NATIONAL RESEARCH UNIVERSITY

HIGHER SCHOOL OF ECONOMICS

SCHOOL OF BUSINESS INFORMATICS

Контрольное домашнее задание

Разработка базы данных по информации   
о научных журналах и их цитированиях

Казачинская Анастасия,

Семина Анна,

Сторожок Мария,

Чургель Анастасия

MOSCOW

2020

Оглавление

[Постановка задачи 4](#_Toc74602402)

[Предметная область 4](#_Toc74602403)

[Цель создания БД 4](#_Toc74602404)

[Возможные пользователи 4](#_Toc74602405)

[Сценарии взаимодействия 4](#_Toc74602406)

[Описание предметной области 5](#_Toc74602407)

[Проектирование базы данных 6](#_Toc74602408)

[Инфологическая модель предметной области 6](#_Toc74602409)

[Описание сущностей предметной области 7](#_Toc74602410)

[Связи между сущностями предметной области 8](#_Toc74602411)

[Даталогическая модель данных 8](#_Toc74602412)

[Реализация базы данных в СУБД 10](#_Toc74602413)

[Хранимые процедуры 11](#_Toc74602414)

[Первая хранимая процедура (рис.4): 11](#_Toc74602415)

[Вторая хранимая процедура (рис.5): 11](#_Toc74602416)

[Третья хранимая процедура: 12](#_Toc74602417)

[Четвертая хранимая процедура (рис. 11): 13](#_Toc74602418)

[Пятая хранимая процедура (рис. 12): 14](#_Toc74602419)

[Функции 15](#_Toc74602420)

[Первая функция (рис. 13): 15](#_Toc74602421)

[Вторая функция (рис. 14): 15](#_Toc74602422)

[Третья пользовательская функция (рис. 15): 16](#_Toc74602423)

[Представления 17](#_Toc74602424)

[Первое представление (рис. 16): 17](#_Toc74602425)

[Второе представление (рис. 17): 17](#_Toc74602426)

[Запросы 19](#_Toc74602427)

[Простой запрос с условием и формулами в SELECT 19](#_Toc74602428)

[Запрос с коррелированным подзапросом в SELECT 19](#_Toc74602429)

[Запрос с подзапросом в FROM 20](#_Toc74602430)

[Запрос с подзапросом в FROM, агрегированием, группировкой и сортировкой 20](#_Toc74602431)

[Запрос с коррелированным подзапросом в WHERE 21](#_Toc74602432)

[Запрос, использующий оконную функцию LAG или LEAD для выполнения сравнения данных в разных периодах 21](#_Toc74602433)

[Запрос с агрегированием и выражением JOIN, включающим не менее 2 таблиц 22](#_Toc74602434)

[Запрос с EXISTS 22](#_Toc74602435)

[Запрос, использующий манипуляции с множествами 23](#_Toc74602436)

[Запрос с внешним соединением и проверкой на наличие NULL 23](#_Toc74602437)

[Запрос с агрегированием и выражением JOIN, включающим не менее 3 таблиц/выражений 23](#_Toc74602438)

[Запрос с CASE (IIF) и агрегированием 23](#_Toc74602439)

[Запрос с HAVING и агрегированием 24](#_Toc74602440)

[Запрос SELECT INTO для подготовки выгрузки 24](#_Toc74602441)

[Триггеры 26](#_Toc74602442)

# Постановка задачи

### Предметная область

Российские научные экономические журналы, издающиеся на данный момент, информация по их издательствам, авторам и статьям, а также кросс-цитирования по годам.

### Цель создания БД

Создать удобный ресурс для анализа и размещения научных журналов, который позволит проводить исследования в этой сфере и принесет коммерческую выгоду.

### Возможные пользователи

Авторы, научные журналы, ВУЗы, исследователи и исследовательские центры.

### Сценарии взаимодействия

1. Первый вариант: для размещения. Авторы /журналы/ВУЗы могут размещать свои статьи за определенную плату и иметь проценты с каждой покупки доступа к их статьям.
2. Второй вариант: для анализа. Пользователи могут купить доступ к сервису или конкретной статье для последующего прочтения.
3. Третий вариант: для долгосрочного сотрудничества. Крупные пользователи (например, ВУЗы) могут заключить договор на постоянное пополнение базы данных их журналами и статьями, а также на предоставление доступа к ресурсу на особых условиях.

# Описание предметной области

Разработанная база данных предназначена для систематизации, анализа и мгновенного доступа к информации о научных журналах, их характеристиках и кросс-цитированиях.

Использование базы данных помогает быстро получать данные, работать с ними (сортировать, анализировать и т.д.), добавлять новые и вносить правки о старых без потери информации. Актуальность нашей базы данных в том, что сейчас в научном сообществе нет подобного инструмента для работы и исследований. Единственный сервис – это РИНЦ (Российский индекс научного цитирования) и его сайт eLibrary.ru. Однако, этот сервис крайне неудобен, работает очень медленно, так как его сайт старый и сервера слабые, сотрудники очень долго отвечают на запросы предоставления платного доступа или не отвечают вовсе, а сам платный доступ оставляет желать лучшего. Поэтому мы решили сделать свою удобную базу данных, которая будет отвечать современным требованиям и уровню баз WoS и Scopus.

