

## **59. Проектирование баз данных. Цели проектирования баз данных. Понятие и процедура нормализации. Зависимости. Нормальные формы.**

### **Первая нормальная форма**

Проектирование базы данных (БД) – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы (ИС). В результате её решения должны быть определены содержание БД, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

Основная цель проектирования БД – это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте.

Цель процесса проектирования БД состоит в получении такого проекта, который удовлетворяет следующим требованиям:

- Корректность схемы БД, т.е. база должна быть гомоморфным образом моделируемой предметной области (ПО), где каждому объекту предметной области соответствуют данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных.
- Обеспечение ограничений (на объёмы внешней и оперативной памяти и другие ресурсы вычислительной системы).
- Эффективность функционирования (соблюдение ограничений на время реакции системы на запрос и обновление данных).
- Защита данных (от аппаратных и программных сбоев и несанкционированного доступа).
- Простота и удобство эксплуатации.
- Гибкость, т.е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей.

Этапы проектирования базы данных

- Инфологическое проектирование.
- Определение требований к операционной обстановке, в которой будет функционировать информационная система.
- Выбор системы управления базой данных (СУБД) и других инструментальных программных средств.
- Логическое проектирование БД.
- Физическое проектирование БД.

Этапы проектирование(Лямин)

- Семантическое моделирование
- Нормализация базы данных
- Определение правил поддержки целостности базы данных

### *Инфологическое проектирование*

Основными задачами инфологического проектирования являются определение предметной области системы и формирование взгляда на ПО с позиций сообщества будущих пользователей БД, т.е. инфологической модели ПО.

Инфологическая модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Это описание выражается в терминах не отдельных объектов ПО и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу предметной области из одного состояния в другое.

Виды:

- Функциональный подход к проектированию БД
- Предметный подход к проектированию БД
- Проектирование с использованием метода "сущность-связь"

*Определение требований к операционной обстановке*

На этом этапе производится оценка требований к вычислительным ресурсам, необходимым для функционирования системы, определение типа и конфигурации конкретной ЭВМ, выбор типа и версии операционной системы. Объём вычислительных ресурсов зависит от предполагаемого объёма проектируемой базы данных и от интенсивности их использования. Если БД будет работать в многопользовательском режиме, то требуется подключение её к сети и наличие соответствующей многозадачной операционной системы.

#### *Выбор СУБД и других программных средств*

Выбор СУБД является одним из важнейших моментов в разработке проекта БД, так как он принципиальным образом влияет на весь процесс проектирования БД и реализацию информационной системы. Теоретически при выборе СУБД нужно принимать во внимание десятки факторов. Но практически разработчики руководствуются лишь собственной интуицией и несколькими наиболее важными критериями, к которым, в частности, относятся:

- тип модели данных, которую поддерживает данная СУБД, её адекватность потребностям рассматриваемой предметной области;
- характеристики производительности системы;
- запас функциональных возможностей для дальнейшего развития ИС;
- степень оснащённости системы инструментарием для персонала администрирования данными;
- удобство и надёжность СУБД в эксплуатации;
- стоимость СУБД и дополнительного программного обеспечения.

#### *Логическое проектирование БД*

На этапе логического проектирования разрабатывается логическая структура БД, соответствующая логической модели ПО. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД. Результатом выполнения этого этапа являются схемы БД концептуального и внешнего уровней архитектуры, составленные на языках определения данных (DDL, Data Definition Language), поддерживаемых данной СУБД.

#### *Физическое проектирование БД*

Этап физического проектирования заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных, т.е. отображении логической структуры БД в структуру хранения. Решается вопрос размещения хранимых данных в пространстве памяти, выбора эффективных методов доступа к различным компонентам "физической" БД. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL). Принятые на этом этапе решения оказывают определяющее влияние на производительность системы.

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа.

База данных (БД) – это набор данных, организованный с определенной целью.

Система управления базой данных (СУБД) – это комплекс программных средств, который управляет доступом к базе данных.

Сущность – это объект, который может быть идентифицирован некоторым способом, отличающим его от других объектов. Каждая сущность обладает набором атрибутов. Атрибут – отдельная характеристика сущности.

Сущность состоит из экземпляров, каждый из которых должен отличаться от другого экземпляра. Пример: сущность – «Город», экземпляры сущности «Город» – Пушкин, Павловск, Колпино.

