Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

**ОТЧЕТ по лабораторной работе №4**

по дисциплине «Базы данных»

**Выполнил:**

студент группы 5130902/20201 \_\_\_\_\_\_\_\_\_\_\_\_\_ А. И. Сафонов

подпись

**Проверил:**

Кандидат тех. наук., доцент \_\_\_\_\_\_\_\_\_\_\_\_ С. А. Нестеров

подпись

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.

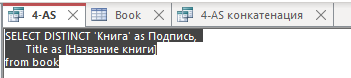
Санкт-Петербург, 2025

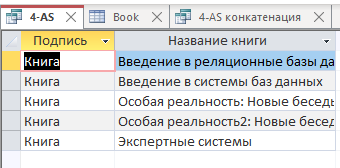
Оглавление

[Яндекс практикум 15](#_Toc190869791)

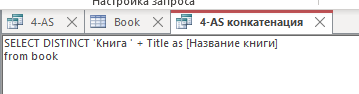
[Авиаперевозки 25](#_Toc190869792)

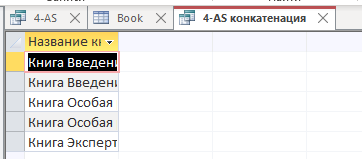
Задание. Напишите запрос, выводящий два столбца: первый должен называться «Подпись» и содержать во всех строках слово «Книга», второй столбец называется «Название книги» и содержит называния книг, взятых без повторения из таблицы Book.





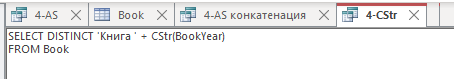
Задание. Измените предыдущий запрос таким образом, чтобы он выводил единственный столбец, в котором была бы подпись «Книга », а за ней в двойных кавычках шло название книги, например: «Книга "Экспертные системы"». Для сцепления строк используйте операцию «+».

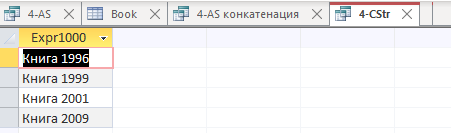




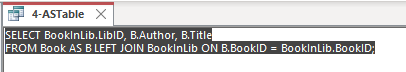
SELECT DISTINCT 'Книга ' + BookYear FROM Book;

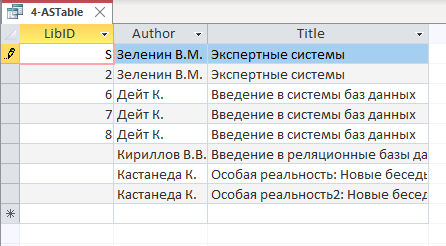
Задание. Измените представленный выше запрос так, чтобы он корректно сцеплял подпись и год издания.



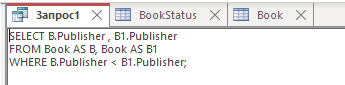


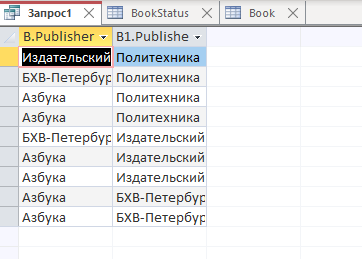
Задание. По аналогии с рассмотренным примером напишите свой запрос, использующий псевдонимы таблиц. Проверьте его работу.

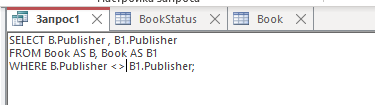


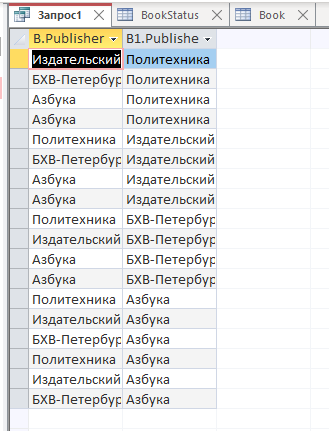


Задание. По аналогии с рассмотренным примером напишите свой запрос, выводящий без повторения все сочетания разных издательств. (self-join)

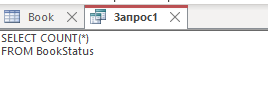


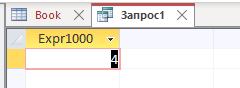




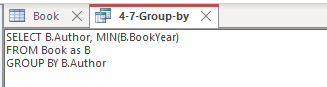


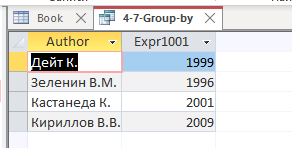
Задание. Напишите запрос, подсчитывающий количество статусов книг в таблице BookStatus.



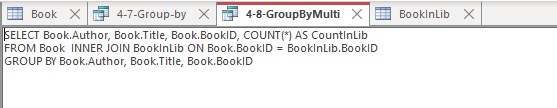


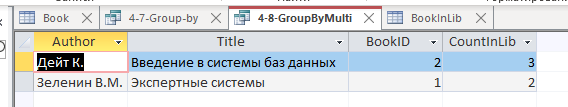
Задание. Напишите запрос, выводящий список авторов и год, в который каждый из них опубликовал первую книгу. При необходимости, измените набор данных в таблице Book так, чтобы там были авторы с несколькими книгами, изданными в разные годы. Проверьте работу вашего запроса.



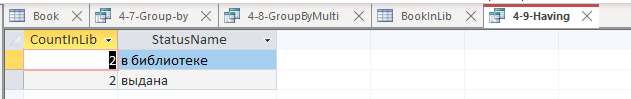
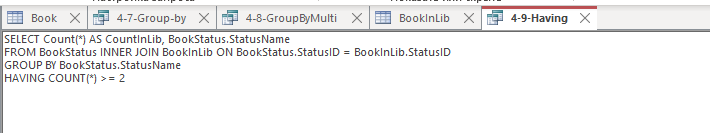


Задание. Для книг, экземпляры которых присутствуют в библиотеке (с любым статусом), выведите идентификатор издания, название, автора и число экземпляров книги.

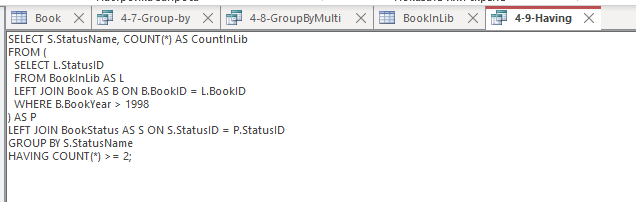


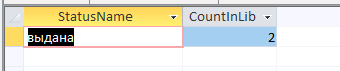


Задание. Напишите запрос, выводящий список названий статусов книг и количество экземпляров книг с эти статусом, но только если с этим статусом у нас 2 или более книги.

