

## Лабораторная работа № 3 по теме: «Логическое проектирование. Модели в нотации IDEF1X»

**Цель:** получение практических навыков составления ER-моделей в нотации IDEF1X, нормализация

### Ход работы:

1. Изучите теоретическую часть
2. Оформите титульный лист
3. Выполните задания (по вариантам) и оформите отчет
4. Ответьте на контрольные вопросы (устно или письменно)
5. Сдайте лабораторную работу преподавателю (на следующей паре)

### Отчет по лабораторной работе должен содержать (выполняется в MS Word):

1. ФИ, группа в верхнем колонтитуле на каждой странице, кроме первой
2. Первая страница – титульный лист
3. Каждая страница пронумерована, кроме первой
4. Выполненные задания начиная со второй страницы документа (каждое задание подписано, при необходимости присутствуют скриншоты)
5. Ответы на **контрольные вопросы** (можно устно, либо письменно, на последней странице отчета)
  - 1) Для чего предназначено логическое проектирование, чем оно отличается от концептуального?
  - 2) Этапы логического проектирования
  - 3) Что такое сложная связь в БД? Как от нее избавиться?
  - 4) Что такое рекурсивная связь в БД? Как от нее избавиться?
  - 5) Что такое избыточная связь в БД? Как от нее избавиться?
  - 6) Что такое нормальная форма. Как доказать, что модель БД в 1НФ, 2НФ, 3НФ?
  - 7) Что такое декомпозиция и для чего она применяется?
  - 8) Что такое декомпозиция без потерь?

### Теоретическая часть:

Цель логического проектирования – развить концептуальную схему БД с учетом выбранной модели БД (иерархической, сетевой, реляционной и т.д.).

#### Этапы логического проектирования:

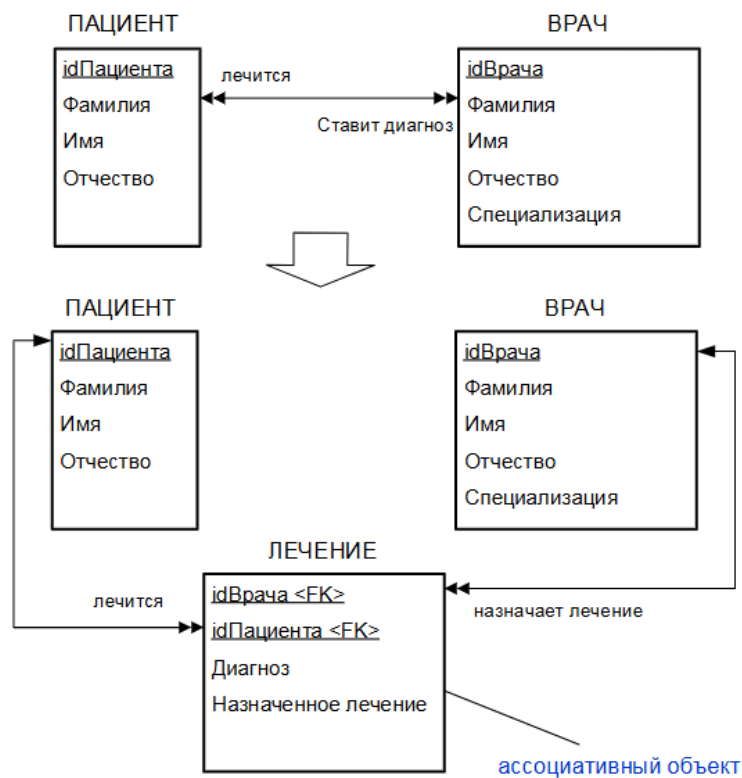
##### 1 этап – Удаление и проверка элементов, не отвечающих принятой модели данных

###### *а) Удаление связей многие-ко-многим*

Если в концептуальной схеме присутствуют связи N:M, то их следует устранить путем определения промежуточной (ассоциативной) сущности. Связь N:M заменяется двумя связями типа 1:M, устанавливаемыми со вновь созданной сущностью.

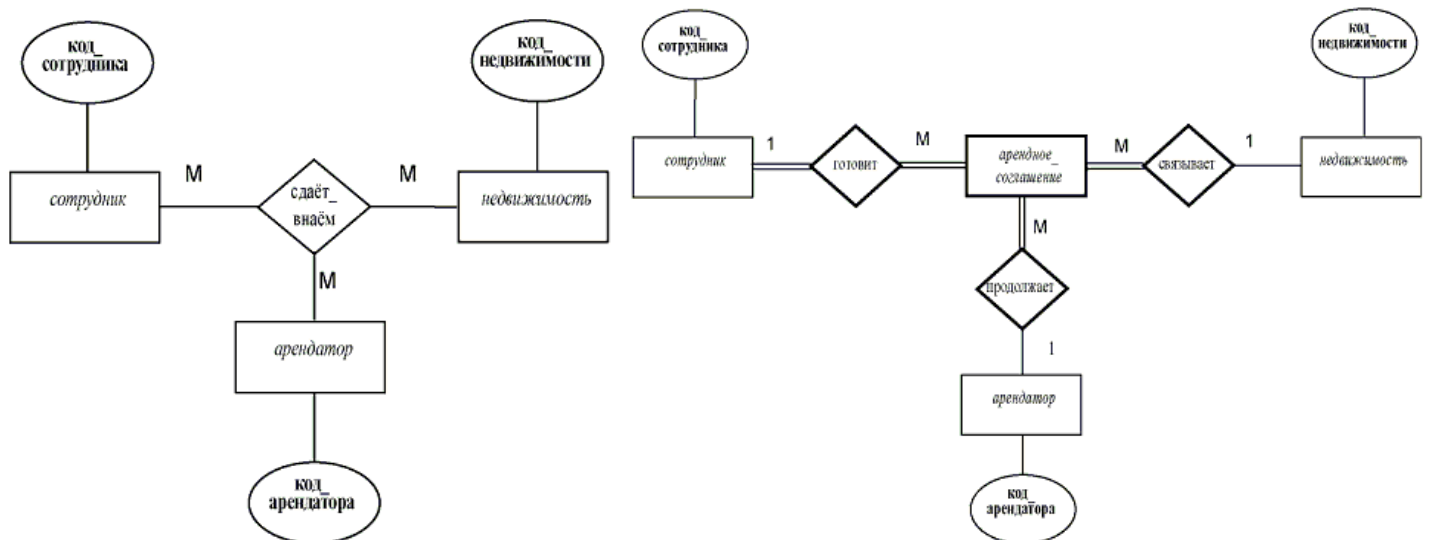
###### *б) Удаление связей с атрибутами*

Связи с атрибутами должны быть преобразованы в сущности. Например, есть модель Пациент-Врачи. Связь многие-ко-многим. При этом диагноз и лечение предназначены для одного пациента, но назначены врачом. Таким образом, куда бы мы не поместили атрибуты «диагноз» и «лечение», они будут зависимы от обеих сущностей. Создадим новую сущность ДИАГНОЗ, и эта проблема будет решена.