Связь - это логическая ассоциация, устанавливаемая между сущностями. Связь определяет количество экземпляров данной сущности, которое могут быть связаны с одним экземпляром другой сущности.

Связи бывают следующих типов:

- один к одному;
- один ко многим;
- многие ко многим.

Связи многие ко многим – пример:

```
CREATE TABLE С_Д (  
    ИД_Д,  
    ИД_С,  
  
    PRIMARY KEY (ИД_Д,ИД_С)  
  
    FOREIGN KEY (ИД_Д) REFERENCES Должность(ИД_Д)  
    ON DELETE CASCADE  
    FOREIGN KEY (ИД_С) REFERENCES Сотрудник(ИД_С)  
    ON DELETE CASCADE  
);
```

Ключ - минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.

Первичный ключ сущности позволяет идентифицировать ее экземпляры, а внешний – экземпляры сущности, которая находится в связи с данной сущностью.

Независимая сущность. Для определения экземпляра сущности нет необходимости ссылаться на другие сущности.

Независимые сущности – пример:

```
CREATE TABLE Должность (  
    ИД_Д,  
    Название,  
    Ставки,  
    PRIMARY KEY (ИД_Д)  
);
```

Зависимая сущность. Для определения экземпляра такой сущности необходимо сослаться на экземпляр независимой сущности, с которой связана зависимая сущность.

Зависимые сущности

```
CREATE TABLE Оклад (  
    ИД_Д,  
    Дата,  
    Сумма,  
  
    PRIMARY KEY (ИД_Д, Дата, Сумма),  
  
    FOREIGN KEY (ИД_Д) REFERENCES Должность(ИД_Д)  
    ON DELETE CASCADE  
);
```

В каталоге или словаре содержится подробная информация, касающаяся таблиц, индексов, ограничений поддержки целостности и т.д.

Пример 1:

```
SELECT OBJECT_NAME, OBJECT_TYPE  
FROM USER_OBJECTS;  
SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE  
FROM USER_TAB_COLUMNS  
WHERE TABLE_NAME = 'название_таблицы';
```

Представление - это пустая именованная таблица, определяемая перечнем тех столбцов таблиц и признаками тех их строк, которые хотелось бы в ней увидеть.

Пример 2:

```
CREATE VIEW C AS  
SELECT A.a, A.b, B.c
```

```
FROM A, B
WHERE A.a = B.b;
```

Курсор – это логический указатель, который может использоваться в приложении для перемещения по набору строк, указывая поочередно на каждую из них и таким образом обеспечивая возможность адресации этих строк – по одной за один раз.

Пример 3:

```
DECLARE C CURSOR FOR
SELECT A.a, A.b, B.c
FROM A, B
WHERE A.a = B.b;
```

```
OPEN C;
```

```
FETCH C;
```

Транзакция – это логическая единица работы, обычно включающая несколько операций над базой данных.

Транзакция начинается при выполнении операции BEGIN TRANSACTION и прекращается при выполнении операции COMMIT или ROLLBACK.

Пример 4:

```
-- Перевод денег со счета A на счет B
```

```
BEGIN TRANSACTION
```

```
UPDATE ACCOUNT A; -- Списание денег со счета A
```

```
UPDATE ACCOUNT B; -- Зачисление денег на счет B
```

```
IF <все выполнено нормально>
```

```
THEN COMMIT; -- Нормальное завершение
```

```
ELSE ROLLBACK; -- Аварийное завершение
```

```
END IF;
```

Триггером называют сочетание трех компонентов: событие, условие и действие.

Событием является операция в базе данных

Условие – это логическое выражение, которое должно принимать значение «Истина» для того, чтобы выполнено действие.

Действие – это триггерная процедура.

Пример 5:

```
CREATE OR REPLACE TRIGGER person_t
```

```
BEFORE INSERT ON person
```

```
FOR EACH ROW
```

```
BEGIN
```

```
SELECT person_s.NEXTVAL INTO :new.id FROM DUAL;
```

```
END;
```

Ограничения целостности

Целостность (integrity - нетронутость, неприкосновенность, сохранность, целостность) понимается как правильность данных в любой момент времени.

Группы правил целостности

- целостность по первичным ключам;
- целостность по внешним ключам;
- целостность, определяемая пользователем.