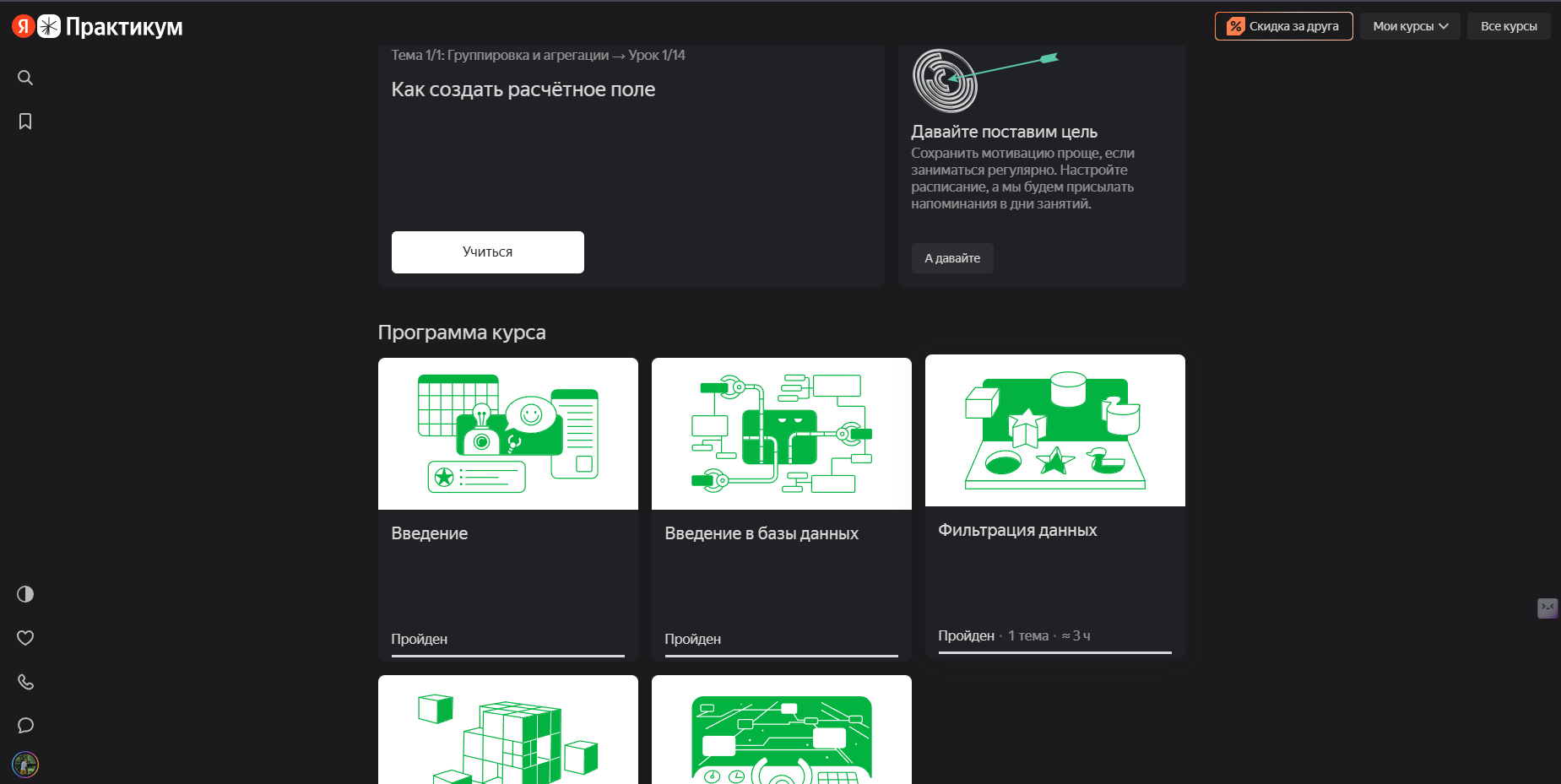


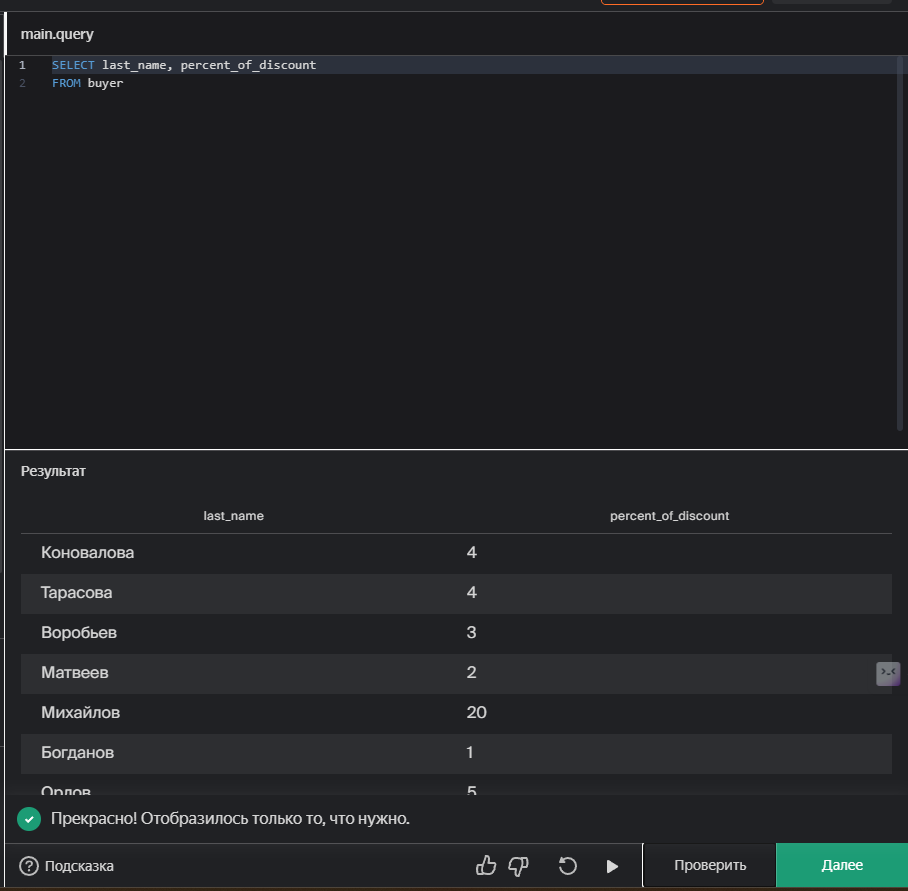
Задание. Модифицируйте запрос из предыдущего задания, учитывая при подсчете только книги, изданные после 1998 года.