# Проектирование базы данных

Методика проектирования базы данных – последовательность действий, позволяющая реально существующий объект предметной области представить в машинном виде.

Таким образом, процесс проектировки состоит из следующих этапов:

* Описание предметной области (уже сделали)
* Построение инфологической модели
* Выбор СУБД и создание даталогической модели
* Проектирование физической модели

Итак, следующая стадия – инфологическое проектирование.

## Инфологическая модель предметной области

Инфологическая модель предметной области (Entity-Relationship Model, ER-модель) обеспечивает первоначальное описание информационного содержания автоматизируемых процессов (предметной области), согласовывая и объединяя в себе представления всех категорий пользователей. Далее представлена инфологическая модель нашей базы данных (рис.1). Данная модель была построена с использованием ресурса ERDPlus, основанного на методологии Питера Чена (использующего нотацию «вороньи лапки»).

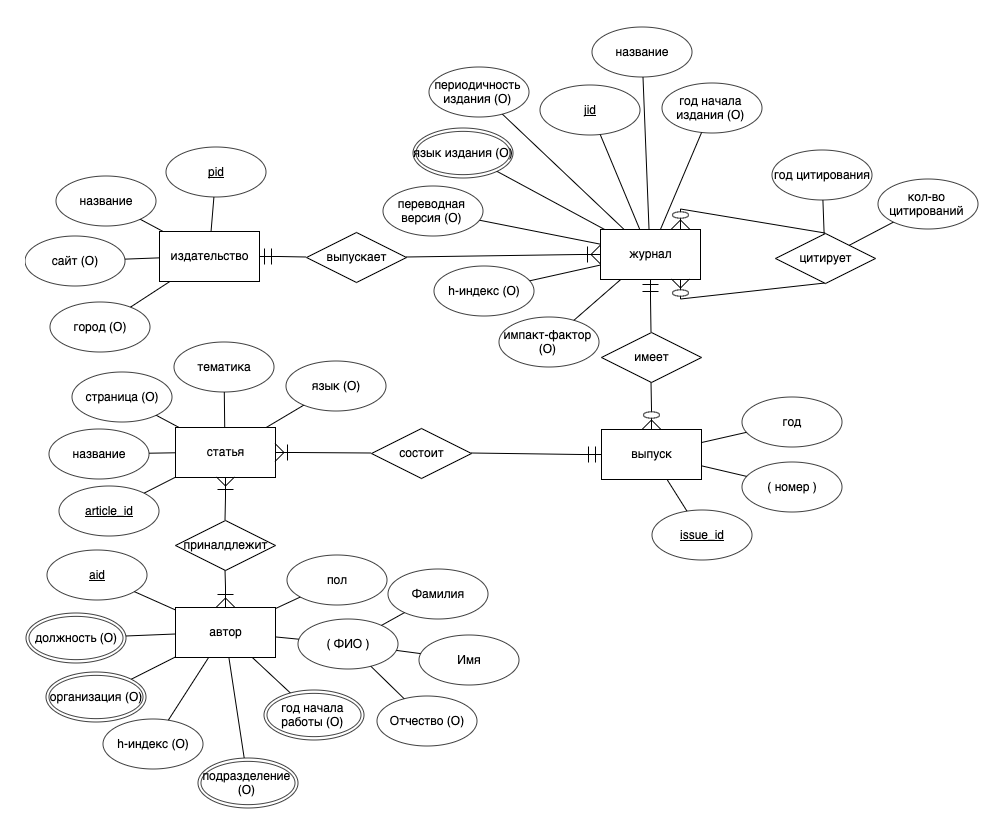


Рисунок 1. Схема инфологической модели предметной области

### Описание сущностей предметной области

Для качественного создания базы данных нам понадобились 5 сущностей.

1. **Журнал** отражает информацию о научных журналах  
   Атрибуты: уникальный код журнала(jid), название, год начала издания, периодичность издания, язык издания, наличие переводной версии, h-индекс, импакт-фактор
2. **Автор** содержит информацию об авторах журналов  
   Атрибуты: уникальный код автора, ФИО, пол, должность, организация, h-индекс, год начала работы, подразделение
3. **Издательство** показывает, каким издательствам принадлежат журналы  
   Атрибуты: уникальный код издательства, название, сайт, город
4. **Выпуск** хранит сведения о выпусках каждого журнала  
   Атрибуты: номер, месяц, год, количество статей
5. **Статья** содержит информацию о статьях в каждом выпуске журнала  
   Атрибуты: название, тематика, язык, страница

