пример выполнения:

SELECT \*

FROM dbo.CountExcursionsGuides('2021-10-25', '2021-12-02')

Результат 1

SELECT \* FROM dbo.CountExcursionsGuides('2021-10-25', '2021-12-02')

ID	GUID	количество проведённых экскурсий
1	2	2
2	3	2
3	4	3
4	5	5
5	6	3

рис.40. Функция №2

## 6.6.Представления

Первое: Статистика(среднее и медиана) по экскурсоводам. Может быть полезно руководству музея, чтобы оценить эффективность работы определенных сотрудников, а также сравнить оценку Свободных посещений и экскурсий. Например, из данного представления мы видим, что Мария Тюльпанова нравится посетителям больше, чем Свободные посещения, а Сергей Винидиктов - меньше. Также у Марии Тюльпановой и Анны Олышанской самые высокие средние и медианные рейтинги, возможно им стоит выделить премию

```

CREATE VIEW GuideRating
as
SELECT
distinct AVG(Review.Rating) OVER (PARTITION BY Guide.Surname) AS 'Средний рейтинг',
PERCENTILE_DISC(0.5) WITHIN GROUP (ORDER BY Review.Rating)
OVER (PARTITION BY Guide.Surname) AS 'Медианный рейтинг',
CASE WHEN Guide.GID is NULL
THEN 'Свободное посещение' ELSE Guide.Name
END AS 'Имя',
CASE WHEN Guide.GID is NULL
THEN 'Свободное посещение' ELSE Guide.Surname
END AS 'Фамилия',
CASE WHEN Guide.GID is NULL
THEN 'Свободное посещение' ELSE Guide.Patronymic
END AS 'Отчество'
FROM Review
INNER JOIN Ticket
ON Ticket.TID = Review.ID_Ticket
LEFT JOIN Excursion
ON Excursion.EID = Ticket.ID_Excursion
LEFT JOIN Guide
ON Guide.GID = Excursion.ID_Guide;

SELECT * FROM GuideRating

```

	Средний рейтинг	Медианный рейтинг	Имя	Фамилия	Отчество
1	5	5	Сергей	Винидиктов	Валентинович
2	5	6	Андрей	Перельман	Давидович
3	5	6	Екатерина	Шульман	Михайловна
4	5	6	Свободное посещение	Свободное посещение	Свободное посещение
5	6	7	Анна	Ольшанская	Андреевна
6	6	7	Мария	Тюльпанова	Витальевна

рис.41. Представление №1

Второе: Выводит статистику по рейтингу, выручке и посещениям экскурсий. Может быть полезно для ценообразования и распределения экскурсоводов на экскурсии. Например, мы видим, что экскурсия “Творцы и их творения” является одной из самых популярных, согласно рейтингам, но у нее сравнительно высокая средняя выручка и мало посещений. Возможно, стоит немного понизить цену, тогда суммарная выручка и посещения будут расти.

```

CREATE VIEW ExcursionRating
as
SELECT
AVG(Review.Rating) as 'Рейтинг',
SUM(TicketType.Price) as 'Суммарная выручка',
count(Ticket.TID) as 'Посещения',
AVG(TicketType.Price) as 'Средняя выручка',
CASE WHEN Excursion.EID is NULL
THEN 'Свободное посещение' ELSE Excursion.Name
END AS 'Название экскурсии'
FROM Review
INNER JOIN Ticket
ON Ticket.TID = Review.ID_Ticket
INNER JOIN TicketType
ON TicketType.TTID = Ticket.ID_TicketType
LEFT JOIN Excursion
ON Excursion.EID = Ticket.ID_Excursion
GROUP BY Excursion.EID, Excursion.Name;

SELECT * FROM ExcursionRating

```

рис.42. Представление №2 часть 1

Для раскраски использовалась встроенная функция DBeaver

«T SELECT \* FROM ExcursionRating Введите SQL выражение чтобы отфильтровать результаты

	123 Рейтинг	123 Суммарная выручка	123 Посещения	123 Средняя выручка	abc Название экскурсии
1	7	2 750	5	550	Творцы и их творения
2	7	4 350	15	290	Час с мастером
3	6	4 400	12	366	Прогулка в прошлое
4	6	6 050	13	465	Шаг в неизвестность
5	6	1 850	6	308	Обзор искусства 19-20 века
6	6	6 700	14	478	Ценителям на заметку
7	6	4 150	11	377	Наедине с искусством
8	6	3 450	12	287	Введение в творения галереи
9	5	7 200	15	480	Искусство -- в народ
10	5	40 000	178	224	Свободное посещение
11	5	4 350	15	290	Работы великих
12	5	3 700	6	616	Искусство понимать
13	5	4 800	18	266	Любимые творения
14	5	3 950	7	564	Художественная пауза
15	4	3 450	12	287	Время искусства
16	4	5 050	11	459	Арт-терапия

рис.43. Представление №2 часть 2

## 6.6.Хранимые процедуры

### Процедура №1

Одна из процедур должна включать в себя обработку ошибок TRY CATCH, а также работу с транзакциями (begin transaction, rollback transaction, commit transaction)

Процедура добавляет произведение в таблицу, но если автора/коллекции не существует, то автор/коллекция сначала добавляются в соответствующие таблицы, а потом уже добавляется произведение.