Целостность, определяемая пользователем

Для любой конкретной базы данных существует ряд дополнительных специфических правил, которые относятся к ней одной и определяются разработчиком. Чаще всего контролируется:

- уникальность тех или иных атрибутов;
- диапазон значений (экзаменационная оценка от 2 до 5);
- принадлежность набору значений (пол "М" или "Ж").

Функциональная зависимость (ФЗ)

Пусть R – таблица, а A и B – произвольные подмножества ее полей. Тогда B функционально зависит от A и пишут  $A \rightarrow B$ , если каждое значение множества A связано только с одним значением множества B.

Пример 6:

Фамилия	Пол	Должность
Иванов И.П.	м	учитель
Петров И.И.	м	завуч

Сидоров П.П.	м	учитель
Иванова Н.П.	ж	директор
Иванов И.П.	м	секретарь

(Фамилия) > (Пол)

В рамках реляционной модели данных Э.Ф. Коддом (E.F. Codd) был разработан аппарат нормализации отношений и предложен механизм, позволяющий любое отношение преобразовать к третьей нормальной форме.

Нормализация схемы отношения выполняется путём декомпозиции схемы. Декомпозицией схемы отношения R называется замена её совокупностью схем

$$R = \bigcup_i A_i$$

отношений  $A_i$  таких, что  $A_i$  и не требуется, чтобы отношения  $A_i$  были непересекающимися.

Введём понятие простого и сложного атрибута. Простой атрибут – это атрибут, значения которого атомарны (т.е. неделимы). Сложный атрибут может иметь значение, представляющее собой конкатенацию нескольких значений одного или разных доменов. Аналогом сложного атрибута может быть агрегат или повторяющийся агрегат данных.

Процедура нормализации

- разбить таблицу на проекции, чтобы исключить неполные ФЗ;
- разбить таблицу на проекции, чтобы исключить зависимости от неключевых полей;
- разбить таблицу на проекции, чтобы МЗ являлись также ФЗ;
- разбить таблицу на проекции, чтобы в каждой ее полной декомпозиции все проекции содержали потенциальный ключ.

Теорема Хита

Пусть A, B и C поля таблицы. Если  $A \twoheadrightarrow B$ , то таблицу можно представить как результат соединения ее проекций (A, B) и (A, C).

Пример 7:

Фамилия	Пол	Должность
Иванов И.П.	м	учитель
Петров И.И.	м	завуч
Сидоров П.П.	м	учитель
Иванова Н.П.	ж	директор
Иванов И.П.	м	секретарь

Фамилия	Пол
Иванов И.П.	м
Петров И.И.	м
Сидоров П.П.	м
Иванова Н.П.	ж

Фамилия	Должность
Иванов И.П.	учитель
Петров И.И.	завуч
Сидоров П.П.	учитель
Иванова Н.П.	директор
Иванов И.П.	секретарь

Полная ФЗ

Поле B находится в полной функциональной зависимости от составного поля A, если оно функционально зависит от A и не зависит функционально от любого подмножества поля A.

Первая нормальная форма

Таблица находится в первой нормальной форме (1НФ) тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Вторая нормальная форма

Таблица находится во второй нормальной форме (2НФ), если она удовлетворяет определению 1НФ и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом.

Пример 8:

Фамилия	Пол	Должность	Зарплата
Иванов И.П.	м	учитель	15000
Петров И.И.	м	завуч	17500
Сидоров П.П.	м	учитель	17000
Иванова Н.П.	ж	директор	20000
Иванов И.П.	м	секретарь	50000

(Фамилия) > (Пол)

(Фамилия, Должность) > (Зарплата)

Пример 9:

Фамилия	Пол	Должность	Зарплата
Иванов И.П.	м	учитель	15000
Петров И.И.	м	завуч	17500
Сидоров П.П.	м	учитель	17000
Иванова Н.П.	ж	директор	20000
Иванов И.П.	м	секретарь	10000

Фамилия	Пол
Иванов И.П.	м
Петров И.И.	м
Сидоров П.П.	м
Иванова Н.П.	ж

Фамилия	Должность	Зарплата
Иванов И.П.	учитель	15000
Петров И.И.	завуч	17500
Сидоров П.П.	учитель	17000
Иванова Н.П.	директор	20000
Иванов И.П.	секретарь	10000

Третья нормальная форма

Таблица находится в третьей нормальной форме (3НФ), если она удовлетворяет определению 2НФ и не одно из ее неключевых полей не зависит функционально от любого другого неключевого поля.