### Связи между сущностями предметной области

1. Связь **“Выпускает”** – не идентифицирующая связь между сущностями Издательство и Журнал, определяется отношением “один ко многим”, так как издательство может иметь много журналов, но журнал может принадлежать лишь одному издательству. Ни со стороны журнала, ни со стороны издательства null-значения не разрешаются, так как журнал должен кем-то выпускаться, а издательство, не выпустившее ни одного журнала, не имеет ценности нахождения в нашей базе данных.
2. Связь **“Цитирует”** – не идентифицирующая рекурсивная связь сущности Журнал, связанной с самой собой. Она описывает связь между журналами в нашей базе данных, которые цитируют друг друга и определяется отношением “многие ко многим”. Пустые значения в этой связи допускаются с обеих сторон, так как каждый журнал может и цитировать другие, и цитироваться в них как много раз, так и вовсе ни разу.
3. Связь **“Имеет”** – идентифицирующая связь между сущностями Журнал и Выпуск, определяющаяся отношением “один ко многим”, так как журнал может реализовывать много выпусков, но каждый выпуск принадлежит одному конкретному журналу. Пустые значения не разрешены.
4. Связь **“Состоит”** – не идентифицирующая связь (с суррогатным ключом вместо идентифицирующей связи с составным первичным ключом, состоящим из 3 атрибутов) между сущностями Выпуск и Статья, определяющаяся отношением “один ко многим”. В одном выпуске может содержаться много или хотя бы одна статья, а статья, в свою очередь, закреплена за конкретным выпуском, поэтому пустые значения запрещены.
5. Связь **“Принадлежит”** – не идентифицирующая связь между сущностями Статья и Автор. Определяется отношением “многие ко многим”, так как каждый автор может писать много статей и каждая статья может быть написана несколькими авторами. Пустые значения с обеих сторон запрещены, так как мы вносим автора в базу данных лишь в том случае, когда он участвовал в написании хотя бы одной статьи одного из журналов, при этом логично, что у статьи не может не быть ни одного автора.

## Даталогическая модель данных

Модель, строящаяся на даталогическом уровне проектирования базы данных, основывается на уже построенной ER-диаграмме, описывающей предметную область. Эта модель отображает данные, содержащиеся в базе данных, а также связи между этими данными. Далее показана даталоческая модель нашей базы данных (рис. 2).

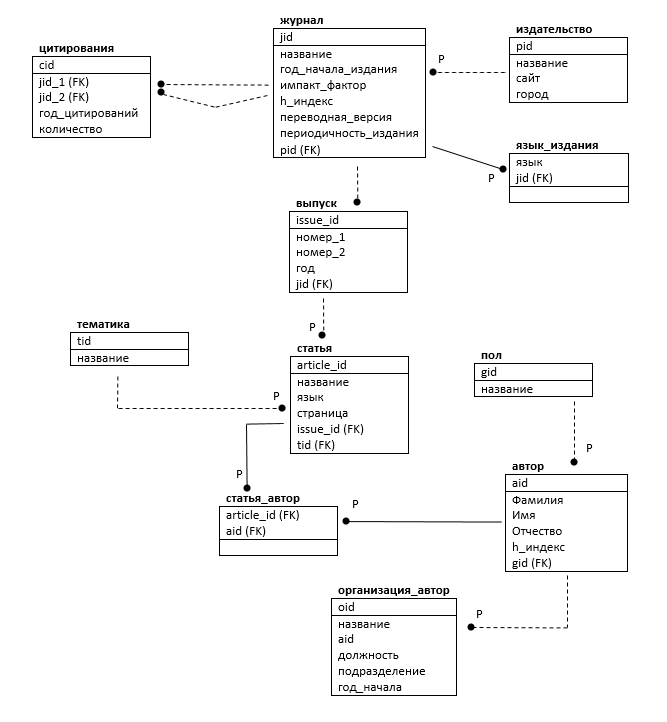


Рисунок 2. Схема даталогической модели данных

# Реализация базы данных в СУБД

Создадим базу данных научных журналов на сервере ВШЭ при помощи средства DBeaver. Она будет называться кдз\_213. Мы реализовали таблицы базы данных, заполнили их тестовыми данными.

Теперь можем посмотреть на диаграмму получившейся базы данных (рис.3):

Изображение выглядит как текст, снимок экрана, черный, серебряный

Автоматически созданное описание

Рисунок 3. Диаграмма сконструированной базы данных в СУБД

Теперь база данных реализована и готова к использованию. Покажем на примере разных запросов, чем она может быть полезна.

## Хранимые процедуры

### Первая хранимая процедура (рис.4):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4. Хранимая процедура 1

С течением времени индекс Хирша авторов, а также индекс Хирша и импакт-фактор журналов меняются. Данная хранимая процедура позволяет мгновенно внести изменения в базу данных, вызвав ее и передав ей на вход идентификатор нужного автора или журнала вместе с названиями таблицы (‘author’ либо ‘journal’, по умолчанию ‘author’) и параметра (‘h’, т.е. индекс Хирша, либо ‘if’, т.е. импакт-фактор, по умолчанию ‘h’), которые мы хотим изменить, а также новым значением параметра, который хотим обновить.