---

```

CREATE PROCEDURE dbo.InsertMasterpiece
@masterpieceID nvarchar(100),
@masterpieceName nvarchar(300),
@year nvarchar(100),
@country nvarchar(100),
@authorSurname nvarchar(100),
@authorName nvarchar(100),
@collection nvarchar(100)
AS
BEGIN

```

```

BEGIN TRY
    DECLARE @idAuthor int
    DECLARE @idCollection int
    SET @idAuthor = (SELECT a.AID
    FROM kdz_2022_team_25_gallery.dbo.Author a
    WHERE a.Surname = @authorSurname AND (a.Name = @authorName
    OR a.Name = NULL OR a.Name = ''))
    SET @idCollection = (SELECT c.CID
    FROM kdz_2022_team_25_gallery.dbo.Collection c
    WHERE c.Name = @collection)
    BEGIN TRANSACTION
    IF (@idAuthor IS NULL)
    BEGIN
        INSERT INTO
        kdz_2022_team_25_gallery.dbo.Author(Surname, Name)
        VALUES(@authorSurname, @authorName)
        SET @idAuthor = (SELECT IDENT_CURRENT ('Author'))
        PRINT 'Автор добавлен'
        END
    IF (@idCollection IS NULL)
    BEGIN
        INSERT INTO kdz_2022_team_25_gallery.dbo.Collection(Name)
        VALUES(@collection)
        SET @idCollection = (SELECT IDENT_CURRENT ('Collection'))
        PRINT 'Коллекция добавлена'
    END
    INSERT INTO
    kdz_2022_team_25_gallery.dbo.Masterpiece(MuseumNumber, Name, [Year],
    Country, ID_Author, ID_Collection)
    VALUES (@masterpieceID, @masterpieceName, @year, @country,
    @idAuthor, @idCollection)
    PRINT 'Произведение добавлено в таблицу'
    COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
    PRINT 'Невозможно добавить произведение по входным
    параметрам'
    ROLLBACK TRANSACTION
    END CATCH
END

```

---

пример 1: добавления произведения с новым автором и коллекцией.



EXEC dbo.InsertMasterpiece 'Ж-1123456', 'Проверка 4', NULL, 'Россия', 'Иван', 'Иванов', 'СуперВек'

Статистика 1 x

EXEC dbo.InsertMasterpiece 'Ж-1123456', 'Проверка 4', NULL, 'Россия', 'Иван', 'Иванов', 'СуперВек'

Name	Value
Updated Rows	3
Query	EXEC dbo.InsertMasterpiece 'Ж-1123456', 'Проверка 4', NULL, 'Россия', 'Иван', 'Иванов', 'СуперВек'
Finish time	Sat Jun 11 21:45:38 MSK 2022

2	Ж-1123456	Меркурий	1580 (1564-1585 ?)	Италия	174	839
3	Ж-1123456	Проверка 4	NULL	Россия	845	848
4	Ж-1214	Обыкновенный ретинице	1785-1792	Франция	627	845

8	845	Живопись
9	848	СуперВек

Запись	391	794	Гоген	Поль
	392	803	Иван	Иванов

рис.44. Процедура №1 часть 1

Пример 2: добавление произведения, музейный номер которого уже есть в базе данных.

EXEC dbo.InsertMasterpiece 'Ж-1226', 'Проверка 5', NULL, 'Россия', 'Иван', 'Иванов', 'СуперВек'

Статистика 1 x

EXEC dbo.InsertMasterpiece 'Ж-1226', 'Проверка 5', NULL, 'Россия', 'Иван', 'Иванов', 'СуперВек'

Name	Value
Updated Rows	0
Query	EXEC dbo.InsertMasterpiece 'Ж-1226', 'Проверка 5', NULL, 'Россия', 'Иван', 'Иванов', 'СуперВек'
Finish time	Sat Jun 11 21:50:20 MSK 2022

рис.45. Процедура №1 часть 2

## Процедура №2

Процедура, которая добавляет нового экскурсовода

```
CREATE PROCEDURE dbo.AddGuide
@surname NVARCHAR(50), @name NVARCHAR(50), @patronymic
NVARCHAR(50), @phd NVARCHAR(50)
AS
INSERT INTO kdz_2022_team_25_gallery.dbo.Guide(Surname, Name, Patronymic,
Phd)
VALUES (@surname, @name, @patronymic, @phd)
```

---

пример выполнения:

The screenshot shows the SQL Server Enterprise Manager interface. At the top, a query window displays the execution of the stored procedure `EXEC dbo.AddGuide 'Дудь', 'Юрий', 'Александрович', NULL`. Below the query window, a statistics window shows the execution details: 1 row updated, query text, and finish time. At the bottom, a table window displays the data from the `Guide` table.

