

Контрольная работа 4

Логическое проектирование реляционной БД. Часть 2.

Уважаемый, студент!
Вам необходимо:

1. Провести нормализацию полученных реляционных отношений (до 4НФ)
2. Определить дополнительные ограничения целостности
3. Описать группы пользователей и права доступа

На этапе логического проектирования – ER-диаграмма формальным способом преобразуется в схему реляционной базы данных (РБД). На основании схемы РБД и описания сущностей ПрО составляются отношения (таблицы) базы данных (контрольная работа 3). Потом выполняется нормализация отношений. Это необходимо сделать для того, чтобы исключить нарушения логической целостности данных и повысить таким образом надёжность и достоверность данных. В отдельных случаях после нормализации может выполняться денормализация, но причина для этого может быть только одна: повышение эффективности выполнения критических запросов.

В результате всех этих операций создаётся концептуальная схема БД – основной документ для базы данных.

Немного теории

Проектирование реляционной базы данных проходит в том же порядке, что и проектирование БД других моделей данных, но имеет свои особенности.

Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных. Они обусловлены отсутствием средств явного представления типов множественных связей между объектами ПрО и неразвитостью средств описания ограничений целостности на уровне модели данных.

Для решения подобных проблем проводится **нормализация отношений**. Механизм нормализации реляционных отношений разработал Э.Ф. Кодд (E.F. Codd). Этот механизм позволяет по формальным признакам любое отношение преобразовать к третьей нормальной форме.

Проектирование реляционной базы данных проходит в том же порядке, что и проектирование БД других моделей данных, но имеет свои особенности. Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных. Они обусловлены отсутствием средств явного представления типов множественных связей между объектами ПрО и неразвитостью средств описания ограничений целостности на уровне модели данных. Для решения подобных проблем проводится нормализация отношений. Механизм нормализации реляционных отношений разработал Э.Ф. Кодд (E.F. Codd). Этот механизм позволяет по формальным признакам любое отношение преобразовать к третьей нормальной форме.

Нормализация схемы отношения выполняется путём декомпозиции схемы. **Декомпозицией** схемы отношения R называется замена её совокупностью схем отношений A_i таких, что

$$R = \bigcup_i A_i$$

и не требуется, чтобы отношения A_i были непересекающимися.

Первая нормальная форма относится к понятию простого и сложного (составного или многозначного) атрибута.

Первая нормальная форма (1НФ).

Отношение приведено к 1НФ, если все его атрибуты простые.

Для того чтобы привести к 1НФ отношение, содержащее сложные атрибуты, нужно:

- 1) разбить составные атрибуты на простые,
- 2) построить декартово произведение всех многозначных атрибутов с кортежами, к которым они относятся.

Для идентификации кортежа в этом случае понадобится составной ключ, включающий первичный ключ исходного отношения и все многозначные атрибуты.

Вторая нормальная форма основана на понятии *функциональной зависимости*. Пусть X и Y – атрибуты некоторого отношения. Если в любой момент времени каждому значению X соответствует единственное значение Y , то говорят, что Y функционально зависит от X ($X \rightarrow Y$). Атрибут X в функциональной зависимости $X \rightarrow Y$ называется *детерминантом* отношения.

В нормализованном отношении все неключевые атрибуты функционально зависят от ключа отношения. Неключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части составного ключа.

Вторая нормальная форма (2НФ).

Отношение находится во 2НФ, если оно приведено к 1НФ и каждый неключевой атрибут функционально полно зависит от составного первичного ключа. (Таким образом, если отношение в 1НФ имеет простой первичный ключ, оно сразу находится во второй нормальной форме).

Для того чтобы привести отношение к 2НФ, нужно: \square

- построить его проекцию, исключив атрибуты, которые не находятся в функционально полной зависимости от составного первичного ключа; \square
- построить дополнительно одну или несколько проекций на часть составного ключа и атрибуты, функционально зависящие от этой части ключа.

Третья нормальная форма основана на понятии *транзитивной зависимости*. Пусть X , Y , Z – атрибуты некоторого отношения. При этом $X \rightarrow Y$ и $Y \rightarrow Z$, но обратное соответствие отсутствует, т.е. Z не зависит от Y или Y не зависит от X . Тогда говорят, что Z транзитивно зависит от X ($X \rightarrow \rightarrow Z$).

Третья нормальная форма (3НФ).

Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Для того чтобы привести отношение к 3НФ, нужно: \square

- построить проекцию, исключив транзитивно зависящие от ключа атрибуты;
- построить дополнительно одну или несколько проекций на детерминанты исходного отношения и атрибуты, функционально зависящие от них.