### Вторая хранимая процедура (рис.5):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5. Хранимая процедура 2

Журналы часто цитируют друг друга, и когда какой-то журнал (jid\_1 в таблице с цитированиями) использует в своем последнем выпуске отрывок одной или даже нескольких статей из другого журнала (с идентификатором цитируетjid\_2 в той же таблице), мы можем просто воспользоваться данной процедурой, которая увеличит количество цитирований журнала цитируетjid\_2 журналом jid\_1 в настоящем году на @number. Причем если до этого момента журнал не цитировался первым, в таблицу цитирований вставляется новая строка.

### Третья хранимая процедура:

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 6. Хранимая процедура 3.

Эта хранимая процедура ничего не изменяет. Она показывает, какое количество раз журнал цитировался другими за все время. Причем нас может интересовать как число его цитирований каким-то конкретным журналом, тогда мы подаем на вход название этого журнала (@jname1, так и всеми журналами сразу.

Мы можем подать процедуре на вход @jname1 = DEFAULT либо вовсе не упомянать название цитирующего журнала (не писать даже @jname1 = DEFAULT), так как, по умолчанию, этот параметр принимает значение ‘ALL’, то есть ‘ВСЕ’, на экран будет выведено общее количество цитирований журнала с поданным на вход названием @jname2 всеми остальными журналами.

Посмотрим, как работает данная процедура.

Сначала подадим на вход журнал, который точно хотя бы раз цитировался (рис.7 и 8):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7. Пример 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8. Пример 2

Теперь попробуем подать на вход несуществующий журнал, который точно никогда никем не цитировался (рис. 9):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9. Пример 3

Попробуем не указывать DEFAULT при вызове хранимой процедуры (когда хотим посмотреть на общее число цитирований всеми журналами) (рис. 10):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10. Пример 4

### Четвертая хранимая процедура (рис. 11):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11. Хранимая процедура 4

Данная процедура позволяет удалить из базы данных информацию о работе конкретного автора на конкретном месте, если по какой-либо причине нам больше неинтересно хранение данной информации.

### Пятая хранимая процедура (рис. 12):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12. Хранимая процедура 5

Данная хранимая процедура проверяет, существует ли в базе данных автор с поданным ей на вход идентификатором, и если сведения о таком авторе хранятся в таблице «автор», процедура добавляет в бд данные о его работе в заданной организации либо изменяет уже имеющиеся.

В начале процедуры явно инициализирована транзакция. Если первоначальное условие выполняется, изменения, произошедшие в ходе выполнения транзакции (вставка новой строки либо изменение имеющейся), сохраняются в базе данных при выполнении команды COMMIT. Иначе происходит отмена еще не сохраненной транзакции, за это отвечает команда ROLLBACK.

## Функции

### Первая функция (рис. 13):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13. Функция 1

Данная функция возвращает название журнала, который больше других цитирует журнал, чье название подано ей на вход. В случае, если интересующий нас журнал никем не цитирует, в том числе самим собой, функция возвращает вместо названия самого цитирующего журнала слово ‘нет’.

