

Базы данных. Нормализация баз данных.

Преподаватель:

Косова Екатерина Александровна

Ekosova@mesi.ru

Избыточное дублирование и аномалии

Следует различать простое (неизбыточное) и избыточное дублирование данных.

Наличие первого из них допускается в базах данных, а избыточное дублирование данных может приводить к проблемам при обработке данных.

Пример избыточного дублирования

S_T

| Сотрудник | Телефон |
|--------------|---------|
| Иванов И.М. | 3721 |
| Петров М.И. | 4328 |
| Сидоров Н.Г. | 4328 |
| Егоров В.В. | 4328 |

Комментарий

Для сотрудников, находящихся в одном помещении, номера телефонов совпадают. Номер телефона 4328 встречается несколько раз, хотя для каждого служащего номер телефона уникален. Поэтому ни один из номеров не является избыточным. При удалении одного из номеров телефонов будет утеряна информация о том, по какому номеру можно дозвониться до одного из служащих.

Пример избыточного дублирования

S_T_H

| Сотрудник | Телефон | H_комн |
|--------------|---------|--------|
| Иванов И.М. | 3721 | 109 |
| Петров М.И. | 4328 | 111 |
| Сидоров Н.Г. | 4328 | 111 |
| Егоров В.В. | 4328 | 111 |

Комментарий

Естественно предположить, что все служащие в одной комнате имеют один и тот же телефон. Следовательно, в рассматриваемом отношении имеется избыточное дублирование данных. Так, в связи с тем, что Сидоров и Егоров находятся в той же комнате, что и Петров, их номера можно узнать из кортежа со сведениями о Петрове.

Пример неудачного отношения

С_Т_Н

| Сотрудник | Телефон | Н_комн |
|--------------|---------|--------|
| Иванов И.М. | 3721 | 109 |
| Петров М.И. | 4328 | 111 |
| Сидоров Н.Г. | — | 111 |
| Егоров В.В. | — | 111 |

Неудачность подобного способа исключения избыточности:

- 1) при программировании придется потратить дополнительные усилия на создание механизма поиска информации для прочерков таблицы.
- 2) память все равно выделяется под атрибуты с прочерками, хотя дублирование данных и исключено.
- 3) при исключении из коллектива Петрова кортеж со сведениями о нем будет исключен из отношения, а значит, уничтожена информация о телефоне 111-й комнаты, что недопустимо.

Возможный способ декомпозиции

T_H

| Телефон | Н_комн |
|---------|--------|
| 3721 | 109 |
| 4328 | 111 |

C_H

| Сотрудник | Н_комн |
|--------------|--------|
| Иванов И.М. | 109 |
| Петров М.И. | 111 |
| Сидоров Н.Г. | 111 |
| Егоров В.В. | 111 |

Возможный способ

Теперь, если Петрова уволят из учреждения и, как следствие этого, удалят всякую информацию о нем из баз данных учреждения, это не приведет к утере информации о номере телефона в 111-й комнате.

Процедура декомпозиции отношения S_{T_N} на два отношения S_N и N_T является основной процедурой **нормализации отношений.**

Избыточное дублирование данных создает проблемы при обработке кортежей отношения - **аномалии** обновления отношения. Для некоторых отношений проблемы возникают при попытке удаления, добавления или редактирования их кортежей.

Аномалии

Аномалиями будем называть такую ситуацию в таблицах БД, которая приводит к противоречиям в БД либо существенно усложняет обработку данных.

- Аномалии **модификации** проявляются в том, что изменение значения одного данного может повлечь за собой просмотр всей таблицы и соответствующее изменение некоторых других записей таблицы.

Пример

- Изменение номера телефона в комнате 111 потребует просмотра всей таблицы и изменения поля Н_комн

С_Т_Н

| Сотрудник | Телефон | Н_комн |
|--------------|---------|--------|
| Иванов И.М. | 3721 | 109 |
| Петров М.И. | 4328 | 111 |
| Сидоров Н.Г. | 4328 | 111 |
| Егоров В.В. | 4328 | 111 |

- Аномалии **удаления** состоят в том, что при удалении какого-либо данного из таблицы может пропасть и другая информация, которая не связана напрямую с удаляемым данным.

Пример

- Удаление записи о сотруднике Иванове приводит к исчезновению информации о номере телефона, установленного в 109-й комнате

С_Т_Н

| Сотрудник | Телефон | Н_комн |
|--------------|---------|--------|
| Иванов И.М. | 3721 | 109 |
| Петров М.И. | 4328 | 111 |
| Сидоров Н.Г. | 4328 | 111 |
| Егоров В.В. | 4328 | 111 |

- Аномалии **добавления** возникают в случаях, когда информацию в таблицу нельзя поместить до тех пор, пока она неполная, либо вставка новой записи требует дополнительного просмотра таблицы.

Пример

- Операция добавления нового сотрудника.
Противоестественно хранить сведения в этой таблице только о комнате и номере телефона в ней, пока никто из сотрудников не помещен в нее. Если в таблице С_Т_Н поле «Служащий» является ключевым, то хранение в ней неполных записей недопустимо.