с) Удаление сложных связей (со степенью участия более 2)

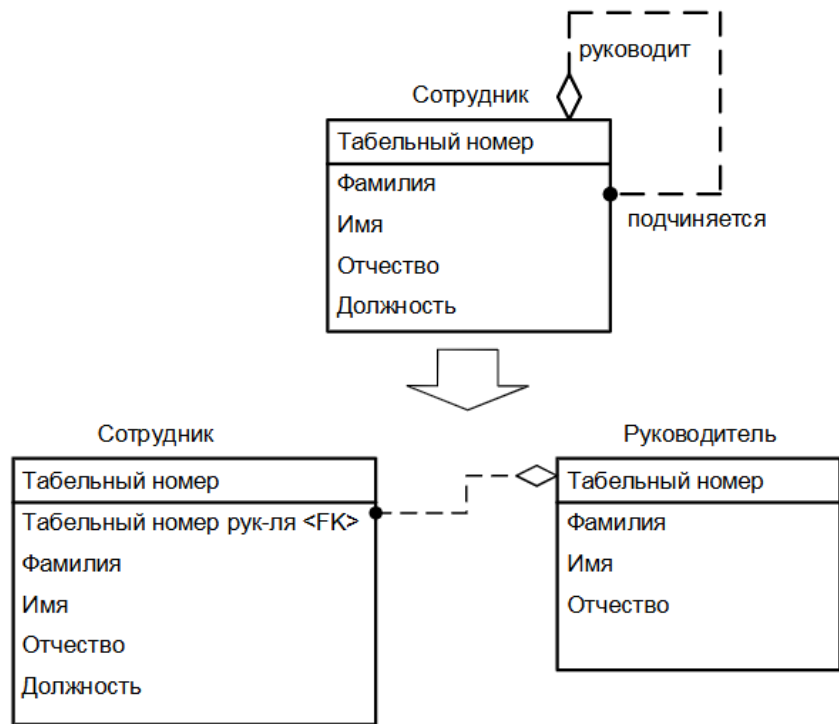
**Сложная связь в базе данных (БД)** — это связь со степенью участия более двух. Степень участия определяет количество сущностей, участвующих в связи. Сложную связь заменяют необходимым количеством бинарных связей 1:M со вновь созданной сущностью, которая и показывает эту связь.



d) Удаление рекурсивных связей

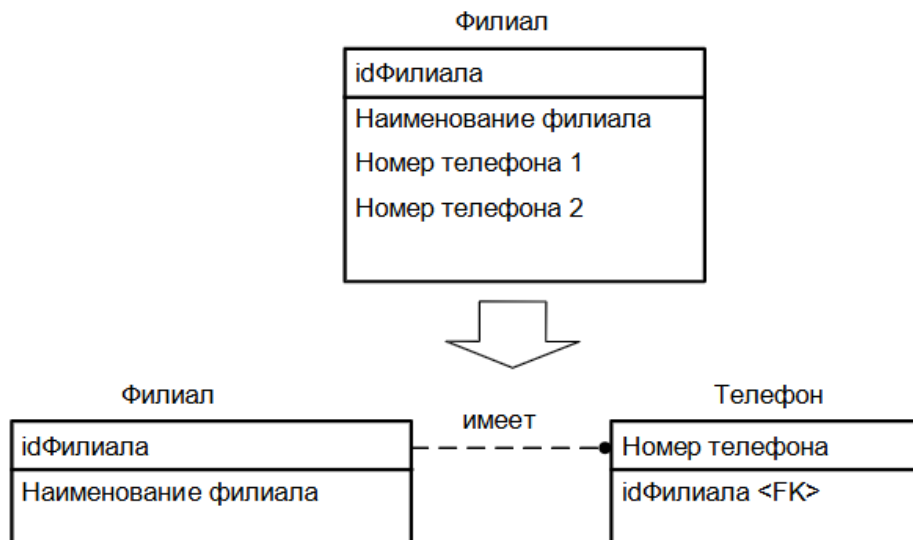
**Рекурсивная связь** — связь, в которой один и тот же тип сущности участвует более одного раза в разных ролях. Если концептуальная модель содержит рекурсивные связи, они должны быть устранены посредством определения некоторой промежуточной сущности.

Пример удаления рекурсивной связи в базе данных:



е) Удаление многозначных атрибутов (атрибутов имеющих несколько значений)

Многозначность устраняется путем введения новой сущности и связи 1: М. Например, несколько номеров организации выводятся в отдельную сущность "Телефоны". Данное преобразование, помимо соответствия реляционной модели данных, также позволяет хранить любое количество телефонов по одному филиалу.



ф) Удаление избыточных связей.

Связь является избыточной, если одна и та же информация может быть получена не только через нее, но и с помощью другой связи.

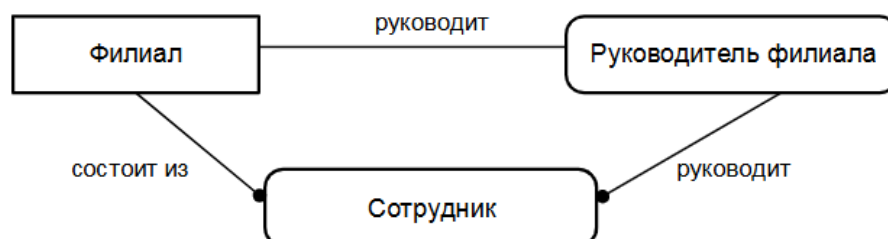


Рис. 32. Избыточные связи

В приведенном примере одну из связей «Руководит» можно смело удалить (лучше между «Руководителем филиала» и «Сотрудником»).

г) Перепроверка связей 1:1

В процессе определения сущностей могли быть созданы сущности, которые на самом деле являются одной. В этом случае их следует объединить. Например, из примера выше сущности «Филиал» и «Руководитель филиала» лучше объединить.

В то же время не всегда можно выполнить такое объединение.

## 2 этап - проверка модели с помощью правил нормализации

Основная идея нормализации заключается в том, чтобы каждый факт хранился в одном месте, т. е. чтобы не было дублирования данных. Многие из требований нормализации, как правило, уже учитываются при выполнении предыдущих шагов проектирования.

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- первая нормальная форма (1НФ);
- вторая нормальная форма (2 НФ);
- третья нормальная форма (3 НФ);
- нормальная форма Бойса-Кодда (БКНФ);
- четвертая нормальная форма (4 НФ);
- пятая нормальная форма, или нормальная форма проекции-соединения (5 НФ).

Наибольшую практическую значимость имеют первые три нормальные формы. Аномалии более высоких форм не оказывают существенного влияния на результаты обработки отношений и встречаются редко.

Основные свойства нормальных форм:

- каждая следующая нормальная форма в некотором смысле лучше предыдущей
- при переходе к следующей нормальной форме свойства предыдущих нормальных свойств сохраняются

### Первая нормальная форма

1НФ	Отношение находится в первой нормальной форме (1НФ), если все его атрибуты являются атомарными, т.е. состоящими из неделимых значений.
-----	--

Понятие атомарности является условным: будем считать значение атомарным, если оно не используется по частям. То есть ФИО не обязательно разбивать на отдельные атрибуты, если в будущей БД не будет запросов, типа: вывести всех Ивановых и т.д.