### Вторая функция (рис. 14):

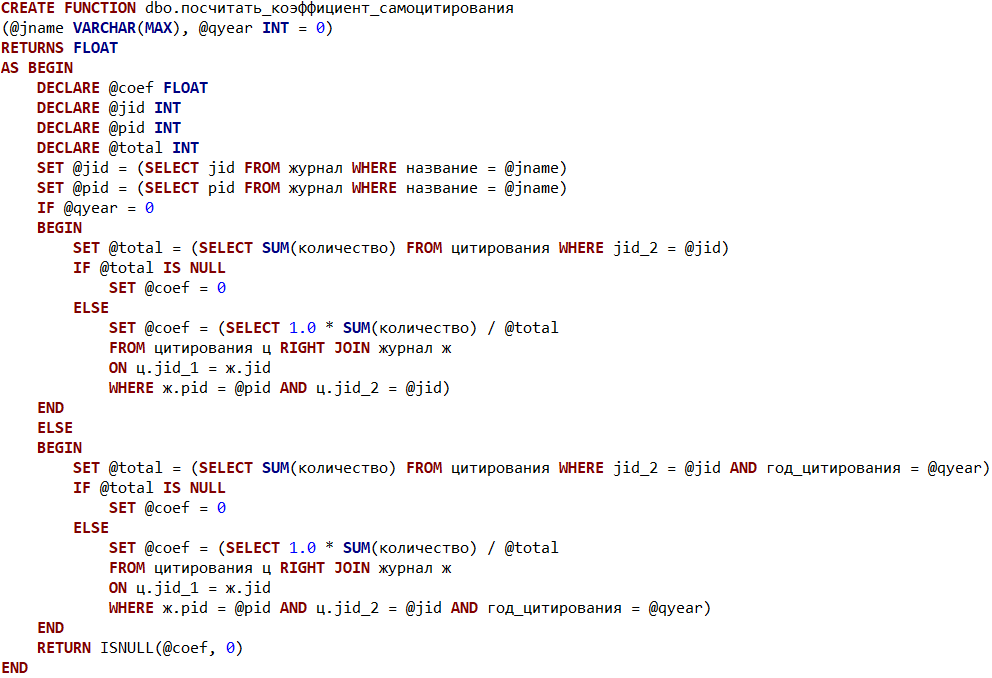


Рисунок 14. Функция 2

Данная функция, как можно понять из ее названия, вычисляет коэффициент самоцитирования для конкретного журнала. Причем, можно посчитать данный показатель как за конкретный год, так и за все время сразу. В первом случае нужно будет подать функции на вход интересующий нас год. Во втором случае нужно указать слово DEFAULT либо ничего не указывать, тогда функция вернет показатель за все годы, находящиеся в базе данных. Что касается самого коэффициента самоцитирования, он вычисляется следующим образом: число цитирований нужного журнала журналами того же издательства делится на общее число цитирований этого журнала всеми журналами (включая те случаи, когда этот журнал цитирует сам себя).

### Третья пользовательская функция (рис. 15):

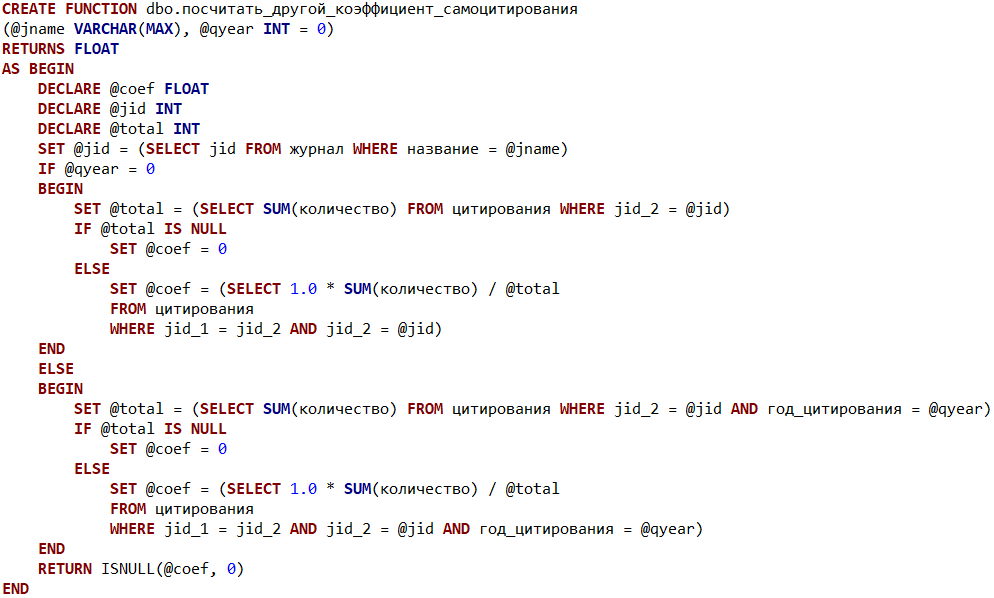


Рисунок 15. Пользовательская функция

Эта функция вычисляет другой коэффициент самоцитирования, в котором учитывается лишь цитирование журнала самим собой, без учета других журналов того же издательства. С помощью данной функции также можно рассчитать необходимый коэффициент как для конкретного года, так и по всем годам, по которым имеется информация в базе данных.

## Представления

### Первое представление (рис. 16):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 16. Представление 1

Приведенное выше представление выводит на экран основную информацию по журналам, которая часто интересна пользователям и может быть ими использована для составления какой-либо отчетности либо проведения исследовательских работ.

### Второе представление (рис. 17):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 17. Представление 2

Это представление показывает не менее важную информацию по журналам, а именно, сразу оба коэффициента самоцитирования, которые вычисляются с помощью описанных ранее функций.

С помощью данного представления мы также можем посмотреть рейтинг самоцитирования журналов, который поможет нам понять, пытаются ли журналы «обмануть» пользователей и искусственно добавить себе лишние цитирования (т.е. процитировать сами себя, если недостаточно цитирований со стороны других журналов):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 18. Пример использования представления

## Запросы

### Простой запрос с условием и формулами в SELECT

* Запрос показывает журналы, которые издаются уже 30 и более лет, и их основные показатели: импакт-фактор и h-индекс.

**SELECT** j.название, j.импакт\_фактор , j.h\_индекс , (2021 - j.год\_начала\_издания) **AS** 'Срок\_издания'

**FROM** кдз\_213.dbo.журнал j

**Where** (2021 - j.год\_начала\_издания) >= 30

**ORDER** **BY** (2021 - j.год\_начала\_издания) **DESC**

* Запрос показывает журналы, у которых присутствует "скрытое" само цитирование, то есть долю цитирования журналами того же издательства

**SELECT** c.журнал, (c.Издания\_Самоцитирования- c.Самоцитирования) **AS** "Разница"

**FROM** кдз\_213.dbo.Самоцитирование c

**WHERE** (c.Издания\_Самоцитирования-c.Самоцитирования) != 0

### Запрос с коррелированным подзапросом в SELECT

* Запрос показывает, насколько коэффициент само цитирования журналов отличается от максимального