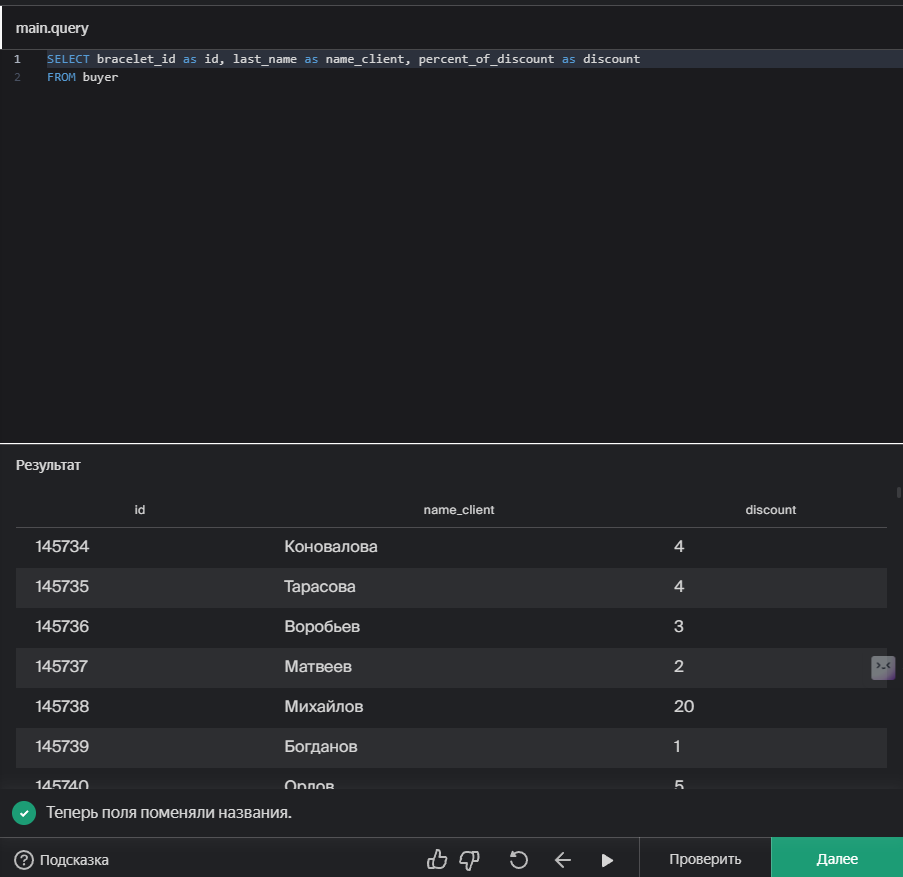


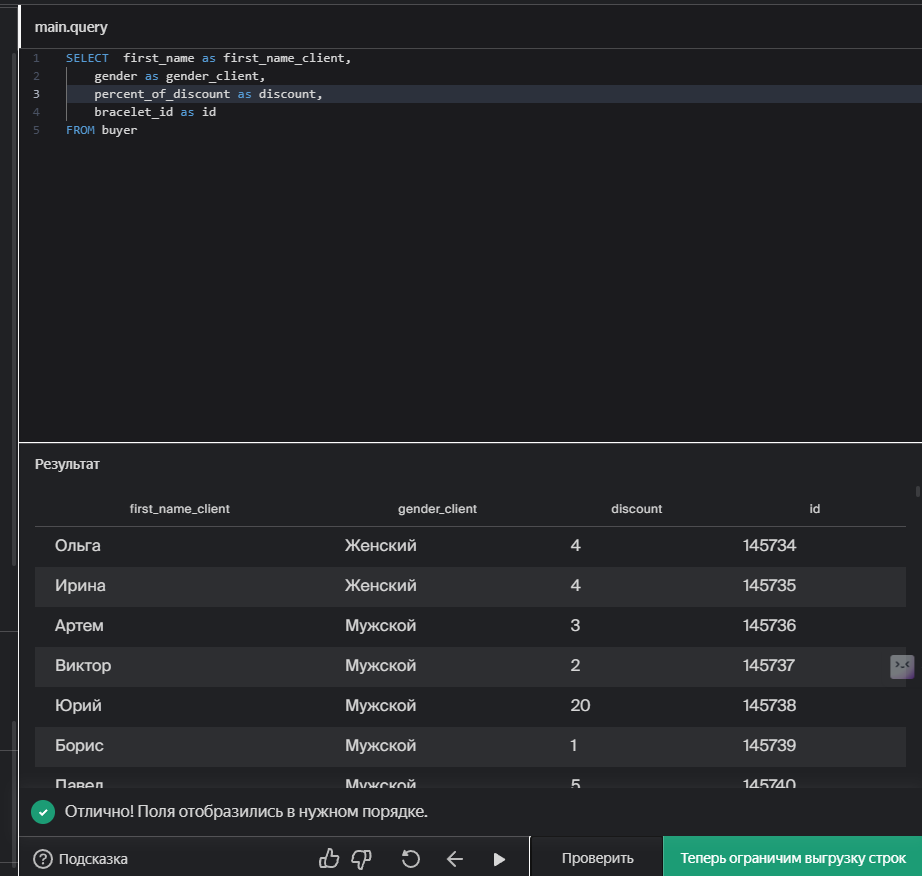


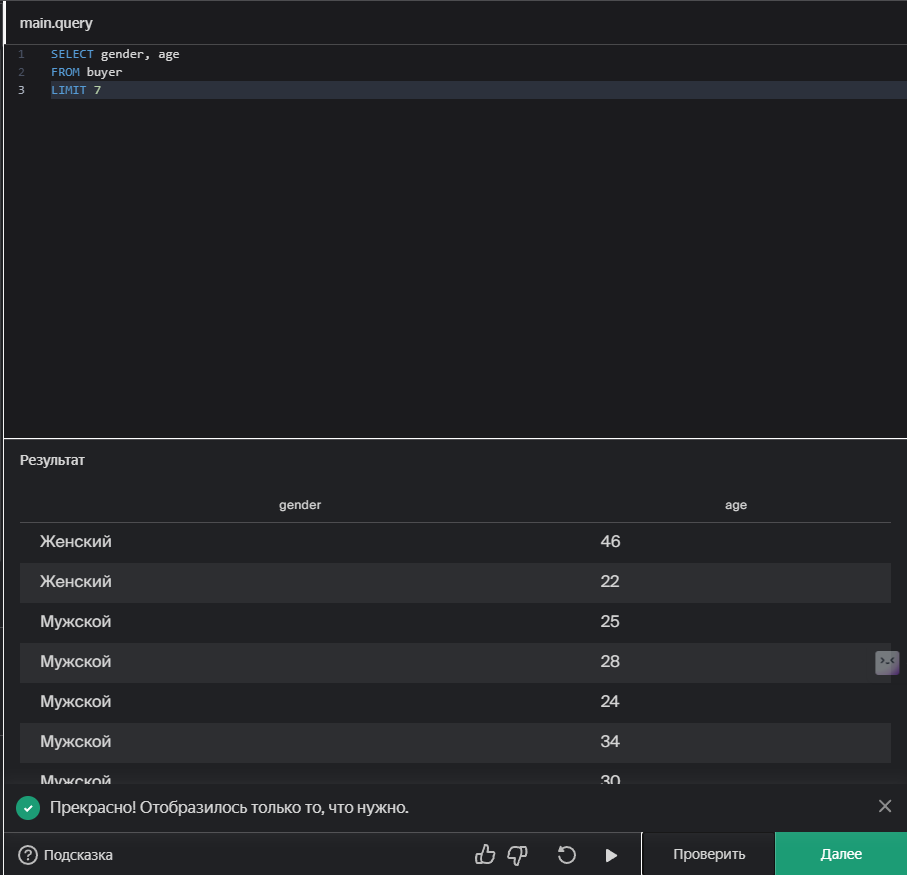
Яндекс практикум

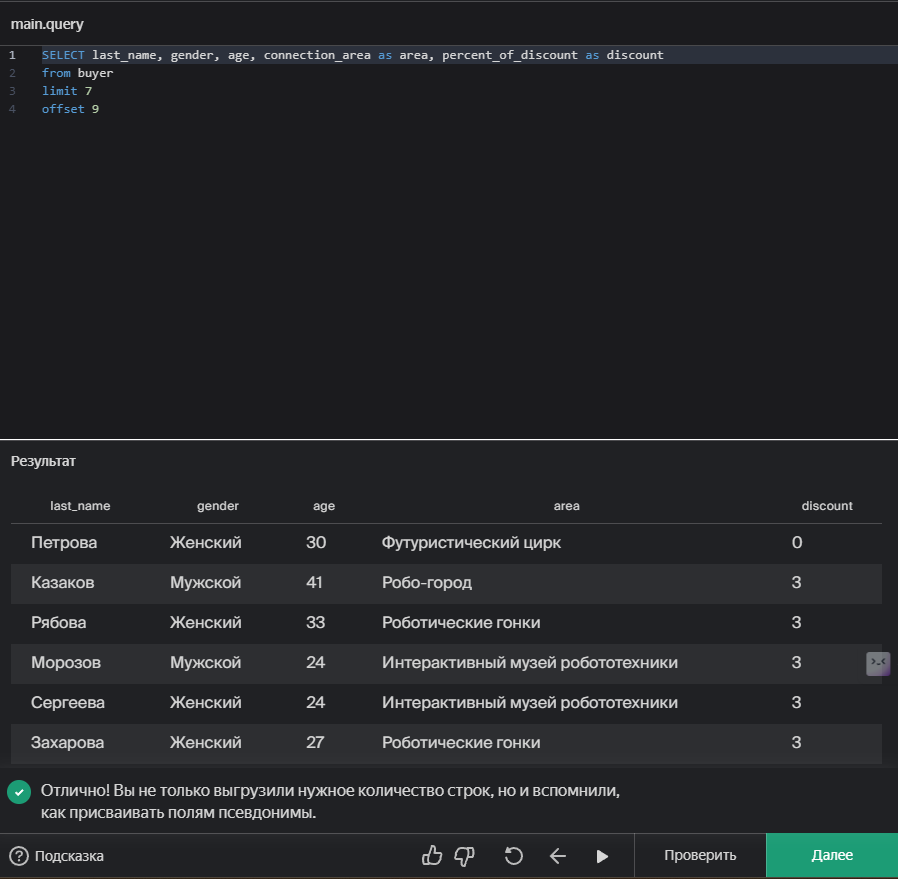


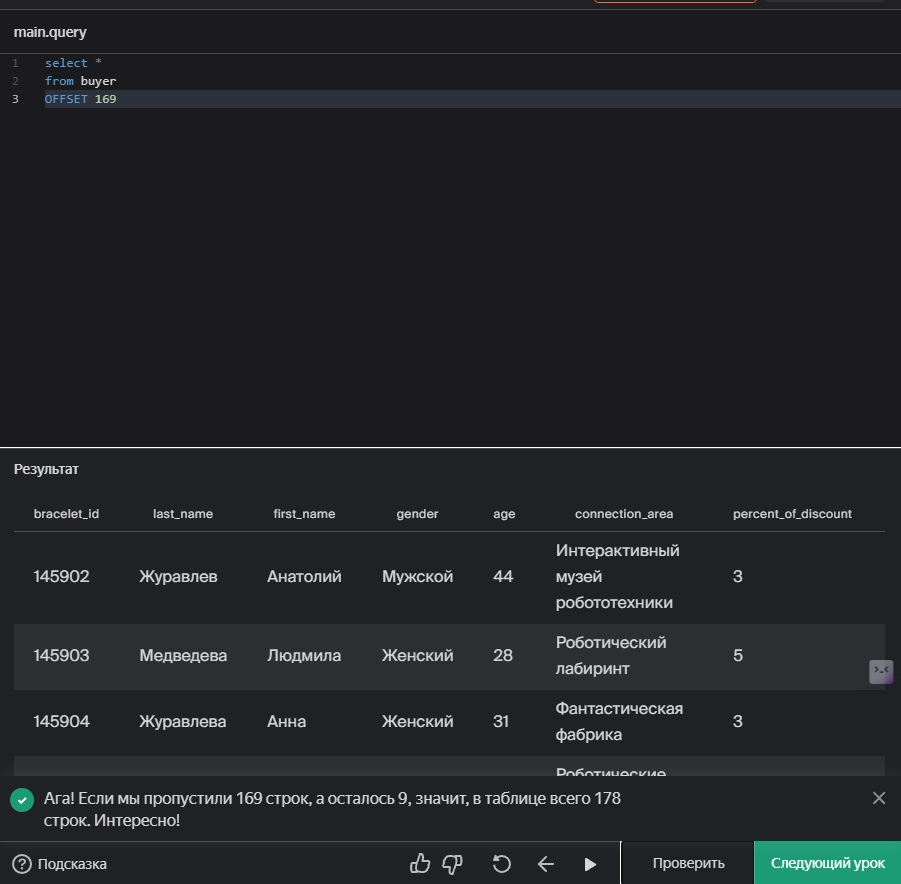










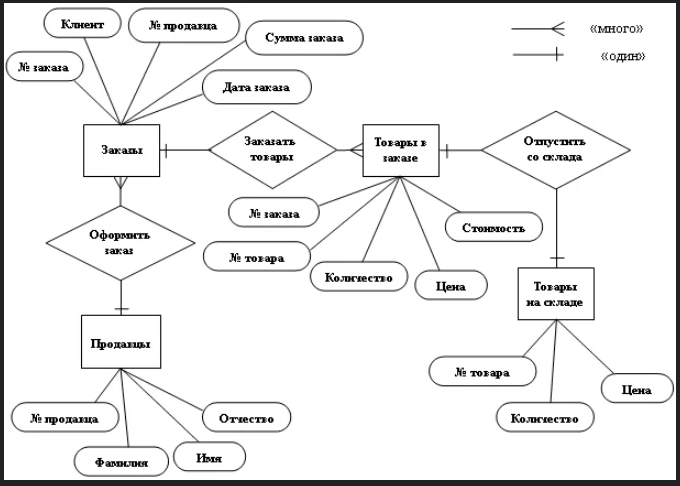


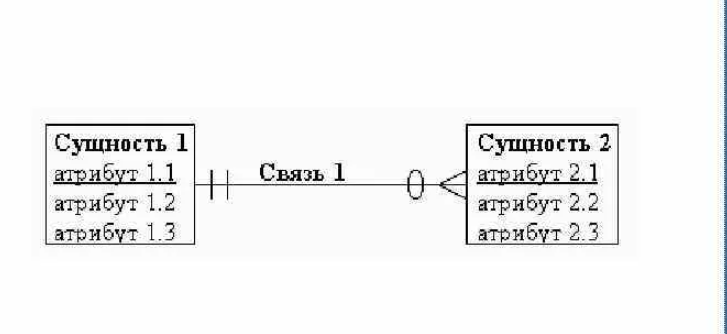
Литература

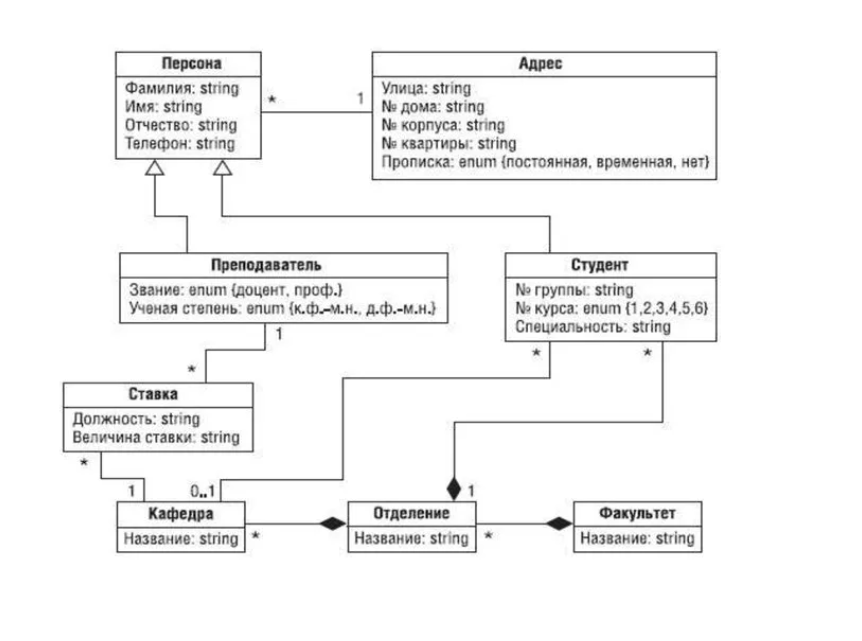
Семантическое моделирование, основанное на диаграммах сущность-связь (Entity-Relationship, ER-диаграммы), является важным этапом проектирования базы данных. Оно позволяет визуализировать структуру данных, их взаимосвязи и атрибуты, что делает процесс проектирования более понятным и эффективным.

Нотации ER-диаграмм:

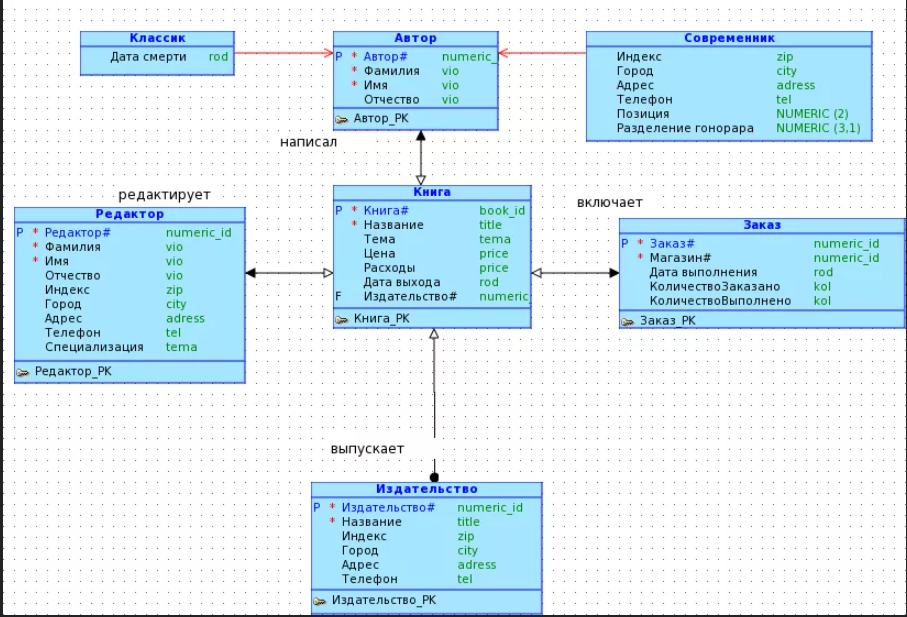
Нотация Чена. Сущности изображаются прямоугольниками, атрибуты — овалами, связи — ромбами.



* Нотация Мартина. Использует специальные символы для обозначения модальности (например, “воронья лапка” для связи “многие”). Популярна благодаря своей наглядности.  
  