2НФ	Отношение находится во второй нормальной форме (2НФ) в том и только в том случае, когда находится в 1НФ, и каждый неключевой атрибут полностью зависит от первичного ключа.
-----	---

По-другому, правила 2НФ выглядят следующим образом:

- Таблица должна находиться в первой нормальной форме
- Таблица должна иметь ключ
- Все неключевые столбцы таблицы должны зависеть от полного ключа (в случае если он составной)

Если ключ составной, т.е. состоит из нескольких столбцов, то все остальные неключевые столбцы должны зависеть от всего ключа, т.е. от всех столбцов в этом ключе. Если какой-то атрибут (столбец) зависит только от одного столбца в ключе, значит, база данных не находится во второй нормальной форме.

Иными словами, в таблице не должно быть данных, которые можно получить, зная только половину ключа, т.е. только один столбец из составного ключа. **Например.**

ФИО	Должность	Подразделение	Описание подразделения
Иванов И.И.	Программист	Отдел разработки	Разработка и сопровождение приложений и сайтов
Сергеев С.С.	Бухгалтер	Бухгалтерия	Ведение бухгалтерского и налогового учета финансово-хозяйственной деятельности
John Smith	Продавец	Отдел реализации	Организация сбыта продукции

Она удовлетворяет условию 1 НФ, что все значения атомарны (в данном случае ФИО как разные атрибуты не принципиально).

Теперь мы можем начать процесс нормализации этой таблицы до второй нормальной формы.

Что для этого нам нужно сделать? Нам нужно внедрить первичный ключ.

Очевидно, что для таблицы, которая будет хранить список сотрудников, первичным ключом может выступать табельный номер, зная который мы можем четко идентифицировать каждого сотрудника, т.е. каждую строку нашей таблицы. Если бы такого табельного номера у нас не было или в рамках организации он мог повторяться (например, сотрудник уволился, и спустя время его номер присвоили новому сотруднику), то для первичного ключа мы могли бы создать искусственный ключ с целочисленным типом данных, который автоматически увеличивался бы в случае добавления новых записей в таблицу. Тем самым мы бы точно также четко идентифицировали каждую строку в таблице.

Таким образом, чтобы привести эту таблицу ко второй нормальной форме, мы должны добавить в нее еще один атрибут, т.е. столбец с табельным номером.

Таблица сотрудников во второй нормальной форме с простым первичным ключом.

Табельный номер	ФИО	Должность	Подразделение	Описание подразделения
1	Иванов И.И.	Программист	Отдел разработки	Разработка и сопровождение приложений и сайтов
2	Сергеев С.С.	Бухгалтер	Бухгалтерия	Ведение бухгалтерского и налогового учета финансово-хозяйственной деятельности
3	John Smith	Продавец	Отдел реализации	Организация сбыта продукции

В результате, так как наш первичный ключ является простым, а не составным, наша таблица автоматически переходит во вторую нормальную форму.

**Иными словами, если первичный ключ простой (не составной, т.е. состоящий из одного столбца), второе требование, которое предъявляется к таблицам для перехода во вторую нормальную форму, выполнять не требуется, так как оно относится только к таблицам, у которых первичный ключ составной.**

**Пример приведения таблицы ко второй нормальной форме (первичный ключ составной)**

Представим, что наша организация выполняет несколько проектов, в которых может быть задействовано несколько участников, и нам необходимо хранить информацию об этих проектах. В частности мы хотим знать, кто участвует в каждом из проектов, продолжительность этого проекта, ну и возможно какие-то другие сведения. При этом мы понимаем, что отдельно взятый сотрудник может участвовать в нескольких проектах.

Для хранения таких данных мы создали следующую таблицу.

Название проекта	Участник	Должность	Срок проекта (мес.)
Внедрение приложения	Иванов И.И.	Программист	8
Внедрение приложения	Сергеев С.С.	Бухгалтер	8
Внедрение приложения	John Smith	Менеджер	8
Открытие нового магазина	Сергеев С.С.	Бухгалтер	12
Открытие нового магазина	John Smith	Менеджер	12

Как видим, она в первой нормальной форме, значит, мы можем пытаться приводить ее ко второй нормальной форме.

Как Вы помните, чтобы привести таблицу ко второй нормальной форме, необходимо определить для нее первичный ключ.

Посмотрев на эту таблицу, мы понимаем, что четко идентифицировать каждую строку мы можем только с помощью комбинации столбцов, например, «Название проекта» + «Участник», иными словами, зная «Название проекта» и «Участника», мы можем четко определить конкретную запись в таблице, т.е. каждое сочетание значений этих столбцов является уникальным.

Таким образом, мы определили первичный ключ и он у нас составной, т.е. состоящий из двух столбцов (название проекта и участник).

Так как первичный ключ составной, нам необходимо проверить еще и второе требование, которое гласит, что «Все неключевые столбцы таблицы должны зависеть от полного ключа».

Другими словами, остальные столбцы, которые не входят в первичный ключ, должны зависеть от всего первичного ключа, т.е. от всех столбцов, а не от какого-то одного.

Чтобы это проверить, мы можем задать себе несколько вопросов.

Можем ли мы определить «Должность», зная только название проекта? Нет. Для этого нам необходимо знать и участника. Значит, пока все хорошо, по этой части ключа мы не можем четко определить значение неключевого столбца. Идем дальше и проверяем другую часть ключа.

Можем ли мы определить «Должность» зная только участника? Да, можем. Значит наш первичный ключ плохой, и требование второй нормальной формы не выполняется.

Что делать в этом случае?

В этом случае мы будем выполнять действие, которое выполняется, наверное, в 99% случаев на протяжении всего процесса нормализации базы данных – это декомпозиция.

**Декомпозиция** – это процесс разбиения одного отношения (таблицы) на несколько.

Чтобы декомпонировать нашу таблицу и привести базу данных к нормализованной форме, мы должны создать следующие таблицы: ПРОЕКТЫ, УЧАСТНИКИ, СВЯЗЬ проектов и участников.

Идентификатор проекта	Название проекта	Срок проекта (мес.)
1	Внедрение приложения	8
2	Открытие нового магазина	12

Идентификатор участника	Участник	Должность
1	Иванов И.И.	Программист
2	Сергеев С.С.	Бухгалтер
3	John Smith	Менеджер

Идентификатор проекта	Идентификатор участника
1	1
1	2
1	3
2	2
2	3

Теперь БД во 2 НФ!

3НФ	Отношение R находится в третьей нормальной форме (3НФ) в том и только в том случае, если находится в 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.
-----	---

**Транзитивная зависимость** – это когда неключевые столбцы зависят от значений других неключевых столбцов.

Если в первой нормальной форме наше внимание было нацелено на соблюдение реляционных принципов, во второй нормальной форме в центре нашего внимания был первичный ключ, то в третьей нормальной форме все наше внимание уделено столбцам, которые не являются первичным ключом, т.е. неключевым столбцам.

Чтобы нормализовать базу данных до третьей нормальной формы, необходимо сделать так, чтобы в таблицах отсутствовали неключевые столбцы, которые зависят от других неключевых столбцов.

Иными словами, неключевые столбцы не должны пытаться играть роль ключа в таблице, т.е. они действительно должны быть неключевыми столбцами, такие столбцы не дают возможности получить данные из других столбцов, они дают возможность посмотреть на информацию, которая в них содержится, так как в этом их назначение.

Пример.