Пример 10:

Фамилия	Должность	Зарплата
Иванов И.П.	учитель	15000
Петров И.И.	завуч	17500
Сидоров П.П.	учитель	15000
Иванова Н.П.	директор	20000

(Фамилия) > (Должность)

(Должность) > (Зарплата)

Пример 11:

Фамилия	Должность	Оклад
Иванов И.П.	учитель	15000
Петров И.И.	завуч	17500
Сидоров П.П.	учитель	15000
Иванова Н.П.	директор	20000

Фамилия	Должность
Иванов И.П.	учитель
Петров И.И.	завуч
Сидоров П.П.	учитель
Иванова Н.П.	директор

Должность	Оклад
учитель	15000
завуч	17500
директор	20000

### Нормальная форма Бойса-Кодда

Таблица находится в нормальной форме Бойса-Кодда (НФБК), если и только если любая функциональная зависимость между его полями сводится к полной функциональной зависимости от возможного ключа.

### Полная декомпозиция

Полной декомпозицией таблицы называют такую совокупность произвольного числа ее проекций, соединение которых полностью совпадает с содержимым таблицы.

### Многозначная зависимость

Пусть  $A$ ,  $B$  и  $C$  – поля таблицы. Поле  $A$  многозначно определяет поле  $B$  той же таблицы ( $A \twoheadrightarrow B$ ), если множество значений поля  $B$ , соответствующее заданной паре  $(A, C)$ , зависит только от значения  $A$  и не зависит от значения  $C$ .

Пример 12:

Фамилия	Должность	Телефон
Иванов И.П.	учитель	11-111
Иванов И.П.	учитель	22-111
Иванов И.П.	директор	11-111
Иванов И.П.	директор	22-111
Сидоров В.П.	учитель	55-342
Сидоров В.П.	учитель	63-754
Сидоров В.П.	учитель	65-224

(Фамилия)  $\twoheadrightarrow$  (Должность)

(Фамилия)  $\twoheadrightarrow$  (Телефон)

### Теорема Фейгина

Пусть  $A$ ,  $B$  и  $C$  поля таблицы. Если  $A \twoheadrightarrow B$  и  $A \twoheadrightarrow C$ , то таблицу можно представить как результат соединения ее проекций  $(A, B)$  и  $(A, C)$ .

Пример 13:

Фамилия	Должность	Телефон
Иванов И.П.	учитель	11-111
Иванов И.П.	учитель	22-111
Иванов И.П.	директор	11-111
Иванов И.П.	директор	22-111
Сидоров В.П.	учитель	55-342
Сидоров В.П.	учитель	63-754
Сидоров В.П.	учитель	65-224

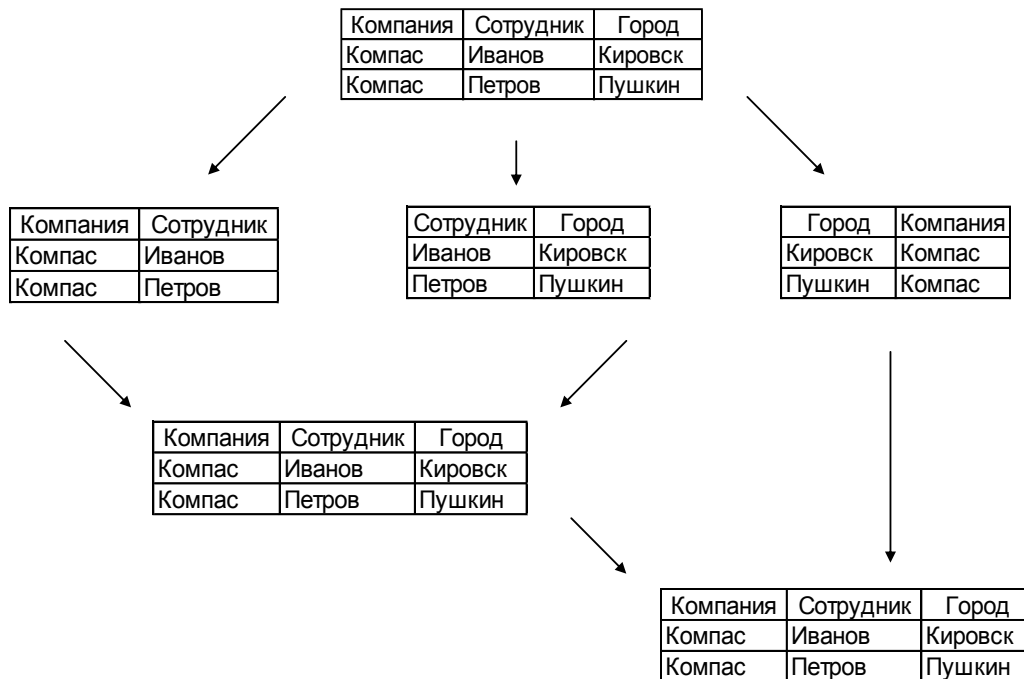
Фамилия	Должность
Иванов И.П.	учитель
Иванов И.П.	директор
Сидоров В.П.	учитель