Исключение составляют случаи, когда для транзитивной зависимости $X \rightarrow \rightarrow Z$ ($X \rightarrow Y$ и $Y \rightarrow Z$) либо Z зависит от Y , либо Y зависит от X , т.е. между атрибутами X и Y , например, существует связь 1:1. В такой ситуации декомпозиция отношения не производится.

Четвертая нормальная форма основана на понятии *многозначной зависимости*. Многозначная зависимость существует, если заданным значениям атрибута X соответствует множество, состоящее из нуля (или более) значений атрибута Y ($X \twoheadrightarrow Y$).

Различают тривиальные и нетривиальные многозначные зависимости. *Тривиальной* называется такая многозначная зависимость $X \twoheadrightarrow Y$, для которой $Y \subseteq X$ или $X \cup Y = R$, где R – рассматриваемое отношение. Тривиальная многозначная зависимость не нарушает 4НФ. Если хотя бы одно из двух этих условий не выполняется, то такая зависимость называется *нетривиальной*.

Четвертая нормальная форма (4НФ).

Отношение находится в 4НФ, если оно находится в 3НФ и в нём отсутствуют нетривиальные многозначные зависимости.

Для того чтобы привести отношение к 4НФ, нужно построить две или более проекции исходного отношения, каждая из которых содержит ключ и одну из многозначных зависимостей.

Пример выполнения заданий (продолжение той же БД, что и в КР2)

1. Нормализация полученных отношений (до 4НФ)

Механизм нормализации подразумевает определённую последовательность преобразования отношений к третьей нормальной форме. Мы не будем чётко придерживаться этой последовательности, т.к. она избыточна, и многозначные атрибуты сразу вынесем в отдельные отношения на первом же этапе. **1НФ.** Для приведения таблиц к 1НФ требуется составить прямоугольные таблицы (одно значение атрибута – одна ячейка таблицы) и разбить сложные атрибуты на простые.

Примечание. В реальных БД сложные атрибуты разбиваются на простые, если: а) этого требует внешнее представление данных; б) в запросах поиск может осуществляться по отдельной части атрибута.

Разделим атрибут Фамилия, имя, отчество на два атрибута Фамилия и Имя, отчество, Паспортные данные на Номер паспорта (уникальный), Дата выдачи и Кем выдан, а Данные об образовании – на Вид образования, Специальность, Номер диплома и Год окончания учебного заведения.

Многозначные атрибуты Комнаты и Телефоны из отношения ОТДЕЛЫ вынесем в отдельное отношение КОМНАТЫ, а домашние и мобильные телефоны и адреса сотрудников – в отношение АДРЕСА-ТЕЛЕФОНЫ. Так как в комнате может не быть телефона, первичный ключ отношения КОМНАТЫ не определен (ПК не может содержать null-значения), но на этих атрибутах можно определить составной уникальный ключ. В отношении АДРЕСА-ТЕЛЕФОНЫ также нет потенциальных ключей: оставим это отношение без первичного ключа, т.к. на это отношение никто не ссылается. Данные об образовании сотрудников также вынесем в отдельное отношение.

Что касается рабочих телефонов сотрудников, то один из этих номеров – основной – определяется рабочим местом сотрудника (рассматриваются только стационарные телефоны). Будем хранить этот номер в атрибуте Рабочий телефон. Наличие других номеров зависит от того, есть ли в том же помещении (комнате) другие сотрудники, имеющие стационарные телефоны. Добавим в отношение СОТРУДНИКИ атрибут Номер комнаты, чтобы дополнительные номера телефонов сотрудника можно было вычислить из других кортежей с таким же номером комнаты.

Связь между отношениями СОТРУДНИКИ и КОМНАТЫ реализуем через составной внешний ключ (Номер комнаты, Рабочий телефон).

Мы также удалим вычисляемый атрибут Полученная сумма из отношения ПРОЕКТЫ, т.к. он является суммой значений аналогичного атрибута из отношения ЭТАПЫ ПРОЕКТОВ. Но атрибут Стоимость проекта оставим, т.к. она фигурирует в документации по проекту. А для обеспечения логической целостности данных предусмотрим в приложении проверку того, что сумма по всем этапам совпадает со стоимостью проекта.

2НФ. В нашем случае составные первичные ключи имеют отношения ЭТАПЫ ПРОЕКТА и УЧАСТИЕ. Неключевые атрибуты этих отношений функционально полностью зависят от составных первичных ключей.