* Нотация IDEF1X. Используется для моделирования реляционных баз данных



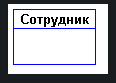
* Нотация Баркера. Разработана для Oracle. Использует простые и понятные обозначения.



Сущность (Entity)

Сущность – это класс однотипных объектов, информация о которых должна быть учтена в модели. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. Примерами сущностей могут быть такие классы объектов как “Поставщик”, “Сотрудник”, “Накладная”.

Каждая сущность в модели изображается в виде прямоугольника с наименованием:



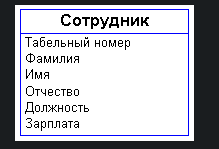
Экземпляр сущности

Экземпляр сущности – это конкретный представитель данной сущности. Например, представителем сущности “Сотрудник” может быть “Сотрудник Иванов”. Экземпляры сущностей должны быть различимы, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.

Атрибут сущности

Атрибут сущности – это именованная характеристика, являющаяся некоторым свойством сущности. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными). Примерами атрибутов сущности “Сотрудник” могут быть такие атрибуты как “Табельный номер”, “Фамилия”, “Имя”, “Отчество”, “Должность”, “Зарплата” и т.п.

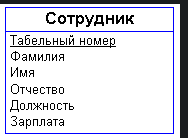
Атрибуты изображаются в пределах прямоугольника, определяющего сущность:



Ключ сущности

Ключ сущности – это неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Неизбыточность заключается в том, что удаление любого атрибута из ключа нарушается его уникальность. Сущность может иметь несколько различных ключей.

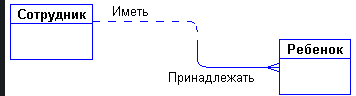
Ключевые атрибуты изображаются на диаграмме подчеркиванием:



Связь (Relationship)

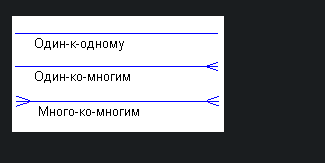
Связь – это некоторая ассоциация между двумя сущностями. Одна сущность может быть связана с другой сущностью или сама с собою. Связи позволяют по одной сущности находить другие сущности, связанные с нею. Например, связи между сущностями могут выражаться следующими фразами - “СОТРУДНИК может иметь несколько ДЕТЕЙ”, “каждый СОТРУДНИК обязан числиться ровно в одном ОТДЕЛЕ”.