Фамилия	Телефон
Иванов И.П.	11-111
Иванов И.П.	22-111
Сидоров В.П.	55-342
Сидоров В.П.	63-754
Сидоров В.П.	65-224

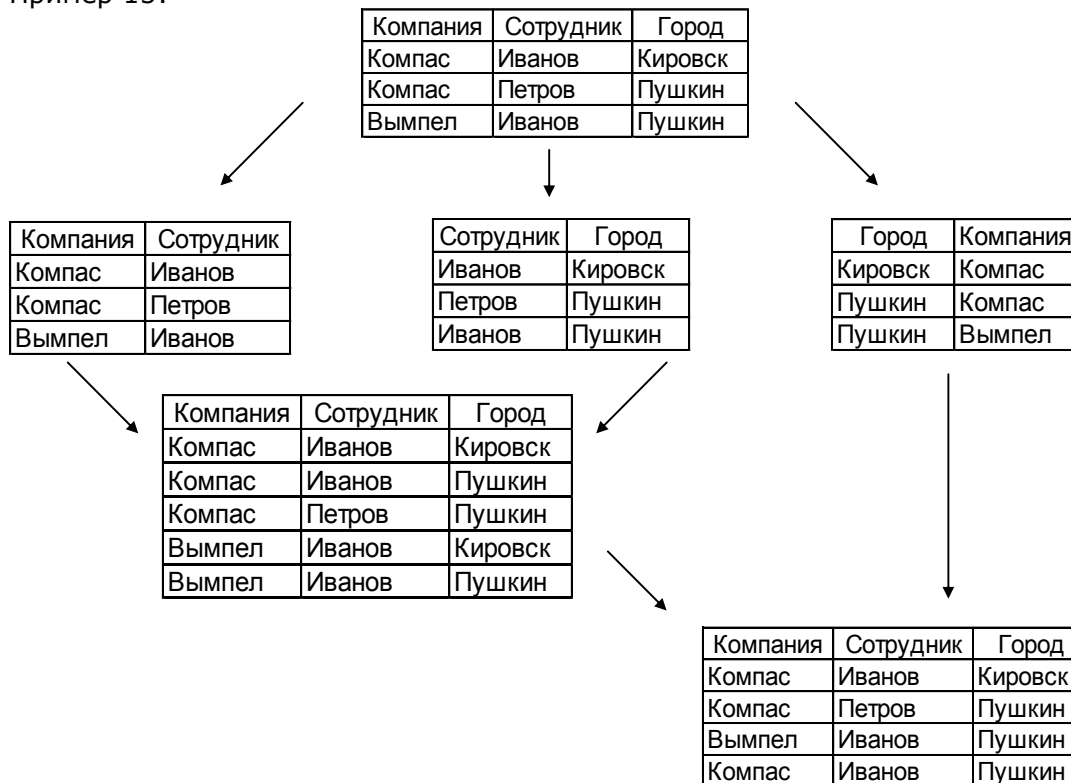
### Четвертая нормальная форма

Таблица находится в четвертой нормальной форме (4НФ), если она удовлетворяет определению НФБК и все многозначные зависимости являются функциональными зависимостями от ключей.

Пример 14:



Пример 15:



Пятая нормальная форма

Таблица находится в пятой нормальной форме (5НФ) тогда и только тогда, когда в каждой ее полной декомпозиции все проекции содержат потенциальный ключ. Таблица, не имеющая ни одной полной декомпозиции, также находится в 5НФ.