Name	Value
Updated Rows	1
Query	EXEC dbo.AddGuide 'Дудь', 'Юрий', 'Александрович', NULL
Finish time	Sat Jun 11 15:42:56 MSK 2022

	GID	Surname	Name	Patronymic	Phd
1	2	Ольшанская	Анна	Андреевна	[NULL]
2	3	Винидиктов	Сергей	Валентинович	[NULL]
3	4	Шульман	Екатерина	Михайловна	[NULL]
4	5	Перельман	Андрей	Давидович	[NULL]
5	6	Тюльпанова	Мария	Витальевна	[NULL]
6	7	Дудь	Юрий	Александрович	[NULL]

рис.46. Процедура №2

### Процедура №3

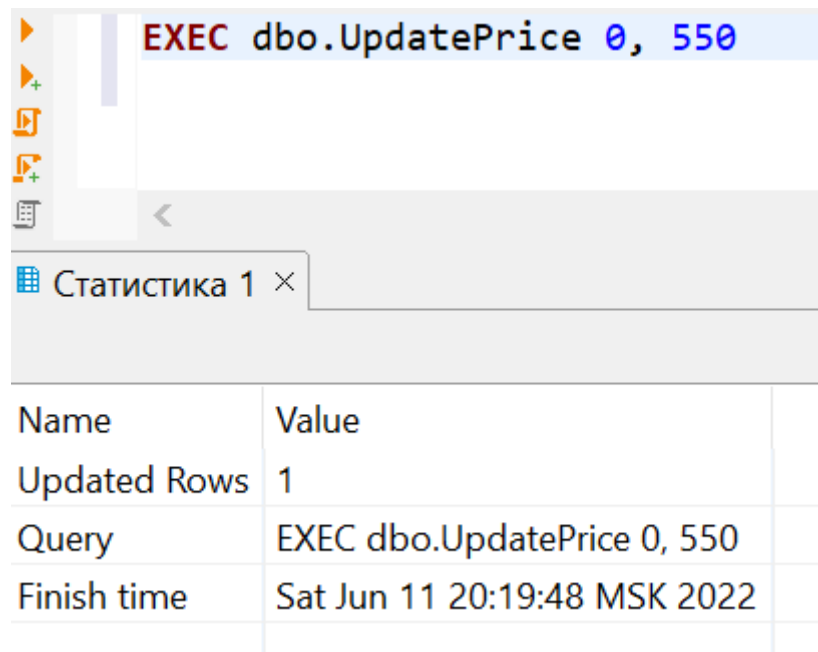
Процедура изменяет цену билета

---

```
CREATE PROCEDURE dbo.UpdatePrice
@TicketTypeID int, @NewPrice int
AS BEGIN
    UPDATE kdz_2022_team_25_gallery.dbo.TicketType
    SET Price = @NewPrice
    WHERE TTID = @TicketTypeID;
END
```

---

Пример выполнения:



The screenshot shows a SQL Server interface. At the top, a command window displays the text `EXEC dbo.UpdatePrice 0, 550`. Below it, a tab labeled 'Статистика 1' (Statistics 1) is open, showing a table of execution statistics.

Name	Value
Updated Rows	1
Query	EXEC dbo.UpdatePrice 0, 550
Finish time	Sat Jun 11 20:19:48 MSK 2022

рис.47. Процедура №3

#### Процедура №4

Одна из процедур должна использовать ветвление и содержать не менее 3 запросов;

Процедура для добавления информации(название, год, дата создания) по какому-то произведению. Если информации по какому-то свойству нет, то происходит добавление, если же есть - выводится предупреждение, что информация уже есть. Может быть полезна для актуализации базы данных, в которой присутствуют пробелы.