**SELECT** с.журнал, (**SELECT** **MAX**(с.Издания\_Самоцитирования)

**FROM** кдз\_213.dbo.Самоцитирование с) - с.Издания\_Самоцитирования **AS** 'разность'

**FROM** кдз\_213.dbo.Самоцитирование с

* Запрос показывает, сколько процентов составляет количество статьей, написанные определенным автором от всего количества статей

**SELECT** a.aid, a.Фамилия, a.Имя, a.Отчество, 100\***COUNT**(sa.article\_id)/

(**SELECT** **COUNT**(sa.article\_id) **FROM** кдз\_213.dbo.статья\_автор sa ) **AS** 'процент'

**FROM** кдз\_213.dbo.автор a **join** кдз\_213.dbo.статья\_автор sa **on** sa.aid = a.aid

**join** кдз\_213.dbo.статья s **on** s.article\_id = sa.article\_id **join** кдз\_213.dbo.выпуск i **on** i.issue\_id = s.issue\_id

**GROUP** **BY** a.aid, a.Фамилия, a.Имя, a.Отчество

### Запрос с подзапросом в FROM

* Запрос показывает название журнала и его индекс Хирша, если он входит в 20-ку журналов с наибольшим импакт-фактором

**SELECT** top20.jid, top20.название, top20.h\_индекс

**FROM** (**SELECT** **TOP** 20 \* **FROM** кдз\_213.dbo.журнал j **ORDER** **BY** j.импакт\_фактор ) **as** top20

* Запрос ранжирует журналы по индексу Хирша внутри каждого города и выделяет 5 с самым высоким индексом

**SELECT** \* **FROM**

(**SELECT** i.pid, i.название **as** 'Издательство', i.город, j.jid, j.название **as** 'Журнал', j.h\_индекс ,

**ROW\_NUMBER** ()**over** (**partition** **by** i.город **order** **by** j.h\_индекс **desc**) **as** место\_журнала

**FROM** кдз\_213.dbo.журнал j **join** кдз\_213.dbo.издательство i **on** i.pid =j.pid ) **as** tb

**WHERE** tb.место\_журнала <= 5 **AND** tb.город **NOT** **LIKE** ''

### Запрос с подзапросом в FROM, агрегированием, группировкой и сортировкой

* Запрос показывает разницу между минимальным и максимальным количеством цитирований журналов в каждом году

**SELECT** tb.год\_цитирования, **MAX**(tb.Цитирования) - **MIN**(tb.Цитирования) **AS** 'Разница'

**FROM** (**SELECT** j.название ,c.год\_цитирования, **SUM**(c.количество) **AS** 'Цитирования'

**FROM** кдз\_213.dbo.журнал j **join** кдз\_213.dbo.цитирования c **on** c.jid\_2 = j.jid

**GROUP** **BY** c.год\_цитирования, j.название ) **AS** tb

**GROUP** **BY** tb.год\_цитирования

**ORDER** **BY** tb.год\_цитирования

* Запрос показывает, сколько раз журнал, который хотя бы раз цитировали, процитировал сам другой журнал. Запрос отсортирован по убыванию, от самых цитируемых журналов, к менее цитируемым

**SELECT** tb1.название, tb1.Цитировали\_журнал, tb2.Цитировал\_кого\_то

**FROM** (**SELECT** j.название , **SUM**(c.количество) **AS** 'Цитировали\_журнал'

**FROM** кдз\_213.dbo.журнал j **join** кдз\_213.dbo.цитирования c **on** c.jid\_2 = j.jid

**GROUP** **BY** j.название ) **AS** tb1

**LEFT** **join** (**SELECT** j.название , **SUM**(c.количество) **AS** 'Цитировал\_кого\_то'

**FROM** кдз\_213.dbo.журнал j **join** кдз\_213.dbo.цитирования c **on** c.jid\_1 = j.jid

**GROUP** **BY** j.название ) **AS** tb2

**on** tb1.название = tb2.название

**ORDER** **BY** tb1.Цитировали\_журнал **DESC**

### Запрос с коррелированным подзапросом в WHERE

* Запрос отбирает авторов, у которых индекс Хирша выше среднего по всем авторам

**SELECT** a.Имя, a.Фамилия, a.Отчество **FROM** кдз\_213.dbo.автор a

**Where** a.h\_индекс > (**SELECT** **AVG**(a.h\_индекс) **FROM** кдз\_213.dbo.автор a )

* Запрос отбирает журналы, у которых импакт - фактор меньше максимального импакт - фактора не более чем на 50%

**SELECT** j.jid, j.название, j.импакт\_фактор **FROM** кдз\_213.dbo.журнал j

**Where** j.импакт\_фактор >= 0.5\*(**SELECT** **MAX**(j.импакт\_фактор) **FROM** кдз\_213.dbo.журнал j )

### Запрос, использующий оконную функцию LAG или LEAD для выполнения сравнения данных в разных периодах

* Запрос показывает динамику изменений цитирований каждого журнала из года в год

**SELECT** tb.название , tb.год\_цитирования,tb.Цитирования,

(tb.Цитирования - (LAG (tb.Цитирования, 1,0) **OVER** (**PARTITION** **BY** tb.название **ORDER** **BY** tb.год\_цитирования ))) **AS** 'Динамика'