Табельный номер	ФИО	Должность	Подразделение	Описание подразделения
1	Иванов И.И.	Программист	Отдел разработки	Разработка и сопровождение приложений и сайтов
2	Сергеев С.С.	Бухгалтер	Бухгалтерия	Ведение бухгалтерского и налогового учета финансово-хозяйственной деятельности
3	John Smith	Продавец	Отдел реализации	Организация сбыта продукции

Чтобы определить, находится ли эта таблица в третьей нормальной форме, мы должны проверить все неключевые столбцы, каждый из них должен зависеть только от первичного ключа, и никаким образом к другим неключевым столбцам он не должен относиться.

Однако, в результате проверки мы выясняем, что столбец «Описание подразделения» не зависит напрямую от первичного ключа. Мы это выяснили, когда задали себе один вопрос «Каким образом описание подразделения связано с сотрудником?». И наш ответ звучит следующим образом: «Атрибут описание подразделения содержит детальные сведения того подразделения, в котором работает сотрудник».

Отсюда следует, что столбец «Описание подразделения» не связан напрямую с сотрудником, он связан напрямую со столбцом «Подразделение», который напрямую связан с сотрудником, ведь сотрудник

работает в каком-то конкретном подразделении. Это и есть транзитивная зависимость, когда один неключевой столбец связан с первичным ключом через другой неключевой столбец.

Чтобы привести эту таблицу к третьей нормальной форме, мы должны сделать что? Правильно, декомпозицию!

Мы должны эту таблицу разбить на две: в первой хранить сотрудников, а во второй подразделения. А для реализации связи в таблице сотрудников создать ссылку на таблицу подразделений, т.е. добавить внешний ключ.

Табельный номер	ФИО	Должность	Подразделение
1	Иванов И.И.	Программист	1
2	Сергеев С.С.	Бухгалтер	2
3	John Smith	Продавец	3

Идентификатор подразделения	Подразделение	Описание подразделения
1	Отдел разработки	Разработка и сопровождение приложений и сайтов
2	Бухгалтерия	Ведение бухгалтерского и налогового учета финансово-хозяйственной деятельности
3	Отдел реализации	Организация сбыта продукции

## Требования нормальной формы Бойса-Кодда

Требования нормальной формы Бойса-Кодда следующие:

- Таблица должна находиться в третьей нормальной форме. Здесь все как обычно, т.е. как и у всех остальных нормальных форм, первое требование заключается в том, чтобы таблица находилась в предыдущей нормальной форме, в данном случае в третьей нормальной форме;
- Ключевые атрибуты составного ключа не должны зависеть от неключевых атрибутов.

Отсюда следует, что требования нормальной формы Бойса-Кодда предъявляются только к таблицам, у которых первичный ключ составной. Таблицы, у которых первичный ключ простой, и они находятся в третьей нормальной форме, автоматически находятся и в нормальной форме Бойса-Кодда.

4НФ	Отношение R находится в четвертой нормальной форме (4NF) в том и только в том случае, если в случае существования многозначной зависимости $A \twoheadrightarrow B$ все остальные атрибуты R функционально зависят от A.
-----	--

Требование четвертой нормальной формы (4NF) заключается в том, чтобы в таблицах отсутствовали нетривиальные многозначные зависимости. Начнем с того, что таблица должна иметь как минимум три столбца, допустим A, B и C, при этом B и C между собой никак не связаны и не зависят друг от друга, но по отдельности зависят от A, и для каждого значения A есть множество значений B, а также множество значений C. Если подобная многозначная зависимость есть в таблице, то она не соответствует четвертой нормальной форме.

5НФ	Отношение R находится в пятой нормальной форме (нормальной форме проекции-соединения - PJ/NF) в том и только в том случае, когда любая зависимость соединения в R следует из существования некоторого возможного ключа в R.
-----	---



Требование пятой нормальной формы (5NF) заключается в том, чтобы в таблице каждая нетривиальная зависимость соединения определялась потенциальным ключом этой таблицы.

До текущего момента, т.е. до 5 нормальной формы, мы осуществляли декомпозицию таблиц и не задумывались ни о какой потере данных, ведь у нас такой потери данных просто не было.

Однако существуют таблицы, которые не получится декомпонировать на две таблицы без потери данных, т.е. какие-то данные мы потеряем при соединении двух итоговых, полученных после декомпозиции, таблиц. Но, если декомпонировать такую таблицу не на две, а на три таблицы, то потери данных можно избежать.

И таблица будет находиться в пятой нормальной форме, если при соединении (JOIN) этих трех таблиц, которые были получены в результате декомпозиции, будут формироваться ровно те же самые данные, что и в исходной таблице до декомпозиции. Однако если этого происходить не будет, т.е. данные будут отличаться, например, какие-то строки были потеряны, или созданы новые, то в этом случае возникает так называемая зависимость соединения, т.е. часть данных одного столбца зависит от части данных другого столбца.

Таким образом, таблица будет находиться в пятой нормальной форме, если она не будет содержать зависимости соединения.

**Декомпозиция без потерь** – процесс разбиения одной таблицы на несколько, при условии, что в случае соединения таблиц, которые были получены в результате декомпозиции, будет формироваться ровно та же самая информация, что и в исходной таблице до декомпозиции.

### **3 этап – определение требований поддержки целостности данных**

Обычно ограничения целостности формулируются на естественном языке и/или с использованием предикатов, хотя в некоторых современных системах автоматизированного проектирования для этого имеются специализированные языковые средства.

Ограничения могут быть внутренними (неявными) и явными.

**Внутренние ограничения целостности** предусмотрены самой моделью данных и тесно связаны со структурой данных. С ограничениями этого типа хорошо согласуются операции манипулирования данными, поэтому контроль за соблюдением внутренних ограничений обычно не вызывает трудностей. Как правило, внутренние ограничения целостности задаются вместе с определением структурных спецификаций средствами языка описания данных.

**К внутренним ограничениям целостности** в нотациях IDEF1X можно отнести:

- ограничения на значения ключевых атрибутов: уникальность значений атрибутов первичных и альтернативных ключей определяется понятием ключа. Эти ограничения задаются соответствующим выделением ключей в модели;
- ограничения на допустимые значения атрибутов: все допустимые значения атрибутов должны удовлетворять условию принадлежности соответствующему домену. Эти ограничения задаются соответствующим описанием доменов и атрибутов. Кроме того, в описании атрибутов указывается обязательность значений определенных атрибутов; (например: СНИЛС обязательно состоит из 11 цифр, ФИО не может быть пустым и т.д.)
- ограничения на существующие значения (ссылочные ограничения): существование одних сущностей (дочерних, типа категория) ставится в зависимость от существования других (родительских, родовых). Эти ограничения представляются с помощью соответствующих связей между множествами сущностей.

**Явные ограничения целостности** задаются разработчиками.

Для описания явных ограничений используются исчисления предикатов и утверждения на естественном языке. Обеспечение контроля явных ограничений целостности представляет собой серьезную проблему реализации информационной системы.

Автоматическая поддержка всех видов ограничений целостности возможна за счет использования операторов SQL. Ссылочная целостность может быть обеспечена за счет использования триггеров.