---

```

CREATE PROCEDURE UpdateMasterpieceEd @Number VARCHAR(10),
@NewName VARCHAR(100), @NewCountry VARCHAR(10), @NewYear
VARCHAR(10)
AS BEGIN
    DECLARE @CurName VARCHAR(100), @CurCountry VARCHAR(10),
@CurYear VARCHAR(10)
    SET @CurName =
        (SELECT Masterpiece.Name
        FROM Masterpiece
        WHERE Masterpiece.MuseumNumber = @Number)
    SET @CurCountry =
        (SELECT Masterpiece.Country
        FROM Masterpiece
        WHERE Masterpiece.MuseumNumber = @Number)

```

```

        SET @CurYear =
            (SELECT Masterpiece.Year
             FROM Masterpiece
             WHERE Masterpiece.MuseumNumber = @Number)
    IF @NewName is not NULL
    BEGIN
        IF (LEN(@CurName) < 1)
        BEGIN
            UPDATE Masterpiece
            SET Masterpiece.Name = @NewName
            WHERE Masterpiece.MuseumNumber = @Number
        END
        ELSE PRINT 'У данного экспоната уже есть название'
    END

    IF @NewCountry is not NULL
    BEGIN
        IF (@CurCountry is NULL OR LEN(@CurCountry) < 1)
        BEGIN
            UPDATE Masterpiece
            SET Masterpiece.Country = @NewCountry
            WHERE Masterpiece.MuseumNumber = @Number
        END
        ELSE PRINT 'У данного экспоната уже есть страна'
    END

    IF @NewYear is not NULL
    BEGIN
        IF (@CurYear is NULL OR LEN(@CurYear) < 1)
        BEGIN
            UPDATE Masterpiece
            SET Masterpiece.Year = @NewYear
            WHERE Masterpiece.MuseumNumber = @Number
        END
        ELSE PRINT 'У данного экспоната уже указан год'
    END
END;

```

---

Например, попробуем добавить год и страну в экспонат ОИ-34.

Masterpiece | Введите SQL выражение чтобы отфильтровать результаты

	MuseumNumber	Name	Year	Country	ID_Author	ID_Collection
367	OИ-29	План Диснейленда	1992		192	840
368	OИ-3	Радикальная композиция	1982-1984		185	840
369	OИ-30	Медведь и Мэрилин Монро			201	840
370	OИ-31	Сталин и Мэрилин Монро			201	840
371	OИ-32	Свобода			201	840
372	OИ-33	Три женщины в черном.	1970-е гг.		204	840
373	OИ-34	Две девушки в троллейбусе.	1970-е гг.		204	840
374	OИ-35	Воспоминание III	1981		182	840
375	OИ-36	Воспоминание IV	1981		182	840
376	OИ-37	Из серии «Биология шедевров»	2016		206	840
377	OИ-38	Гнездо для Москвы #1	2018		189	840
378	OИ-39	Гнездо для Москвы #2	2018		189	840

рис.48. Процедура №4 часть 1

Как мы видим, у экспоната есть год, но нет страны. Процедура добавит страну, а для года выведет предупреждение и никаких изменений не внесет.

```

IF @NewYear is not NULL
BEGIN
    IF (@CurYear is NULL OR LEN(@CurYear) < 1)
    BEGIN
        UPDATE Masterpiece
        SET Masterpiece.Year = @NewYear
        WHERE Masterpiece.MuseumNumber = @Number
    END
    ELSE PRINT 'У данного экспоната уже указан год'
END
END;

EXEC UpdateMasterpieceEd 'OИ-34', NULL, 'Россия', '2022';

GO

SELECT * FROM HumanResources.vEmployeeDepartment;

```

У данного экспоната уже указан год

Name	Value
Updated Rows	1
Query	EXEC UpdateMasterpieceEd 'OИ-34', NULL, 'Россия', '2022'
Finish time	Sun Jun 12 16:20:53 MSK 2022

Masterpiece | Введите SQL выражение чтобы отфильтровать результаты

	MuseumNumber	Name	Year	Country	ID_Author	ID_Collection
367	OИ-29	План Диснейленда	1992		192	840
368	OИ-3	Радикальная композиция	1982-1984		185	840
369	OИ-30	Медведь и Мэрилин Монро			201	840
370	OИ-31	Сталин и Мэрилин Монро			201	840
371	OИ-32	Свобода			201	840
372	OИ-33	Три женщины в черном.	1970-е гг.		204	840
373	OИ-34	Две девушки в троллейбусе.	1970-е гг.	Россия	204	840
374	OИ-35	Воспоминание III	1981		182	840
375	OИ-36	Воспоминание IV	1981		182	840
376	OИ-37	Из серии «Биология шедевров»	2016		206	840

рис.49. Процедура №4 часть 2

## 6.7.Настройка индексов

Скорость запроса до добавления индексов - 157 ms:

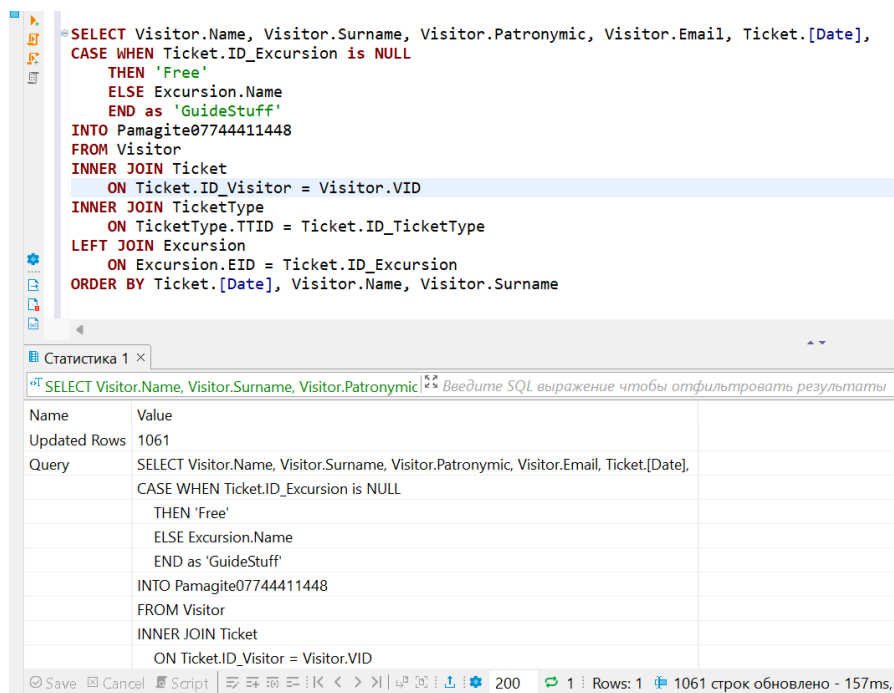


рис.50. Скорость до индексов

Добавили индексы Ticket.ID\_Visitor, Visitor.VID

Скорость увеличилась в 4 раза, сократив время выполнения до 40ms:

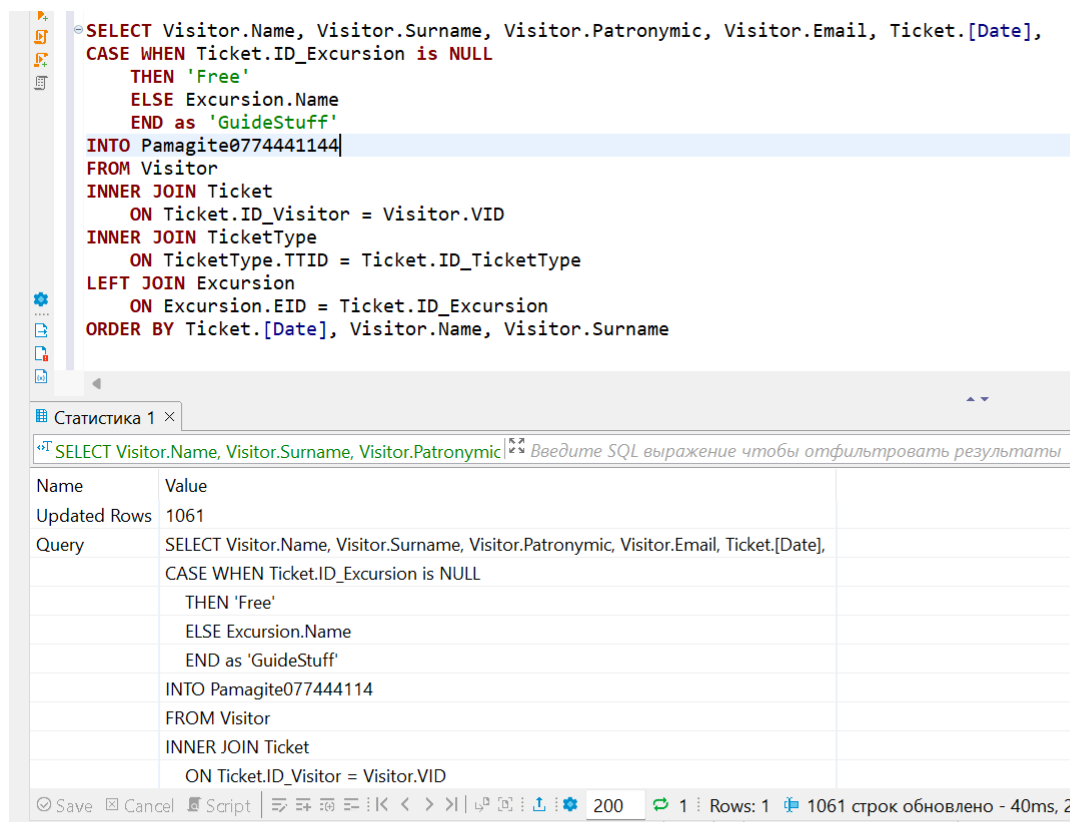


рис.51. Скорость после добавления индексов

## 7. Отчетные формы

### 7.1. Основной отчёт Tableau (Power BI) – агрегированные данные со ссылками на детализацию

Данный отчёт содержит в себе аналитику по 2021 года. Данные графики служат для анализа полученной выручки и общему количеству посещений. Можно сделать вывод о том, что к концу календарного года выручка падает, хотя в июне и октябре виден рост.