**FROM** (**SELECT** j.название ,c.год\_цитирования, **SUM**(c.количество) **AS** 'Цитирования'

**FROM** кдз\_213.dbo.журнал j **join** кдз\_213.dbo.цитирования c **on** c.jid\_2 = j.jid

**GROUP** **BY** c.год\_цитирования, j.название ) **AS** tb

### Запрос с агрегированием и выражением JOIN, включающим не менее 2 таблиц

* Запрос показывает коэффициент само цитирования не включая издательства

**SELECT** sc.jid\_2, sc.Колво\_самоцитирования, ct.Колво\_цитирования,

100\*sc.Колво\_самоцитирования/

(sc.Колво\_самоцитирования+ct.Колво\_цитирования) **AS** 'Процент самоцитирования'

**FROM** (**SELECT** c.jid\_2 ,**SUM**(количество) **as** 'Колво\_самоцитирования' **FROM** кдз\_213.dbo.цитирования c

**WHERE** c.jid\_1 =c.jid\_2 **GROUP** **BY** c.jid\_2) **AS** sc **JOIN**

(**SELECT** c.jid\_2 ,**SUM**(количество) **as** 'Колво\_цитирования' **FROM** кдз\_213.dbo.цитирования c

**WHERE** c.jid\_1 !=c.jid\_2

**GROUP** **BY** c.jid\_2) **AS** ct **on** ct.jid\_2 = sc.jid\_2

* Запрос показывает средний h-индекс авторов, которые публикуют статьи на английском языке

**SELECT** **AVG**(a.h\_индекс) **AS** 'Среднее значение'

**FROM** кдз\_213.dbo.автор a **join** кдз\_213.dbo.статья\_автор sa **on** a.aid = sa.aid

**join** кдз\_213.dbo.статья s **on** sa.article\_id = s.article\_id

**WHERE** s.язык **LIKE** 'рус'

* Запрос показывает, сколько раз в среднем цитировался журнал за каждый код

**SELECT** j.название , c.год\_цитирования, **AVG**(c.количество) **AS** 'Цитирования'

**FROM** кдз\_213.dbo.журнал j **join** кдз\_213.dbo.цитирования c **on** c.jid\_2 = j.jid

**GROUP** **BY** c.год\_цитирования, j.название

### Запрос с EXISTS

* Запрос показывает все статьи, которые имеют определенную тему и их авторов

**SELECT** s.article\_id , a.aid, s.название, a.Фамилия, a.Имя, a.Отчество

**FROM** кдз\_213.dbo.статья s **join** кдз\_213.dbo.статья\_автор sa

**on** s.article\_id = sa.article\_id

**join** кдз\_213.dbo.автор a **on** a.aid = sa.aid

**WHERE** **EXISTS** (**SELECT** s.название **FROM** кдз\_213.dbo.тематика t **WHERE** s.tid = t.tid **AND** t.название = 'Экономика' )

### Запрос, использующий манипуляции с множествами

* Запрос позволяет узнать, включает ли какой-либо журнал в свои выпуски статьи на языках, которые не заявлены как основные для журнала

**SELECT** j.название, s.язык **FROM** кдз\_213.dbo.статья s **join** кдз\_213.dbo.выпуск i **on** i.issue\_id = s.issue\_id **join** кдз\_213.dbo.журнал j **on** i.jid =j.jid

**EXCEPT**

**SELECT** j.название ,l.язык **FROM** кдз\_213.dbo.журнал j **join** кдз\_213.dbo.язык\_издания l **on** l.jid = j.jid

### Запрос с внешним соединением и проверкой на наличие NULL

* Запрос показывает есть ли журнал, который ни разу не цитировали

**SELECT** j.название **FROM** кдз\_213.dbo.журнал j **LEFT** **JOIN**

(**SELECT** **DISTINCT** c.jid\_2 **FROM** кдз\_213.dbo.цитирования c) **AS** tb

**on** tb.jid\_2 = j.jid **WHERE** ISNULL(jid\_2,1) = 1

### Запрос с агрегированием и выражением JOIN, включающим не менее 3 таблиц/выражений

* Запрос показывает каким журналам каждый автор отдает предпочтение для публикаций

**SELECT** a.Фамилия, a.Имя, a.Отчество, j.название **AS**'Журнал', **COUNT**(j.название) **AS** 'Количество'

**FROM** кдз\_213.dbo.автор a **join** кдз\_213.dbo.статья\_автор sa **on** a.aid = sa.aid

**join** кдз\_213.dbo.статья s **on** sa.article\_id = s.article\_id **join** кдз\_213.dbo.выпуск i **on** i.issue\_id = s.issue\_id

**join** кдз\_213.dbo.журнал j **on** j.jid = i.jid

**GROUP** **BY** a.Фамилия, a.Имя, a.Отчество, j.название

### Запрос с CASE (IIF) и агрегированием

* Запрос добавляет атрибут, оценивающий рост цитирований журнала за 4 года, если разница в количестве цитирований меньше нуля, считаем, что оно уменьшилось, если от 0 до 10, то не изменилось, так как рост в 10 статьей не является значительным, и если больше 10, то выросло.