3НФ. В отношении ПРОЕКТЫ атрибут Данные заказчика зависит от атрибута Заказчик, а не от первичного ключа, поэтому его следует вынести в отдельное отношение ЗАКАЗЧИКИ. Но при этом первичным ключом нового отношения станет атрибут Заказчик,

т.е. длинная символьная строка. Целесообразнее перенести в новое отношение атрибуты Заказчик и Данные заказчика и ввести для него суррогатный ПК. Так как с каждым заказчиком может быть связано несколько проектов, связь между отношениями ПРОЕКТЫ и ЗАКАЗЧИКИ будет 1:n и суррогатный ПК станет внешним ключом для отношения ПРОЕКТЫ.

В отношении СОТРУДНИКИ атрибут Оклад зависит от атрибута Должность. Поступим с этой транзитивной зависимостью так же, как в предыдущем случае: создадим отношение ДОЛЖНОСТИ, перенесём в него атрибуты Должность и Оклад, а первичным ключом сделаем название должности. В отношениях СОТРУДНИКИ и ОБРАЗОВАНИЕ атрибуты (Дата выдачи и Кем выдан) и (Номер диплома и Год окончания учебного заведения) зависят не от первичного ключа, а от атрибутов соответственно Номер паспорта и Специальность. Но если мы выделим их в отдельное отношение, то получим связи типа 1:1. Следовательно, здесь декомпозиция нецелесообразна.

4НФ. Отношение АДРЕСА-ТЕЛЕФОНЫ нарушают 4НФ, т.к. не всякий телефон привязан к конкретному адресу (т.е. мы имеем две многозначных зависимости в одном отношении). Но выделять Телефоны в отдельное отношение не стоит, т.к. эти сведения носят справочный характер и не требуется их автоматическая обработка.

Отношения, полученные после нормализации, приведены в табл. 1-10.

Таблица 1. Схема отношения ОТДЕЛЫ (Departs)

Содержание поля	Имя поля	Тип, длина	Примечания
Аббревиатура отдела	D_ID	C(10)	первичный ключ
Название отдела	D_NAME	V(100)	обязательное поле

Таблица 2. Схема отношения КОМНАТЫ (Rooms)

Содержание поля	Имя поля	Тип, длина	Примечания
Отдел	R_DEPART	V(12)	внешний ключ (к Departs)
Номер комнаты	R_ROOM	N(4)	составной уникальный ключ
Телефоны	R_PHONE	V(20)	

Таблица 3. Схема отношения ДОЛЖНОСТИ (Posts)

Содержание поля	Имя поля	Тип, длина	Примечания
Название должности	P_POST	V(30)	первичный ключ
Оклад	P_SAL	N(8,2)	обязательное поле, > 12130 руб.

Таблица 4. Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор сотрудника	E_ID	N(4)	суррогатный первичный ключ
Фамилия	E_NAME	V(25)	обязательное поле
Имя, отчество	E_LNAME	V(30)	обязательное поле
Дата рождения	E_BORN	D	обязательное поле
Пол	E_SEX	C(1)	обязательное поле
Серия и номер паспорта	E_PASP	C(10)	обязательное уникальное поле
Когда выдан паспорт	E_DATE	D	обязательное поле
Кем выдан паспорт	E_PASP	V(50)	обязательное поле
ИНН	E_INN	C(12)	обязательное поле
Номер пенсионного страхового свидетельства	E_PENS	C(14)	обязательное уникальное поле
Отдел	E_DEPART	V(12)	внешний ключ (к Departs)

Должность	E_POST	V(30)	внешний ключ (к Posts)
Номер комнаты	E_ROOM	N(4)	составной внешний ключ (к Rooms)
Рабочий телефоны	E_PHONE	V(20)	
Логин	E_LOGIN	V(30)	

Таблица 5. Схема отношения ОБРАЗОВАНИЕ (Edu)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор сотрудника	U_ID	N(4)	внешний ключ (к Employees)
Вид образования	U_TYPE	V(20)	обязательное поле
Специальность	U_SPEC	V(40)	
Номер диплома	U_DIPLOM	V(15)	
Год окончания учебного заведения	U_YEAR	N(4)	обязательное поле

Таблица 6. Схема отношения АДРЕСА-ТЕЛЕФОНЫ (AdrTel)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор сотрудника	A_ID	N(4)	внешний ключ (к Employees)
Адрес	A_ADDR	V(50)	обязательное поле
Телефон	A_PHONE	V(30)	

Таблицы ОБРАЗОВАНИЕ и АДРЕСА-ТЕЛЕФОНЫ не имеют потенциальных ключей, но мы не будем вводить суррогатные первичные ключи, т.к. на эти таблицы никто не ссылается.