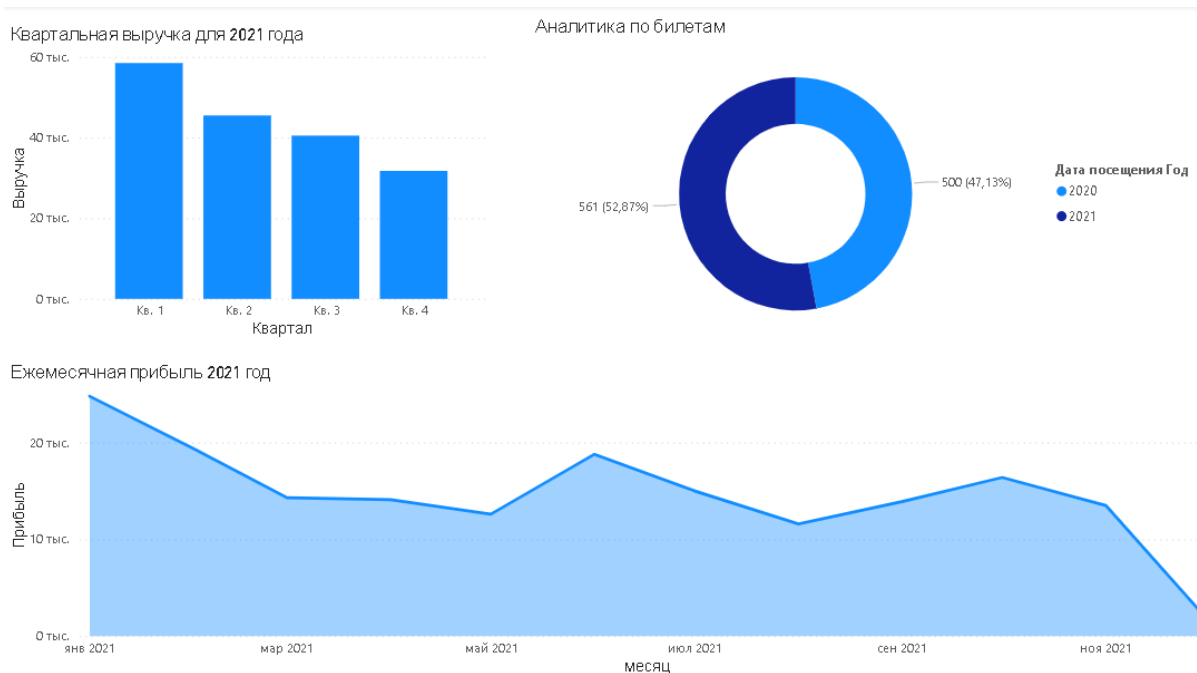


рис.52. Основной отчёт Power BI

### 7.2. Детальный отчёт Tableau (Power BI) для drill down

Детальный отчёт по количеству посещений галереи для 2021 года позволяет увидеть аналитику по году/кварталу/месяцу. Этот анализ помогает понять наиболее загруженные месяцы для музея для дальнейшего прогнозирования занятости персонала на будущий год.