**SELECT** tb1.название, tb1.разница,

**CASE**

**WHEN** разница> 10 **THEN** 'выросло'

**WHEN** разница < 0 **THEN** 'уменьшилось'

**WHEN** разница< 10 **THEN** 'не изменилось'

**END** **AS** 'изменение'

**FROM**

(**SELECT** tb.название, tb.год\_цитирования,

tb.Количество - LAG (tb.Количество, 1,0) **OVER** (**PARTITION** **BY** tb.название **ORDER** **BY** tb.год\_цитирования ) **AS** 'разница'

**FROM**(

**SELECT** j.название, c.год\_цитирования, **SUM**(c.количество) **AS** 'Количество'

**FROM** кдз\_213.dbo.цитирования c **join** кдз\_213.dbo.журнал j **on** j.jid = c.jid\_2

**WHERE** c.год\_цитирования = 2016 **OR** c.год\_цитирования = 2020

**GROUP** **BY** c.год\_цитирования, j.название ) **AS** tb) **AS** tb1

**WHERE** tb1.год\_цитирования = 2020

### Запрос с HAVING и агрегированием

* Запрос показывает авторов, которые уже выпустили хотя бы 1 статью в этом году

**SELECT** a.Фамилия, a.Имя, a.Отчество, **COUNT**(sa.article\_id) **AS** 'Количество статей'

**FROM** кдз\_213.dbo.автор a **join** кдз\_213.dbo.статья\_автор sa **on** a.aid = sa.aid

**join** кдз\_213.dbo.статья s **on** sa.article\_id = s.article\_id **join** кдз\_213.dbo.выпуск i

**on** i.issue\_id = s.issue\_id

**WHERE** i.год = 2021

**GROUP** **BY** a.Фамилия, a.Имя, a.Отчество

**HAVING** **COUNT**(sa.article\_id) > 0

### Запрос SELECT INTO для подготовки выгрузки

**insert** **into** **OPENROWSET**('Microsoft.Jet.OLEDB.4.0',

'Excel 8.0;Database=D:\testing.xls;',

'SELECT \* FROM dbo.издательство') **select** \* **from** dbo.издательство

## Триггеры

Триггер на ограничение удаления важных пользователей с индексом Хирша > 10 (рис.19).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 19. Триггер

## Индексы

Рассмотрим запрос, который показывает динамику изменений цитирований каждого журнала из года в год (рис. 20).

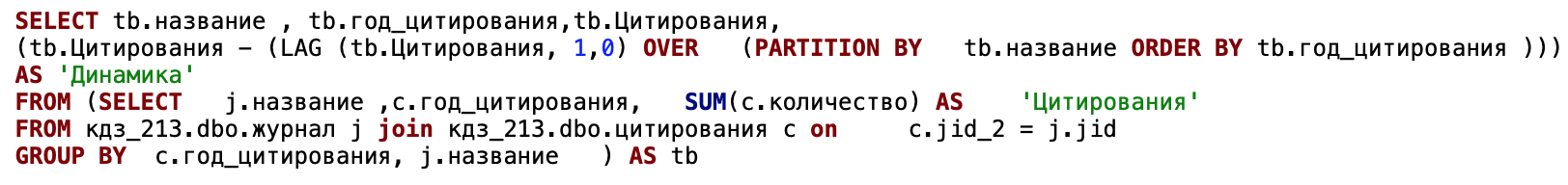


Рисунок 20. Пример запроса (1)

Среднее время его выполнения составляет от 300 до 400 ms (рис. 21).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 21. Результат (1)

Чтобы повысить производительность запроса, создадим индексы для таблицы «Цитирования», а именно для года цитирования и id статьи и количества (рис. 22).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 22. Пример создания индексов

Таким образом, не будет необходимости в сканировании большого количества таблиц, число операций ввода-вывода уменьшится и доступ к данным будет происходить быстрее.

На практике время обработки запроса значительно уменьшилось (рис. 23):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 23. Результат (2)

# Обязанности участниц проекта

Для успешной и продуктивной работы мы четко разделяли обязанности между собой, при этом проверяя работу друг друга и помогая, если необходимо. В таблице 1 представлены обязанности каждой участницы.

Таблица 1  
Обязанности участниц

|  |  |
| --- | --- |
| Участница | Обязанность |
| Казачинская Анастасия | Создание отчетов Power BI и Excel, создание триггеров, создание инфологической модели |
| Семина Анна | Создание отчета Word, монтаж видеоролика, создание базы данных, сбор и загрузка данных в базу, создание инфологической модели |
| Сторожок Мария | Написание хранимых процедур, функций и представлений, создание даталогической модели |
| Чургель Анастасия | Написание запросов, создание индексов, создание даталогической модели |