Графически связь изображается линией, соединяющей две сущности:



Каждая связь имеет два конца и одно или два наименования. Наименование обычно выражается в неопределенной глагольной форме: “иметь”, “принадлежать” и т.п. Каждое из наименований относится к своему концу связи. Иногда наименования не пишутся ввиду их очевидности.

Каждая связь может иметь один из следующих типов связи:

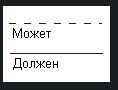


Связь типа один-к-одному означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь один-к-одному чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две.

Связь типа один-ко-многим означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой). Это наиболее часто используемый тип связи. Левая сущность (со стороны “один”) называется родительской, правая (со стороны “много”) - дочерней.

Связь типа много-ко-многим означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.

Каждая связь может иметь одну из двух модальностей связи:



Модальность “может” означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром.

Модальность “должен” означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.

Потенциальный ключ – это такой суперключ, который не содержит подмножества атрибутов, также являющегося суперключом данного отношения. Другими словами, если мы уберем из множества атрибутов потенциального ключа любой атрибут, оставшиеся уже не будут обладать свойством уникальности.

**ПРАКТИЧЕСКОЕ ЗАДАНИЕ ТРЕТЬЕЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ»**

1. Определите потенциальные ключи для каждого из отношений. Если для каких-то отношений таких ключей окажется несколько, выберите один из них на роль первичного (PK), а остальные пометьте альтернативными (AK). Правила выбора первичного ключа перечислены во второй части конспекта лекции.

При выборе первичного ключа ориентируются на следующие

критерии:

 все атрибуты, входящие в его состав, должны быть

обязательными, то есть не должны содержать неизвестных

значений;

 значения атрибутов первичного ключа не должны меняться со

временем, так как именно этот ключ будет использоваться в

качестве основного критерия для идентификации отдельных

кортежей;

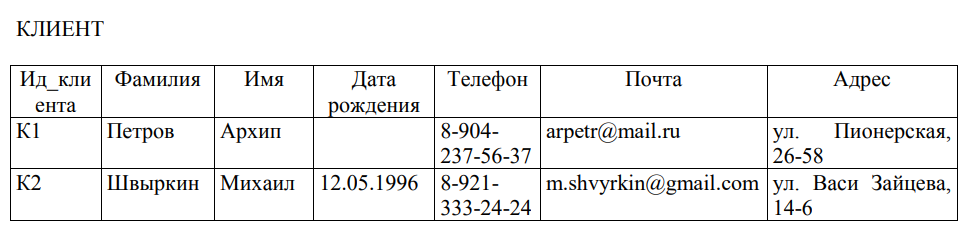
 также значения первичного ключа используются при поиске

данных в базе, поэтому в целях увеличения скорости поиска в

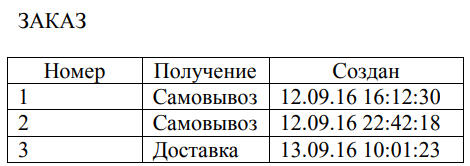
качестве первичного ключа стараются выбрать такой,

совокупная допустимая длина значений атрибутов которого –

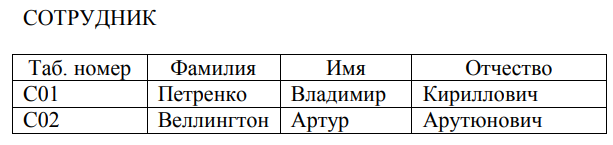
минимальна



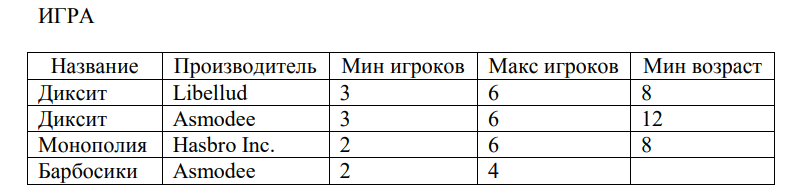
PK – Ид\_клиента, AK – Телефон, Почта



PK – Номер



PK – Таб. номер

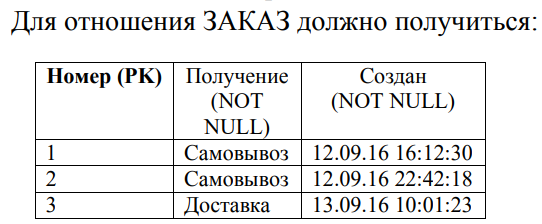


PK – Название, Производитель

2. Определите, какие из атрибутов отношений являются обязательными,

т.е. не допускают наличие неизвестных значений. Пометьте такие

атрибуты как NOT NULL.



***NOT NULL:***

КЛИЕНТ: Ид\_Клиента, Имя, Почта

ИГРА: Название+Производитель, Мин игроков, Макс игроков

ЗАКАЗ: Номер (PK), Получение, Создан

СОТРУДНИК: Таб. Номер (PK), Имя, Фамилия

3. Определите условия проверки значений для атрибутов (где возможно и, на ваш взгляд, имеет смысл).

КЛИЕНТ: Телефон LIKE '8-???-???-??-??', Почта LIKE '\*@\*.\*', Дата Рождения <= Date()

ИГРА: Мин игроков >=1, Мин возраст <= 21

ЗАКАЗ: Получение = 'Самовывоз' OR Получение = 'Доставка’

4. Определите связи между отношениями.

а. Определение типа связи (один-к-одному, один-ко-многим или многие-ко-многим):

**1)** Один заказ оформляется ОДНИМ клиентом

Один клиент может оформить МНОГО заказов

***Связь: один-ко-многим***

**2)** Один заказ должен быть обработан ОДНИМ сотрудником

Один сотрудник может вести МНОГО заказов

***Связь: один-ко-многим***

**3)** Один заказ может содержать МНОГО игр

Одна игра может находиться в МНОГИХ заказах

***Связь: многие-ко-многим***

b. Установите связь между отношениями, добавив атрибуты первичного ключа главного отношения (того, к которому относится слово ОДИН) в состав дочернего (того, к которому относится слово МНОГО) и отметив их как внешний ключ (FK).

1) Добавляем Ид\_Клиента в ЗАКАЗ как FK

2) Добавляем Таб. Номер в ЗАКАЗ как FK

3) Создаем промежуточную таблицу ***ПОЗИЦИИ\_ЗАКАЗА***:

1. ***Номер\_заказа*** – внешний ключ на ЗАКАЗ(Номер)
2. ***Название\_игры+Производитель*** – внешний ключ на ИГРА(Название+Производитель)
3. Дополнительный атрибут – Количество\_экземпляров, который показывает, сколько экземпляров игры заказано

*Номер\_заказа+Название\_игры+Производитель* – составной PK для отношения ПОЗИЦИИ\_ЗАКАЗА

Изображение выглядит как текст, диаграмма, Параллельный, линия

Контент, сгенерированный ИИ, может содержать ошибки.

**ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ**

1. Допустим, что каждый сотрудник может обработать только один заказ (после чего его увольняют). В этом случае связь между отношениями ЗАКАЗ и СОТРУДНИК трансформируется в один-к-одному. Что нужно будет сделать дополнительно для поддержки этой связи? Что может помешать указать одного и того же сотрудника в отношении ЗАКАЗ, ответственным за обработку нескольких разных заказов?

***Ответ:***

Чтобы обеспечить связь **один-к-одному**, нужно сделать так, чтобы **внешний ключ** в дочерней таблице (в данном случае, Таб. номер в ЗАКАЗ) **не допускал повторяющихся значений**. Для этого на него накладывается **ограничение уникальности**, что делает его альтернативным ключом. Это гарантирует, что один и тот же сотрудник не сможет быть назначен на несколько заказов.

2. Представим, что некоторые из сотрудников являются начальниками других (каждый сотрудник может иметь не больше одного начальника) и нам необходимо хранить информацию о том, кто чьим начальником является. Попробуйте определить, какую связь для этого нужно добавить в нашу модель.

***Ответ:***

Начальники сотрудников также являются сотрудниками, поэтому должна быть установлена связь отношения с самим собой. Связи такого рода называют **рекурсивными.**

Для наглядности создадим копию отношения СОТРУДНИК, назовем ее НАЧАЛЬНИК и определим тип связи между копиями:

• один сотрудник может иметь не больше ОДНОГО начальника

• один начальник может руководить МНОГИМИ сотрудниками

Тип связи – один-ко-многим. В дочернюю копию отношения, которая называется СОТРУДНИК необходимо добавить атрибут первичного ключа из родительской копии. Т.к. в одном отношении не может быть атрибутов с одинаковым именем, назовем его «Таб. номер начальника». Вновь добавленный атрибут не нужно делать обязательным, т.к. в организации должен быть самый главный босс, начальников не имеющий (и, к тому же, необходимо иметь возможность добавить самого первого сотрудника, номер начальника которого неизвестен за неимением других сотрудников).

Практическое задание. Написание однотабличных запросов в Ред База Данных.

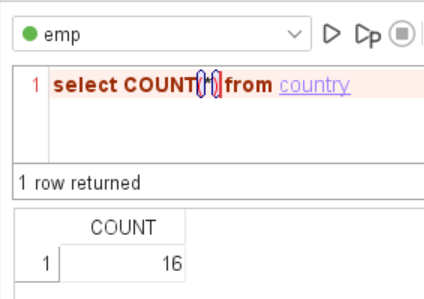
1) Начнем с таблицы country. Выберите все записи (таблица небольшая), опишите содержимое таблицы.



COUNTRY - VARCHAR

CURRENCY – VARCHAR

2) Посчитайте число записей в таблице country.



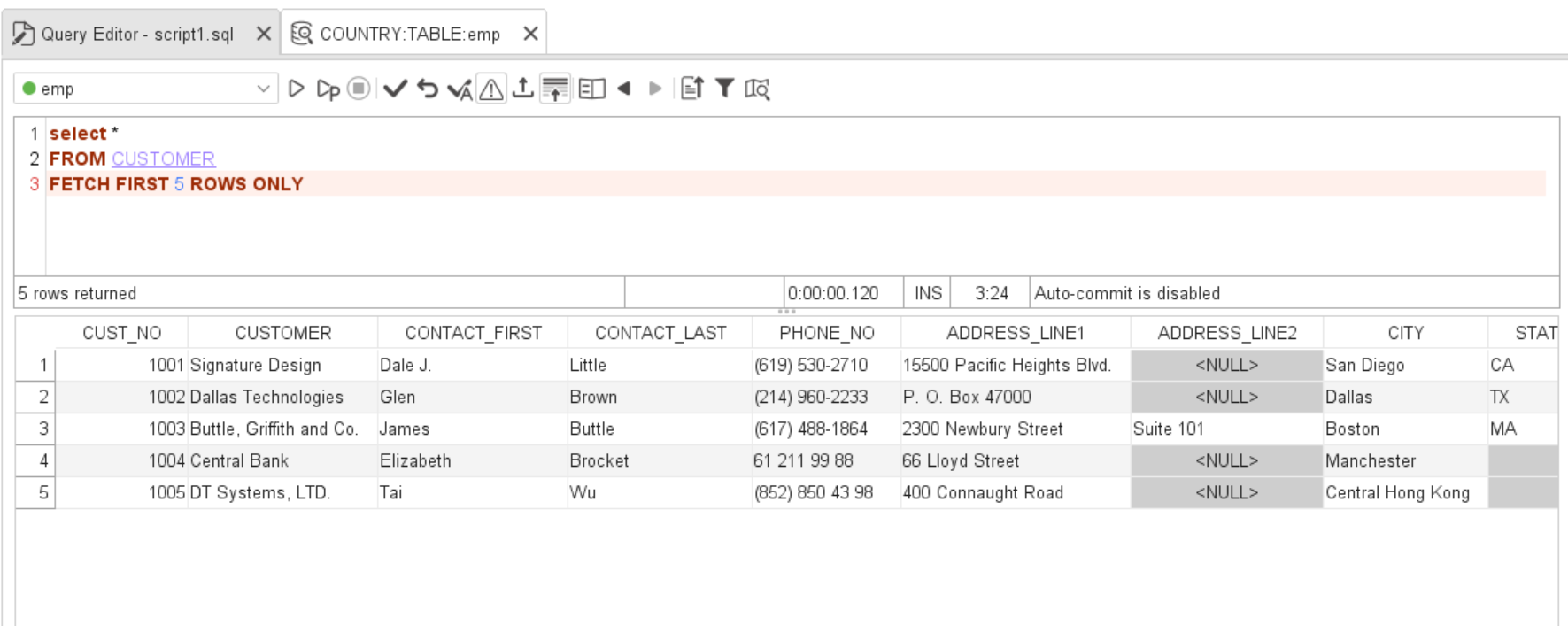
3) Посчитайте число уникальных значений поля currency. Как оно соотносится с общим числом записей в таблице?





4) Для ограничения числа выводимых записей в FireBird/Ред БД используются инструкции FETCH … ROWS ... Например, первые 10 записей таблицы People, упорядоченной по полю name (больше примеров можно найти в документации): SELECT \* FROM T1 ORDER BY COL1 FETCH FIRST 10 ROWS ONLY

Выведите первые 5 записей таблицы customer, кратко опишите имеющиеся в ней поля.