Пример 16 - проектирование:

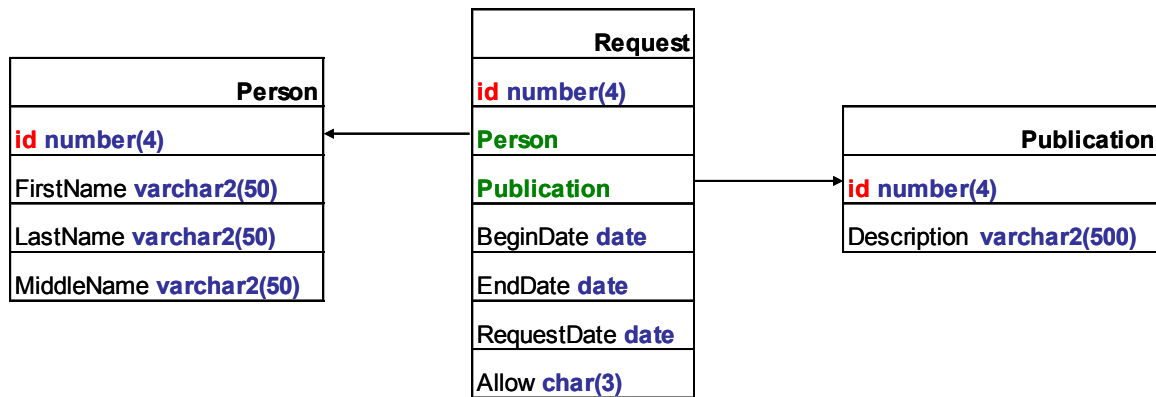
Требуется разработать систему, которая позволяет хранить информацию о книгах частной библиотеки и обеспечивает возможность:

- гостю просматривать список книг, подавать прошение на временное изъятие книг из библиотеки для ознакомления;
- хозяину добавлять и удалять книги, изменять расположение книг в шкафу, удовлетворять или отклонять прошения гостей.

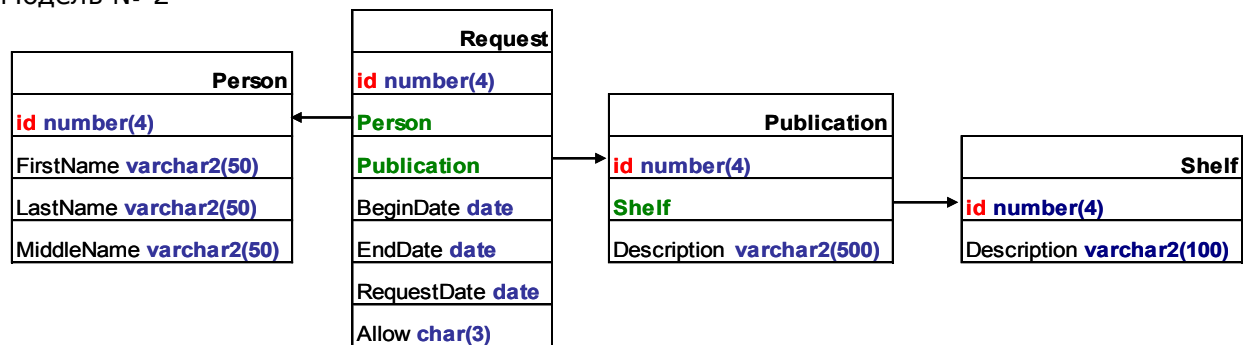
Сущности



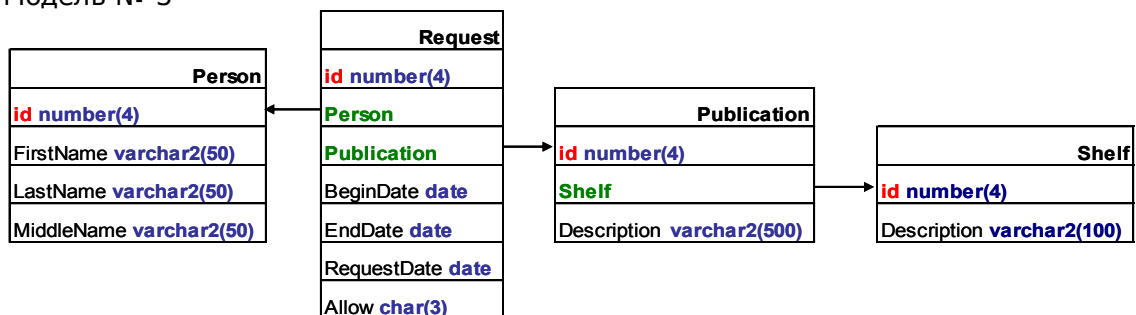
- Книга
  - Прощение
  - Пользователь
- Модель № 1