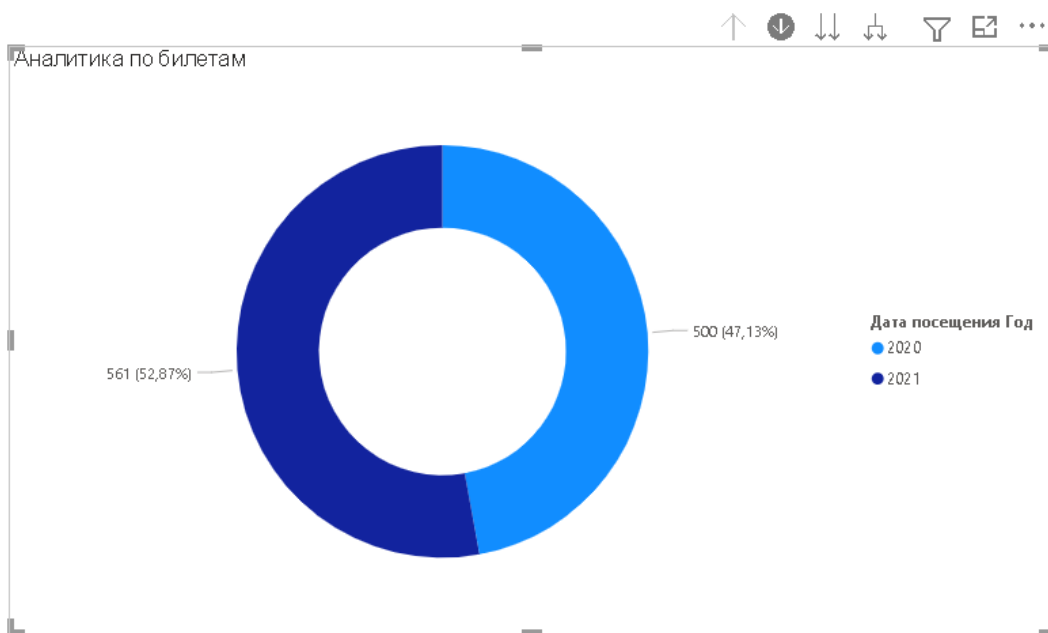


рис.53. Детальный отчет Power BI часть 1  
для 2021 года:

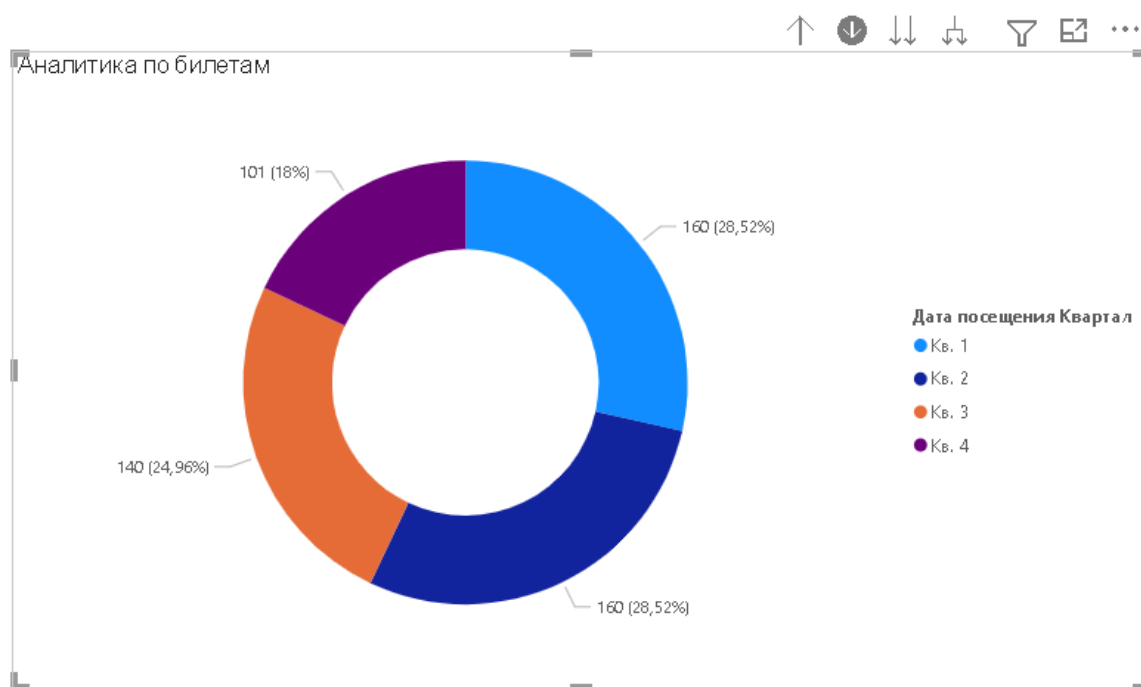


рис.54. Детальный отчет Power BI часть 2  
для 2 квартала:



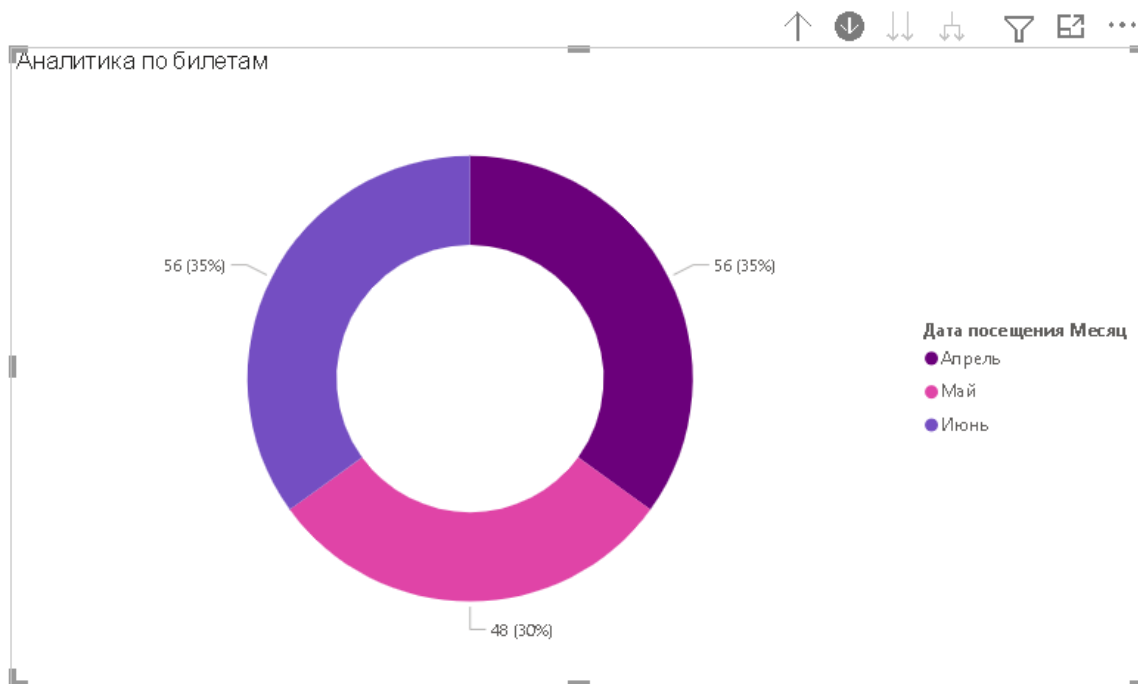


рис.55. Детальный отчет Power BI часть 3

### 7.3. Информационная панель в Excel с графиком и "срезами"

В качестве основы для данной таблицы мы взяли представление GuideRating. Оно позволяет вывести два типа рейтинга — средний и медианный, позволяющие оценить, насколько близка к медиане средняя оценка, то есть сделать примерную поправку на выбросы.

В качестве графика мы выбрали гистограмму с группировкой, позволяющую визуально оценить разницу между средним и медианным рейтингом. Срез по фамилии экскурсовода позволяет оставить только данные о выбранном экскурсоводе.

Этот отчет позволяет проанализировать успешность экскурсовода, а также может послужить основой для принятия решения о повышении зарплаты или выплаты премии.

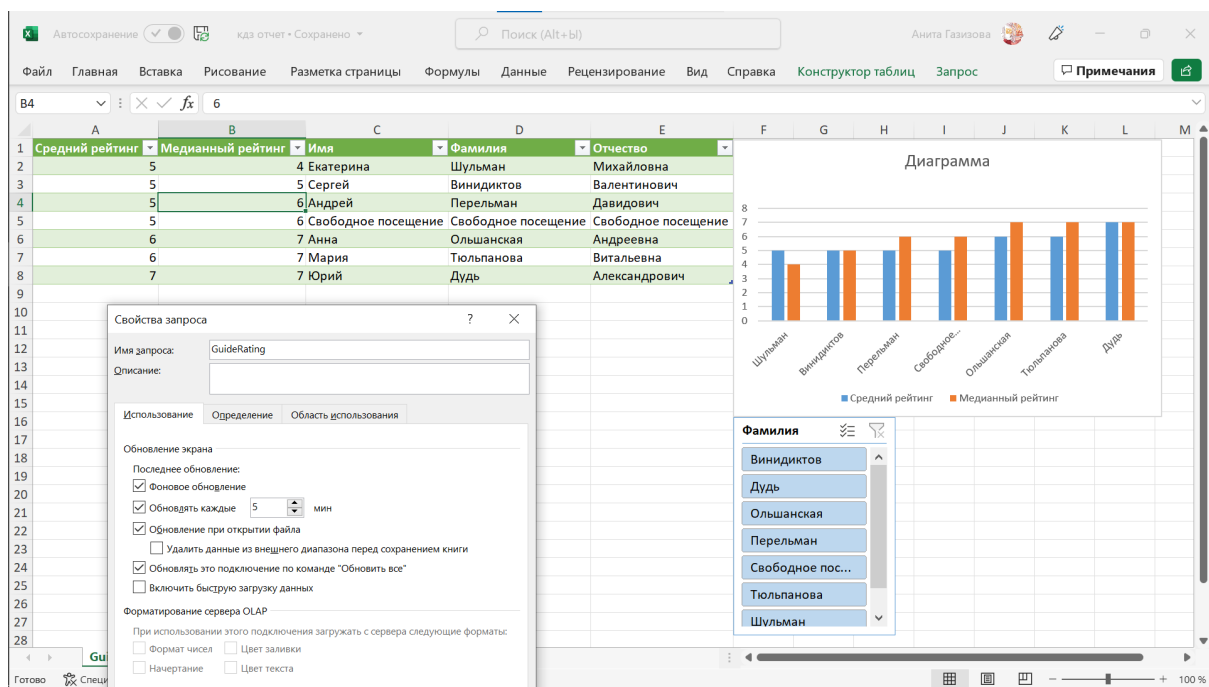


рис.56. Информационная панель в Excel с графиком и "срезами"