## Использование методологии IDEF1X для разработки концептуальной модели данных

Методология IDEF1X - один из подходов к семантическому моделированию данных, основанный на концепции "сущность-связь" (Entity-Relationship). Это инструмент для анализа информационной структуры систем различной природы. Информационная модель, построенная с помощью IDEF1X -методологии, отображает логическую структуру информации об объектах системы.

Сущность изображается на ER-диаграмме в виде прямоугольника, в верхней части которого приводится ее название; далее следует список атрибутов. Ключевые атрибуты могут быть выделены подчеркиванием или иным способом.

Стандарт IDEF1X описывает способы изображения двух типов сущностей - независимой и зависимой, и связей - идентифицирующих и неидентифицирующих.

Каждая сущность может обладать любым количеством связей с другими сущностями.

Сущность является **независимой**, если каждый ее экземпляр может быть однозначно идентифицирован без определения его связей с другими сущностями.

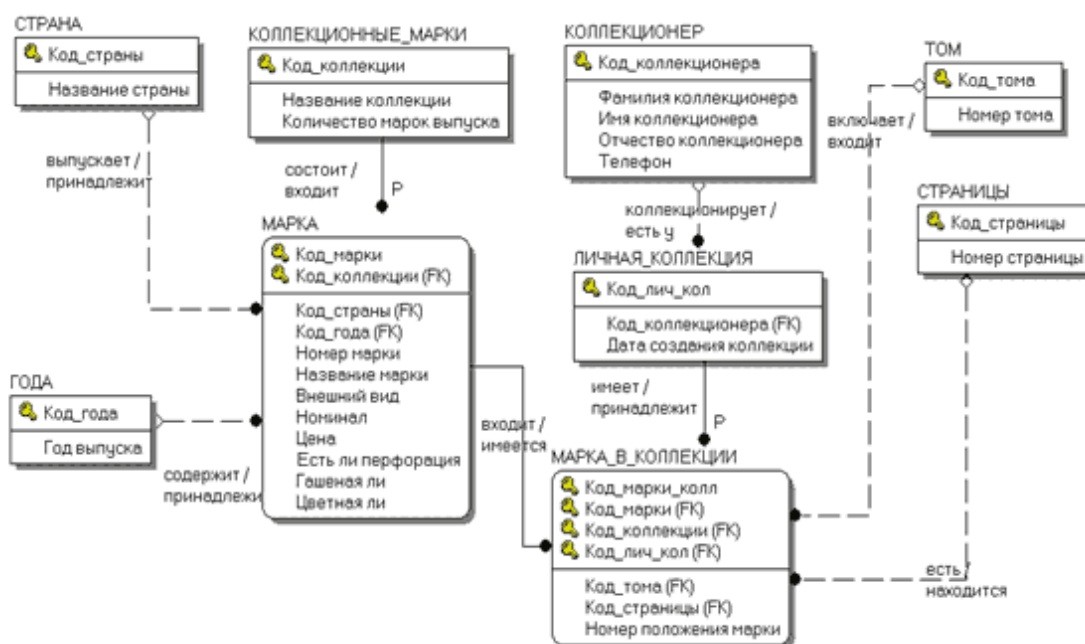
Сущность называется **зависимой**, если однозначная идентификация ее экземпляра зависит от его связей с другими сущностями.

Сущность может обладать атрибутами, которые наследуются через связь с родительской сущностью. Последние обычно являются внешними ключами (FK на рис) и служат для организации связей между сущностями. Если внешний ключ сущности используется в качестве ее первичного ключа (PK) или как часть составного первичного ключа, то сущность является зависимой от родительской сущности. Если внешний ключ не является первичным и не входит в составной первичный ключ, то сущность является независимой от родительской сущности.

Если сущность является зависимой, то связь ее с родительской сущностью называется **идентифицирующей**, в противном случае - **неидентифицирующей**.

Связь изображается на ER-диаграмме линией, проводимой между сущностью-родителем и сущностью-потомком с точкой на конце линии у сущности-потомка. идентифицирующая связь изображается сплошной линией, неидентифицирующая - пунктирной.

Связи дается имя, выражаемое грамматической формой глагола. Для связи дополнительно может присутствовать указание мощности: какое количество экземпляров сущности-потомка может существовать для сущности-родителя. Имя связи всегда формируется с точки зрения родителя, так что может быть образовано предложение, если соединить имя сущности родителя, имя связи, выражение мощности и имя сущности-потомка (например "много СТУДЕНТов - сдают - ЭКЗАМЕН").



Принципы изображения концептуальных моделей баз данных стандарта IDEF1 и IDEF1X используют CASE Studio и другие CASE-средства (PGadmin4 и др). Подобные системы позволяют на основе

концептуальной модели генерировать физическую модель и программный код создания базы данных для большинства наиболее распространенных СУБД и серверов баз данных.

### Ход работы.

Выберите вариант задания, который соответствует вашему порядковому номеру в журнале преподавателя (если порядковый номер превышает 10, то выполняете вариант, соответствующий последней цифре порядкового номера, то есть если ваш номер 12, то выполняете 2 вариант)

#### ЗАДАНИЕ 1.

Для указанного варианта задания выполнить нормализацию отношений методом декомпозиции с учетом выделенных функциональных зависимостей. Представить результат в виде нормализованной реляционной модели. ДОКАЗАТЬ, что БД находится в 3 НФ!

##### 1 вариант

Номер судна	Название	Номер рейса	Дата погрузки	Порт погрузки	Дата прибытия	Порт прибытия	Ф.И.О. капитана	Вид судна	Грузо подъемность, тонны
526	Japan Bear	9201W	5/31/92	SFO	6/6/92	HNL	Емелин А.О.	Сухогруз	500
603	Korea Bear	9202W	5/05/92	OAK	6/19/92	OSA	Крылов О.Б.	Ролкер	1000
531	China Bear	9203W	6/20/92	LAX	7/10/92	PAP	Мухин Е.А.	Универсал	1500
526	Japan Bear	9204W	8/20/92	SFO	8/27/92	HNL	Емелин А.О.	Сухогруз	500

##### 2 вариант

Номер проекта	Наименование проекта	Номер работника	Ф.И.О. работника	Должность	руб/час	трудозатраты в часах	Общие расходы
15	Alpha Edit	101	Семен Иванов	Программист	200	120	24000
		102	Андрей Петров	Программист	200	100	20000
		110	Антон Сидоров*	Сист. аналитик	300	40	12000
18	Beta Base	103	Федот Антонов	Программист	200	250	50000
		102	Андрей Петров	Программист	200	280	56000
		111	Петр Семенов*	Проектировщик БД	250	80	20000
22	Delta CAD	104	Сидор Федотов	Программист	200	180	36000
		105	Иван Андреев	Программист	200	150	30000
		110	Антон Сидоров*	Системный аналитик	300	60	18000

##### 3 вариант

Наименование контрагента	Город	Адрес	Вид контрагента	Должность контактного лица	Ф.И.О. контактного лица	Код города	Телефон
Поршневой завод	Владимир	ул. Кольцевая, 17	Поставщик	зам. дир.	Иванов И.И.	3254	76-15-95
Поршневой завод	Владимир	ул. Кольцевая, 17	Поставщик	нач. отд. сбыта	Петров П.П.	3254	76-15-35
ООО «Вымпел»	Курск	ул. Гоголя, 25	Клиент, Поставщик	директор	Сидоров С.С.	7634	66-65-38
ИП «Альфа»	Владимир	ул. Пушкинская, 37	Клиент, Поставщик	директор	Васильев В.В.	3254	74-57-45