Таблица 7. Схема отношения ЗАКАЗЧИКИ (Clients)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер заказчика	C_ID	N(4)	суррогатный первичный ключ
Заказчик	C_COMPANY	V(40)	обязательное поле
Адрес заказчика	C_ADR	V(50)	обязательное поле
Контактное лицо	C_PERSON	V(50)	обязательное поле
Телефон	C_PHONE	V(30)	

Таблица 8. Схема отношения ПРОЕКТЫ (Projects)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер проекта	P_ID	N(6)	обязательное уникальное поле
Название проекта	P_TITLE	V(100)	обязательное поле
Сокращённое название	P_ABBR	C(10)	первичный ключ
Отдел	P_DEPART	V(12)	внешний ключ (к Departs)
Заказчик	P_COMPANY	N(4)	внешний ключ (к Clients)
Руководитель	P_CHIEF	N(4)	внешний ключ (к Employees)
Дата начала проекта	P_BEGIN	D	обязательное поле
Дата окончания проекта	P_END	D	обязательное поле, больше даты начала проекта
Реальная дата окончания	P_FINISH	D	больше даты начала проекта
Стоимость проекта	P_COST	N(10)	обязательное поле, > 0

Таблица 9. Схема отношения ЭТАПЫ ПРОЕКТА (Stages)

Содержание поля	Имя поля	Тип, длина	Примечания	
Проект	S_PRO	C(10)	внешний ключ (к Projects)	составной первичный ключ
Номер этапа	S_NUM	N(2)		
Название этапа	S_TITLE	V(200)	обязательное поле	
Дата начала этапа	S_BEGIN	D	обязательное поле	
Дата окончания этапа	S_END	D	обязательное поле, больше даты начала	
Реальная дата окончания	S_FINISH	D	больше даты начала этапа	
Стоимость этапа	S_COST	N(10)	обязательное поле	
Полученная сумма по этапу	S_SUM	N(10)	обязательное поле, значение по умолчанию – 0	
Форма отчётности	S_FORM	V(100)	обязательное поле	

Таблица 10. Схема отношения УЧАСТИЕ (Job)

Содержание поля	Имя поля	Тип, длина	Примечания*	
Проект	J_PRO	C(10)	внешний ключ (к Projects)	составной первичный ключ
Сотрудник	J_EMP	N(4)	внешний ключ (к Employees)	
Роль	J_ROLE	V(20)	обязательное поле	
Доплата	J_BONUS	N(2)		

Схема базы данных после нормализации приведена на рис. 1.