CUST\_NO - INTEGER

CUSTOMER - VARCHAR

CONTACT\_FIRST - VARCHAR

CONTACT\_LAST - VARCHAR

PHONE\_NO - VARCHAR

ADDRESS\_LINE1 - VARCHAR

ADDRESS\_LINE2 - VARCHAR

CITY - VARCHAR

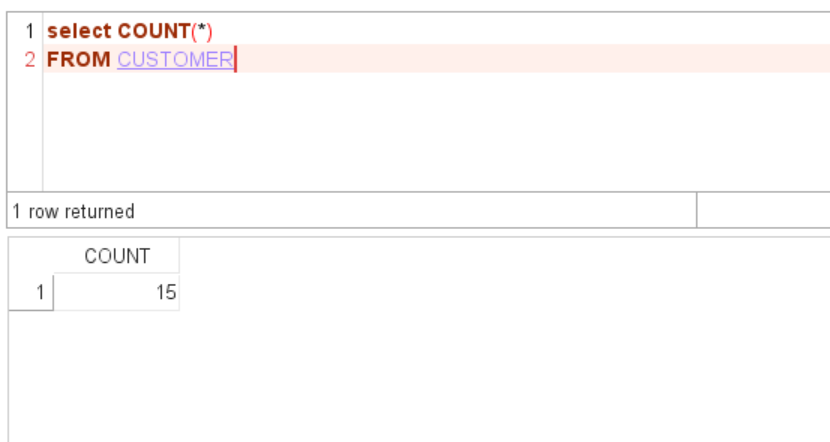
STATE\_PROVINCE - VARCHAR

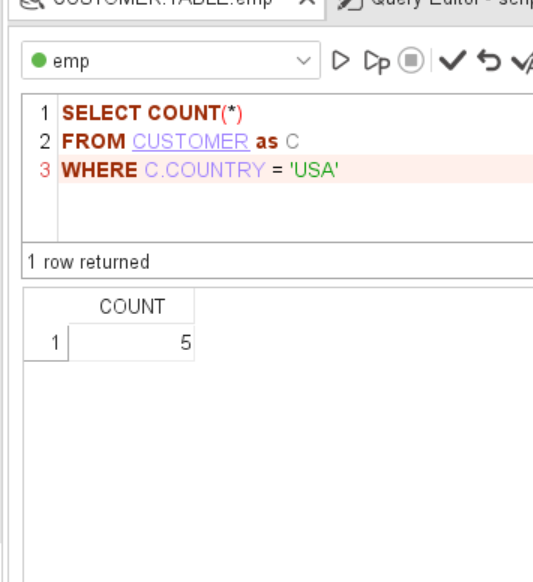
COUNTRY - VARCHAR

POSTAL\_CODE - VARCHAR

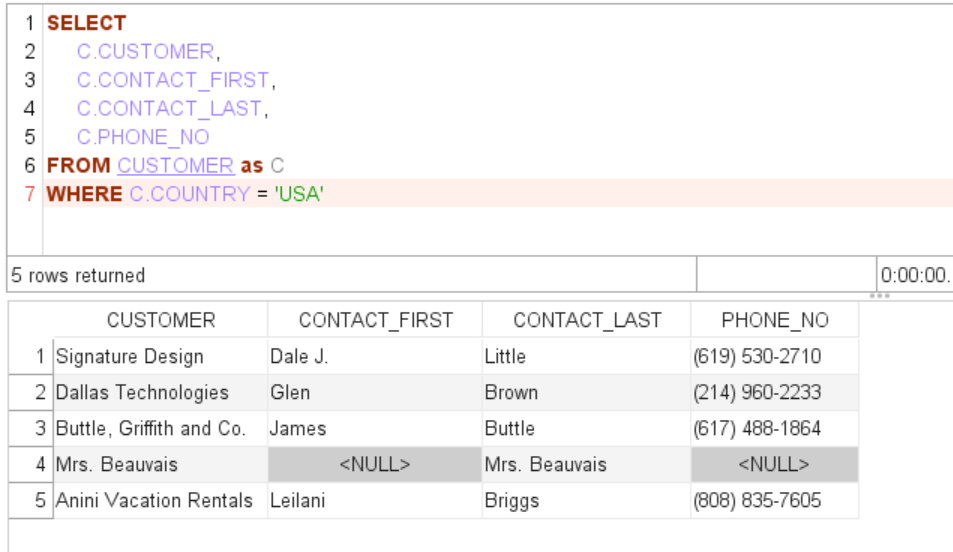
ON\_HOLD - CHAR

5) Посчитайте общее число записей в таблице и число записей о заказчиках из США (USA)

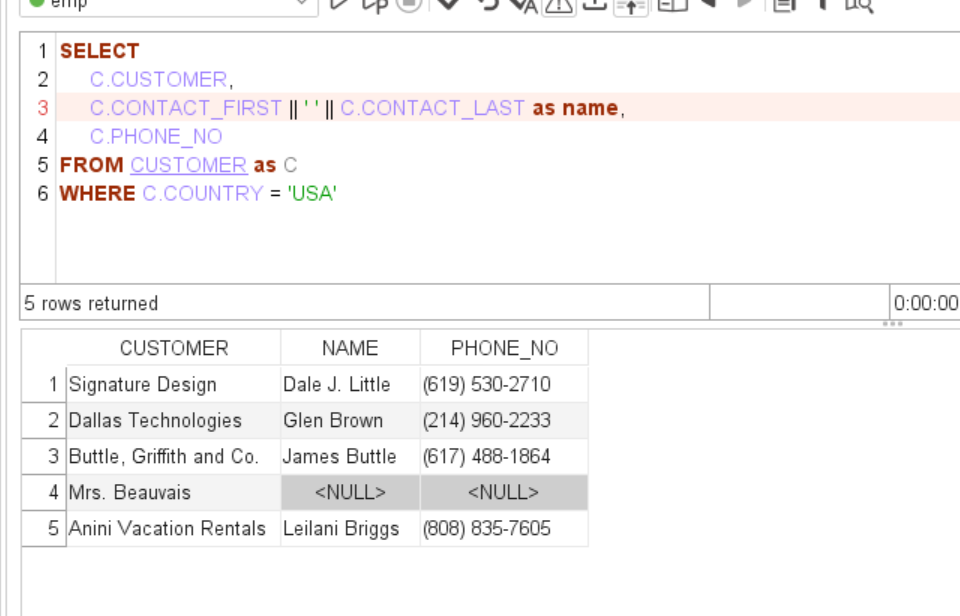




6) Про заказчиков из США выведите (4 столбца): - название компании; - имя контактного лица; - фамилию контактного лица; - телефон.



7) Про заказчиков из США выведите (3 столбца; используйте оператор конкатенации - ||): - название компании; - имя и фамилию контактного лица в одном столбце через пробел c названием столбца name; - телефон. Обратите внимание на запись, где часть имени была не задана. Что сейчас в столбце name?



8) В подобном случае можно использовать функцию COALESCE (, [, ]...), которая возвращает первое непустое значение из списка в скобках. В скобках может стоять как название столбца, так и константа или какое-то выражение. При необходимости по документации посмотрите описание функции coalesce(). Перепишите запрос 7) так, чтобы в случае пустого значения имени, оно заменялось на пробел перед фамилией.