#### 4 вариант

Модель	Цвет	Коробка передач	Обивка	Цена (\$)	№ заказа	Фамилия заказчика	Телефон	Дата заказа
12579 Classic liftback XL	Красный	Ручная	Ткань	35700	123	Иванов Федор Степанович	2859655	3.08.97
12580 Classic liftback GT	Черный	Автоматическая	Ткань	39200	130	Петров Олег Иванович	3856743	6.10.97
12651 Classic Compact XL	Антрацит	Ручная	Кожа	41100	133	Сидоров Николай Сергеевич	342679	25.12.97
12653 Classic Compact GT	Черный	Ручная	Велюр	37900	135	Бендер Остап Ибрагимович	56438	5.01.98
12410 Classic Combi	Антрацит	Автоматическая	Ткань	46200	138	Иванов Сергей Сергеевич	2859655	20.02.98
12653 Classic Compact GT	Черный	Ручная	Велюр	37900	140	Петров Юрий Андреевич	3856743	30.06.98
12410 Classic Combi	Антрацит	Автоматическая	Ткань	46200	145	Сидоров Борис Борисович	342679	25.08.98

#### 5 вариант

Код и название маршрута	Продолжительность	Стоимость	Вид транспорта	Наличие мест
01 С.-Петербург	14	2400	001 Теплоход	✓
02 Ярославль	6	1400	001 Теплоход	✓
03 Кострома	7	1600	001 Теплоход	□
04 Волгоград	14	2600	001 Теплоход	✓
05 Астрахань	21	3300	001 Теплоход	✓
06 С.-Петербург	14	2000	002 ЖД	✓
07 Ярославль	6	1000	002 ЖД	✓
08 Кострома	7	1000	002 ЖД	□
09 Волгоград	14	2100	002 ЖД	✓
10 Астрахань	21	3000	002 ЖД	✓
11 С.-Петербург	12	3400	003 Авиа	✓
12 Ярославль	4	2400	003 Авиа	□

#### 6 вариант

Код товара и название	Марка товара	Цена (\$)	Код и название фирмы - поставщика	Телефон	Адрес
1 Телевизор	GoldStar CM - 2180K	459	1010 М.Видео	(095) 2079464	Маросейка, 6/8
2 Телевизор	Philips 25PT9001	1499	1020 Диал Электронике	(095) 9780443	Новослободская, 14/19
3 Телевизор	Panasonic 25V50R	765	1030 Мир	(095) 1524001	Чонгарский б-р, 16
4 Телевизор	GoldStar CF - 14E20B	230	1010 М.Видео	(095) 2079464	Маросейка, 6/8
5 Видеомагнитофон	Panasonic HS - 800EE	1400	1020 Диал Электроникс	(095) 9780443	Новослободская, 14/19
6 Видеомагнитофон	Philips VR - 756	450	1030 Мир	(095) 1524001	Чонгарский б-р, 16
7 Видеокамера	Samsung VP - J55	530	1010 М.Видео	(095) 2079464	Маросейка, 6/8
8 Видеокамера	Sharp E37	845	1020 Диал Электроникс	(095) 9780443	Новослободская, 14/19
9 Музыкальный центр	Panasonic DH32	320	1030 Мир	(095) 1524001	Чонгарский б-р, 16



## 7 вариант

Фамилия Имя Отчество	Шифр	Телефон	Адрес	Залог	Название диска	Жанр	Шифр
Захаров Петр Сергеевич	ЗПС56	12-34-56	ул. Астахова д.2	360	Diablo-2	RPG	D001
Жуков Игорь Иванович	ЖИИ21	65-43-21	ул. Борисова д.4 кв.3	300	Space War-4	Strategy	S001
Петров Федор Ильич	ПФИ00	нет	ул. Макаренк о д.5 кв. 1	300	Space Quest 9	Adventure	S002
Захаров Петр Сергеевич	ЗПС56	12-34-56	ул. Астахова д.2	360	Space Quest 9	Adventure	S002

## 8 вариант

№ студенческого	Фамилия Имя Отчество студента	Код группы	Классный руководитель	Должность	Педстаж	Аудитория
12345	Егоров Иван Андреевич	ПО- 21к	Иванова Тамара Сергеевна	преподаватель	3	222
12346	Крылов Павел Павлович	ПО- 21к	Иванова Тамара Сергеевна	преподаватель	3	222
12347	Тимофеев Олег Иванович	ТСВТ-22к	Сергеева Антонина Юрьевна	методист	5	232
12348	Ахмедов Аслан Мухамедович	ТСВТ-22к	Сергеева Антонина Юрьевна	методист	5	232
12349	Ковалева Антонина Ивановна	ТСВТ-22к	Сергеева Антонина Юрьевна	методист	5	232
12350	Труфанов Илья Егорович	ТСВТ-33к	Пузырев Антон Юрьевич	Начальник ВЦ	4	ВЦ2
12351	Ольгин Олег Юрьевич	ТСВТ-33к	Пузырев Антон Юрьевич	Начальник ВЦ	4	ВЦ2
12352	Лескова Дарья Игоревна	ТСВТ-33к	Пузырев Антон Юрьевич	Начальник ВЦ	4	ВЦ2
12353	Петров Юрий Егорович	ТСВТ-33к	Пузырев Антон Юрьевич	Начальник ВЦ	4	ВЦ2

## 9 вариант

Код и ФИО владельца	Животное	Порода	Кличка	Дата посещения	Диагноз	Лечение
001 Павлов В.Е.	Енот	Енот-полоскун	Петька	18.09.05	Перелом передней правой лапы	Гипс
002 Николаев А.А.	Кот	Персидский	Веня	13.10.05	Конъюктивит	Глазные капли Тауфон, промывание глаз
003 Краснов Е.А.	Кот	Ангорский	Пенка	14.10.05	Лишай	Мазь тетрациклиновая
004 Сидоров У.К.	Канарейка	-	Люся	15.11.05	Потеря аппетита	Удаление опухоли горла
005 Лядов О.Ю.	Собака	Дворяжка	Бим	20.12.05	Вывих задней левой лапы	Лапа вправлена, одет временный фиксатор

## 10 вариант

№ документа	Тип документа	Ф.И.О. читателя	Название книги	Автор	Жанр	Цена (руб.)	Дата выдачи
123	Студенческий билет	Авдеева Екатерина Ивановна	20000 лье под водой	Жюль Верн	Приключения	500	01.09.06
123	Пропуск	Лынова Елена Олеговна	Биология для высших учебных заведений	Право Л.Л.	Учебник	200	02.09.06
125	Студенческий билет	Троев Антон Антонович	Три мушкетера	А.Дюма	Приключения	220	03.09.06
111	Пропуск	Кононова Алла Ивановна	Office XP . Разработка приложений	Новиков Ф.	Техническая литература	790	04.09.06
122	Студенческий билет	Краснов Петр Егорович	Компьютерра	-	Журнал	35	04.09.06

## ЗАДАНИЕ 2.

Нормализованную БД из предыдущего задания представить в виде модели в нотификации IDEF1X, используя любое case-средство или графический редактор.