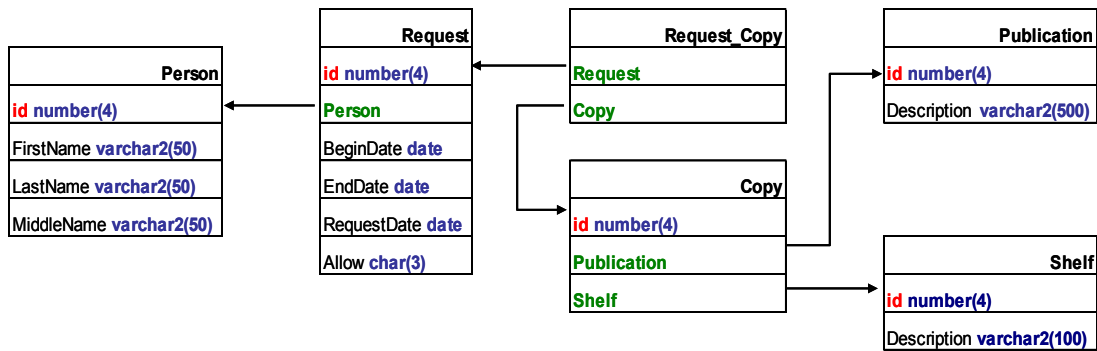
- Сущности
- Книга
  - Прощение
  - Пользователь
  - Книжная полка
- Модель № 2



- Сущности
- Издание
  - Экземпляр
  - Прощение
  - Пользователь
  - Книжная полка
- Модель № 3



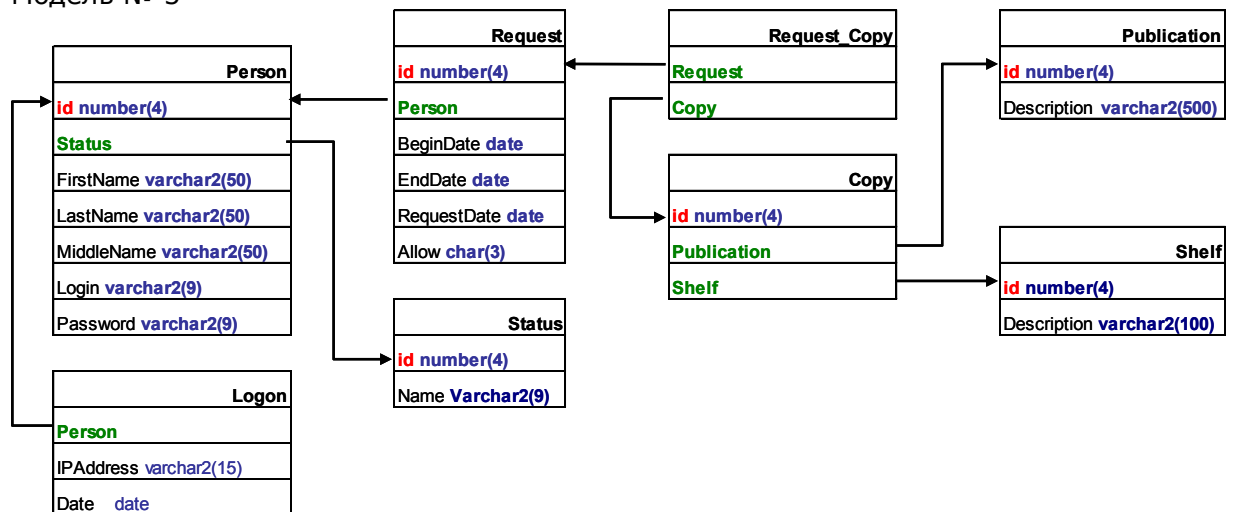
Модель № 4



### Сущности

- Издание
- Экземпляр
- Прощение
- Пользователь
- Книжная полка
- Роль
- Сессия

### Модель № 5



### Литература

- Лекции Лямина
- <http://rema44.ru/resurs/study/dbprj/dbprj.html>