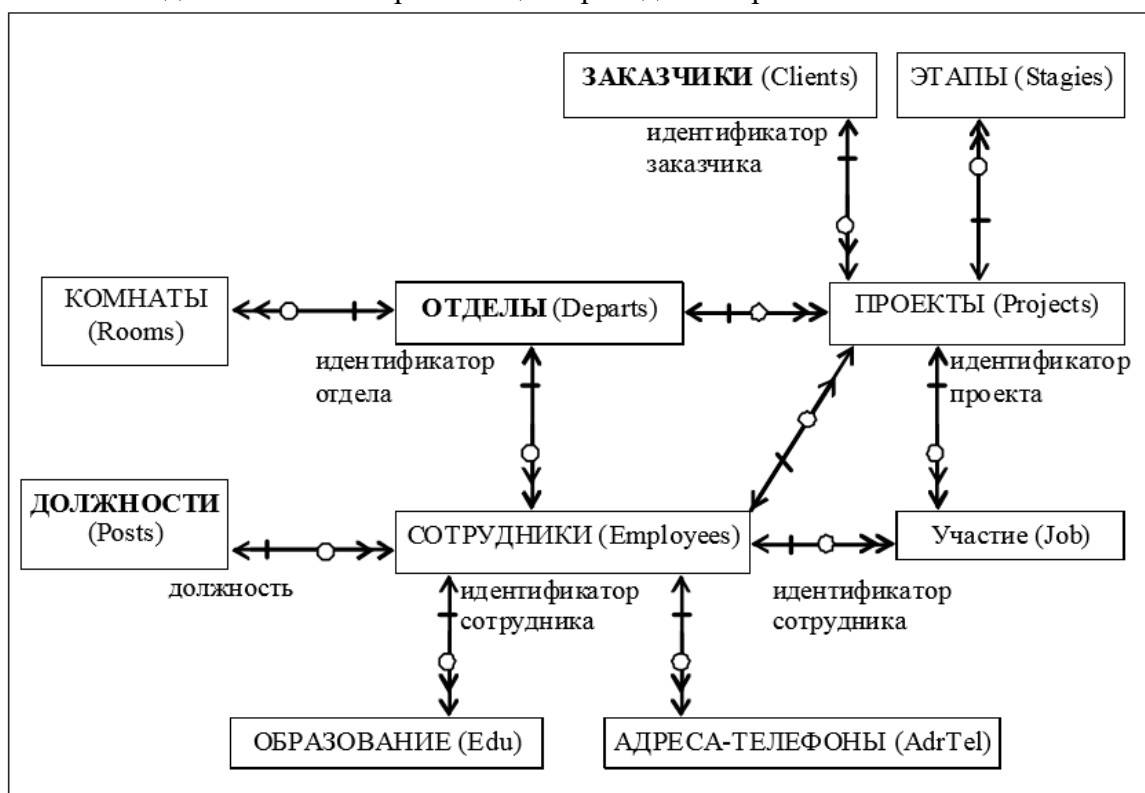


Рис. 1. Окончательная схема БД проектной организации

2. Определение дополнительных ограничений целостности

Перечислим ограничения целостности, которые не указаны в табл. 1–10.

1. Атрибут Вид образования может принимать одно из следующих значений: 'начальное', 'среднее', 'средне-специальное', 'высшее'.
2. Атрибут Роль может принимать одно из двух значений: 'исполнитель' или 'консультант'.
3. В поле Доплата хранится величина доплаты сотруднику за участие в проекте (в процентах к его окладу). Значение поля больше либо равно 0.
4. Нумерация в поле Номер этапа начинается с 1 и является непрерывной для каждого проекта.
5. Дата начала первого этапа проекта должна соответствовать началу проекта в целом, дата завершения последнего этапа должна соответствовать завершению проекта в целом. Этапы не должны пересекаться по времени и между ними не должно быть разрывов.
6. Стоимость проекта должна быть равна сумме стоимостей всех этапов этого проекта. Ограничения 4-6 нельзя реализовать в схеме отношения. В реальных БД подобные ограничения целостности реализуются вручную или программно (через внешнее приложение или специальную процедуру контроля данных – триггер).

Примечание. Вопросы архивирования данных в этом пособии подробно не рассматриваются. Но следует отметить, что обычно архив является частью БД и представляет собой набор отдельных таблиц, которые не связаны с оперативной частью БД внешними ключами. Структура архивных таблиц либо соответствует структуре тех оперативных таблиц, данные которых подлежат архивированию, либо представляет собой денормализованную таблицу, соответствующую декартову произведению оперативных таблиц. Данные в архивные таблицы переносятся специальной программой (или набором запросов) автоматически или по команде пользователя. По истечении периода хранения данные могут удаляться из архива.

3. Описание групп пользователей и прав доступа

Опишем для каждой группы пользователей права доступа к каждой таблице. Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта. Права приведены в табл. 11. Используются следующие сокращения:

- s – чтение данных (select);
- i – добавление данных (insert);
- u – модификация данных (update);
- d – удаление данных(delete).

Таблица 11. Права доступа к таблицам для групп пользователей

Таблицы	Группы пользователей (роли)				
	Руководители организации	Сотрудники отд. кадров	Руководители проектов	Бухгалтеры	Участники проектов
Отделы	S	SIUD	S	S	
Комнаты	S	SUID	S	S	S
Должности	SIUD			S	
Сотрудники	S	SUID	S	S	
Адреса-телефоны	S	SUID	S	S	
Образование	S	SUID	S	S	
Заказчики	SIUD		S		
Проекты	SIUD		S		
Этапы проектов	SIUD		SUI		
Участие	S		S	S	

Права на изменение данных в таблице УЧАСТИЕ будут назначены через представление, т.к. изменять данные этой таблицы может только руководитель проекта.

Права назначает администратор БД (или администратор безопасности, если система сложная и администраторов несколько).

Требования к оформлению:

Обязательно наличие титульного листа, на котором студент указывает выбранную тему, номер группы, Ф.И.О.

Параметры страницы: верхнее – 20 мм; нижнее – 20 мм; левое – 30 мм; правое – 10 мм.

Шрифт Times New Roman, размер шрифта- 14 (текст в таблицах-12), межстрочный интервал – 1,5 строки, абзацный отступ-1,25. Подчеркивание слов и выделение их курсивом не допускается.