## ЗАДАНИЕ 3.

1. Нормализовать БД из примера (про аэропорт) из ЛР №2, доказать (письменно), что БД находится в 3 НФ.
2. Нормализованную БД из предыдущего задания представить в виде модели в нотификации IDEF1X, используя любое case-средство или графический редактор.

## ЗАДАНИЕ 4.

Создать ER- модель в нотации IDEF1X в любом case-средстве на выбор из предложенных (PGAdmin4, MS Visio, MySQL Workbench, SQL Server Enterprise Manager ). БД своего варианта использовать из ЛР №2.

### Р.С. Пример на pgAdmin4

Сначала необходимо установить БД, например PostgreSQL 12.20 (или новее), дистрибутив возьмите с сайта: <https://www.postgresql.org/download/>

Запустить скачанный файл. Сначала инсталлятор проверит наличие всех необходимых компонентов, в частности Visual C++ Redistributable, в случае необходимости, т.е. их отсутствия, он их сам установит.

После этого откроется окно приветствия, нажать «Next».

Затем нужно указать путь к каталогу, в который нужно установить PostgreSQL, но можно оставить и по умолчанию. В случае необходимости указать путь и нажать «Next»

На следующем шаге можно отметить компоненты, которые необходимо установить. В числе компонентов есть и pgAdmin 4. Оставить галочки напротив нужных компонентов и нажать «Next» (оставьте все галочки).

Далее необходимо указать каталог, в котором по умолчанию будут располагаться файлы баз данных. В данном случае лучше указать отдельный диск. Нажать «Next».

Теперь нужно задать пароль для пользователя postgres, иными словами, для администратора PostgreSQL Server. Ввести пароль и подтвердить его. Нажать «Next». **ПАРОЛЬ НУЖНО ЗАПОМНИТЬ!**

Далее в случае необходимости можно изменить порт, на котором будет работать PostgreSQL Server, но можно оставить и по умолчанию. Нажать «Next».

Если есть необходимость указать конкретную кодировку данных в базе, можно выбрать из выпадающего списка Locale. По умолчанию нажать «Next».

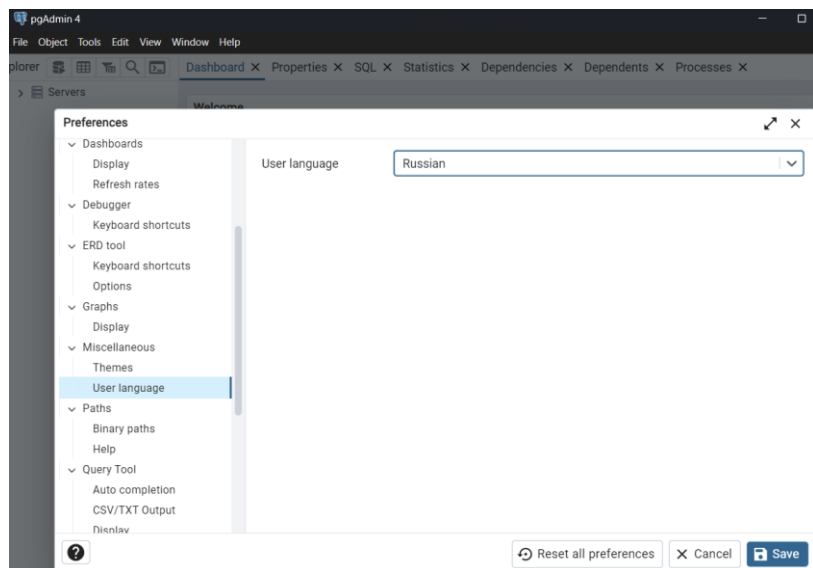
Проверить введенные ранее параметры для установки PostgreSQL, если все правильно, т.е. все то, что введено, нажать «Next» - «Next».

Когда появится окно с сообщением «Completing the PostgreSQL Setup Wizard» установка PostgreSQL, pgAdmin 4 и других компонентов будет завершена.

В последнем окне будет предложено запустить Stack Builder для загрузки и установки дополнительных компонентов. Если ничего не нужно, то снять галочку «Launch Stack Builder at exit?» и нажать «Finish».

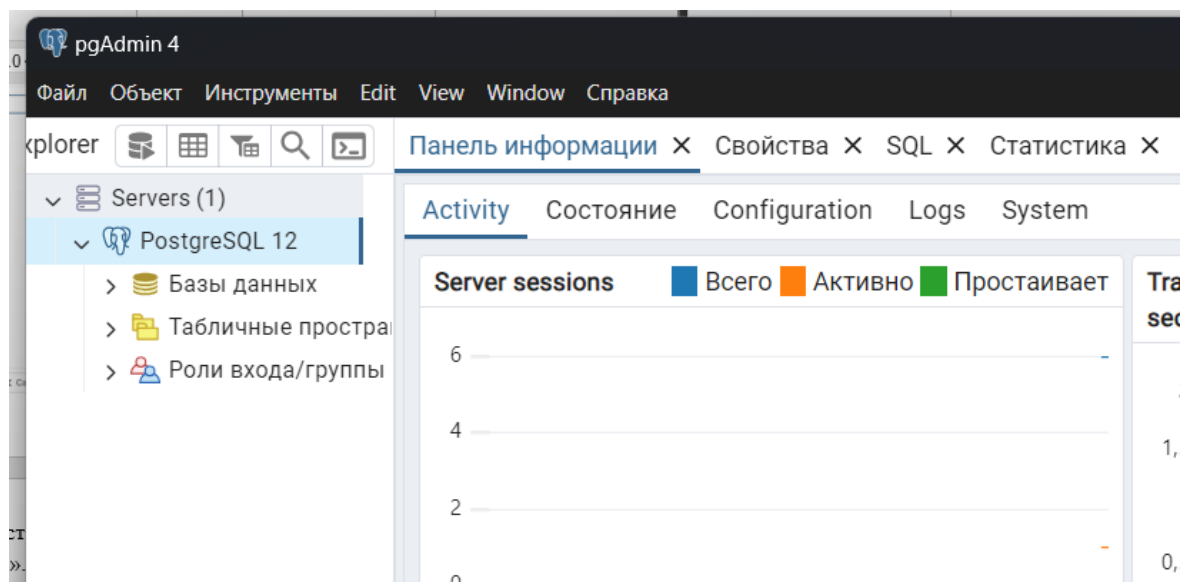
pgAdmin 4 установился вместе PostgreSQL. Для того чтобы запустить pgAdmin 4, нажать «Меню Пуск -> PostgreSQL 1X -> pgAdmin 4».

Далее если вам нужно поменять тему или язык (либо что-то ещё) можно зайти в настройки. На рисунке ниже представлено, где найти смену языка.



Чтобы осуществить подключение к только что установленному локальному серверу PostgreSQL 1X в обозревателе серверов (колонка слева), выбрать пункт «PostgreSQL 1X».

В результате запустится окно «Connect to Server», в котором Вам нужно ввести пароль системного пользователя postgres, т.е. это тот пароль, который придуман при установке PostgreSQL. Ввести пароль, поставить галочку «Save Password», для того чтобы сохранить пароль, и каждый раз не вводить его, и нажать «ОК».



В итоге выполнено подключение к локальному серверу PostgreSQL.

Для определения схемы базы данных используется диалоговое окно Schema (Схема). Схемы – это организационная рабочая область базы данных, похожая на каталоги или пространства имен. Чтобы создать схему, вы должны быть суперпользователем базы данных или иметь привилегию CREATE.

Schema-диалог организует разработку схемы с помощью следующих диалоговых закладок: General и Security . На вкладке SQL отображается код SQL, созданный в диалоговых окнах.

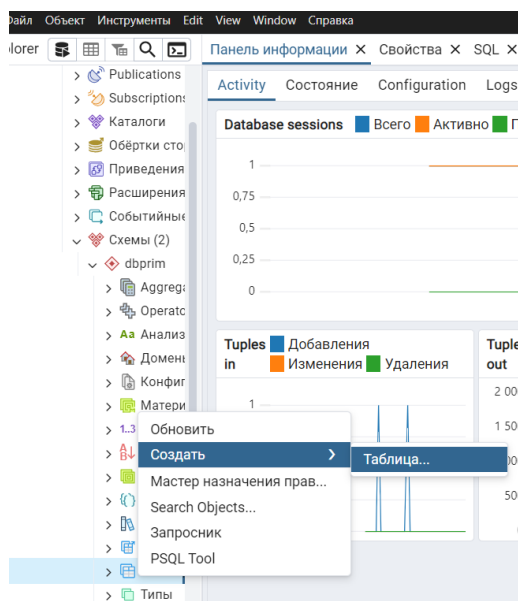
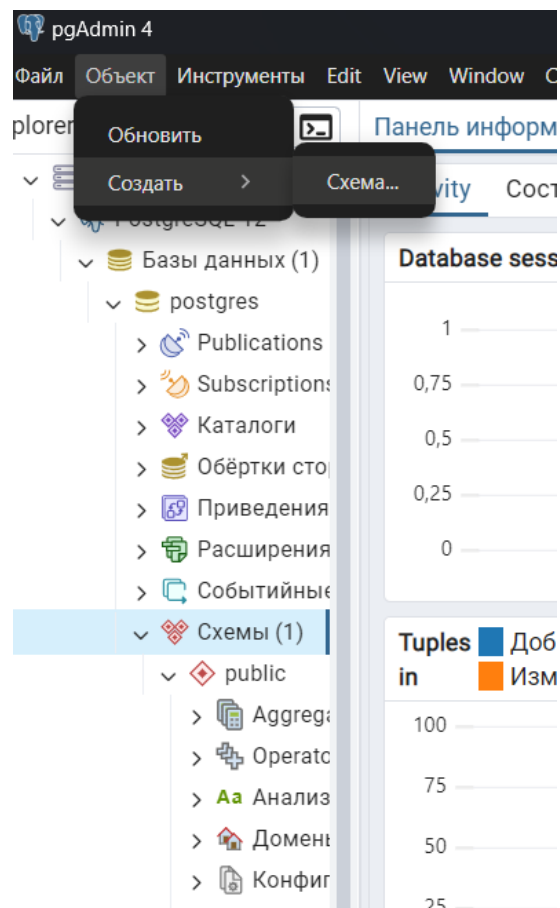
Для идентификации схемы используются поля на вкладке Общие:

- поле «Имя» используется, чтобы добавить описательное имя для схемы. Имя будет отображаться в древовидном элементе управления pgAdmin;
- выбрать владельца схемы из раскрывающегося списка в поле «Владелец»;
- хранить заметки о схеме в поле Комментарий.

pgAdmin 4 предоставляет диалоги, которые позволяют изменять все свойства и атрибуты таблицы.

Используйте диалог Таблица для создания или изменения таблицы.

Диалоговое окно «Таблица » организует разработку таблицы с помощью следующих диалоговых вкладок: «Общие» , «Столбцы» , «Ограничения» , «Дополнительно» , «Параметр» и «Безопасность» . На вкладке SQL отображается код SQL, созданный в диалоговых окнах.



На вкладке Общие задайте имя таблицы, на вкладке Столбцы добавьте имя столбца (нажать на + в правом углу), тип данных, может ли иметь нулевое значение, обозначьте первичный ключ. Задайте оставшиеся атрибуты.



Создание Таблица

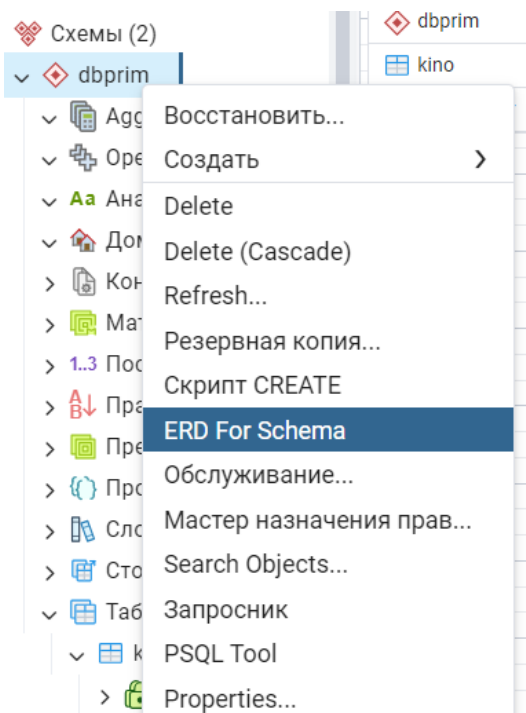
Общие Столбцы Дополнительно Ограничения Секции Параметры Безопасность SQL

Наследуется из таблиц(ы) Выберите источник наследования...

Столбцы		Имя	Тип данных	Length/Precision	Масшт...	Не NULL?	Первичный к...	По умолч...
		id_f	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Добавьте все таблицы, которые есть в вашей схеме.

Для визуального отображения схемы выполните команду ERD for schema



Добавьте связи в таблицу через внешние ключи:

1. На схеме нажмите редактировать таблицу
2. Зайдите Ограничения – внешний ключ, добавьте внешний ключ (заполните локальный столбец, ссылается на, зависимый столбец - Add).

Table: kino (dbprim) ✕

Общие Столбцы Дополнительно Ограничения

Первичный ключ Внешний ключ Уникальность

✎ 🗑

Имя	Столбцы	Referenced Table
<input type="text"/>		

Общие Определение Столбцы Действие

Столбцы

Локальный столбец

id\_janr ✕ | ▾

Ссылается на

(dbprim) ganr ✕ | ▾

Зависимый столбец

ID\_janr ✕ | ▾

Add

Локальный	Ссылается на	Referenced Table

⚠ Выберите столбцы для внешнего ключа.

✕

✕ Закрыть

🔄 Сбросить

💾 Сохранить

3. Сохраните и убедитесь, что связь на схеме появилась.

#### Критерии оценки:

<b>Зачтено</b>	Все задания лабораторной работы самостоятельно, студент может объяснить, как он выполнил то или иное действие; отчет содержит все необходимые сведения, студент уверенно отвечает на вопросы по теории
<b>Не зачтено</b>	Лабораторная работа выполнена не самостоятельно (студент не может ответить ни на один вопрос преподавателя), или выполнена не до конца, или не выполнена совсем