С_Т_Н

| Сотрудник | Телефон | Н_комн |
|--------------|---------|--------|
| Иванов И.М. | 3721 | 109 |
| Петров М.И. | 4328 | 111 |
| Сидоров Н.Г. | 4328 | 111 |
| Егоров В.В. | 4328 | 111 |

Пример

- Ситуация включения в таблицу нового сотрудника. При добавлении таких записей для исключения противоречий желательно проверить номер телефона и соответствующий номер комнаты хотя бы с одним из сотрудников, сидящих в той же комнате. Если окажется, что у нескольких сотрудников, сидящих в одной комнате, имеются разные телефоны, не ясно, что делать (или несколько телефонов в комнате, или какой-то номер ошибочный)

Пример избыточности

ПРЕПОДАВАТЕЛЬ

| ФИО | Должн | Оклад | Стаж | Д_Стаж | Каф | Предм | Группа | ВидЗан |
|--------------|---------|-------|------|--------|-----|---------|--------|--------|
| Иванов И.М. | преп | 500 | 5 | 100 | 25 | СУБД | 256 | Практ |
| Иванов И.М. | преп | 500 | 5 | 100 | 25 | ПЛ/1 | 123 | Практ |
| Петров М.И. | ст.преп | 800 | 7 | 100 | 25 | СУБД | 256 | Лекция |
| Петров М.И. | ст.преп | 800 | 7 | 100 | 25 | Паскаль | 256 | Практ |
| Сидоров Н.Г. | преп | 500 | 10 | 150 | 25 | ПЛ/1 | 123 | Лекция |
| Сидоров Н.Г. | преп | 500 | 10 | 150 | 25 | Паскаль | 256 | Лекция |
| Егоров В.В. | преп | 500 | 5 | 100 | 24 | ПЭВМ | 244 | Лекция |

Явная и неявная избыточность

- **Явная** избыточность заключается в том, что в отношении строки с данными о преподавателях, проводящих занятия в нескольких группах, повторяются соответствующее число раз.
- **Неявная** избыточность проявляется в одинаковых окладах у всех преподавателей и в одинаковых надбавках к окладу за одинаковый стаж

Пример избыточности

ПРЕПОДАВАТЕЛЬ

| ФИО | Должн | Оклад | Стаж | Д_Стаж | Каф | Предм | Группа | ВидЗан |
|--------------|---------|-------|------|--------|-----|---------|--------|--------|
| Иванов И.М. | преп | 500 | 5 | 100 | 25 | СУБД | 256 | Практ |
| Иванов И.М. | преп | 500 | 5 | 100 | 25 | ПЛ/1 | 123 | Практ |
| Петров М.И. | ст.преп | 800 | 7 | 100 | 25 | СУБД | 256 | Лекция |
| Петров М.И. | ст.преп | 800 | 7 | 100 | 25 | Паскаль | 256 | Практ |
| Сидоров Н.Г. | преп | 500 | 10 | 150 | 25 | ПЛ/1 | 123 | Лекция |
| Сидоров Н.Г. | преп | 500 | 10 | 150 | 25 | Паскаль | 256 | Лекция |
| Егоров В.В. | преп | 500 | 5 | 100 | 24 | ПЭВМ | 244 | Лекция |

Средством исключения избыточности в
отношениях и аномалий является
нормализация отношений

- **Нормальная форма** — свойство отношения в модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Процесс преобразования отношений базы данных (БД) к виду, отвечающему нормальным формам, называется **нормализацией**.

Нормализация предназначена для приведения структуры БД к виду, обеспечивающему минимальную логическую избыточность.

Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации.

- Устранение избыточности производится, как правило, за счёт декомпозиции отношений таким образом, чтобы в каждом отношении хранились только первичные факты (то есть факты, не выводимые из других хранимых фактов).

Нормальные формы

- **1) первая нормальная форма (1 NF);**
- **2) вторая нормальная форма (2 NF);**
- **3) третья нормальная форма (3 NF);**
- **4) нормальная форма Бойса – Кодда (BCNF);**
- **5) четвертая нормальная форма (4 NF);**
- **6) пятая нормальная форма, или нормальная форма проекции-соединения (5 NF или PJ/NF).**

Основные свойства нормальных форм состоят в следующем:

- 1) каждая следующая нормальная форма в некотором смысле лучше предыдущей нормальной формы;
- 2) при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

Нормализовывать базы данных или приводить базы данных к нормальному виду – это значит определять такие схемы базовых отношений, чтобы максимально уменьшить необходимость написания программного кода, увеличить производительность работы базы данных, облегчить поддержку целостности данных по состоянию и ссылочной целостности. То есть сделать код и работу с ним максимально простой и удобной разработчикам и пользователям.

- **Составные атрибуты** – это атрибуты, составленные из нескольких простых атрибутов.
- **Многозначные атрибуты** – это атрибуты, представляющие множество значений.

| Атрибут | Простой | Составной |
|----------------|----------------|------------------|
| Однозначный | Телефон | Адрес |
| Многозначный | Телефоны | Адреса |

Определение

Любое базовое отношение находится в **первой нормальной форме** тогда и только тогда, когда схема этого отношения содержит только простые и только однозначные атрибуты, причем обязательно с одной и той же семантикой.

Пример

Сотрудники (*№ табельный*, Фамилия Имя Отчество, Код должности, Телефоны, Дата приема или увольнения);

Ошибки:

1) атрибут «Фамилия Имя Отчество» является составным, т. е. составленным из разнородных элементов;

2) атрибут «Телефоны» является многозначным, т. е. его значением является множество значений;

3) атрибут «Дата приема или увольнения» не имеет однозначной семантики, т. е. в последнем случае не понятно, какая именно дата внесена.

Что же необходимо сделать для приведения этого отношения к нормальной форме?

- 1) необходимо провести разбиение составных атрибутов на простые, для того, чтобы исключить эти самые составные атрибуты, а также атрибуты с составной семантикой.
- 2) необходимо провести декомпозицию этого отношения, т. е. нужно разбить его на несколько новых самостоятельных отношений, с тем чтобы исключить многозначные атрибуты.

Решение

после приведения отношения «Сотрудники» к первой нормальной форме или 1NF путем его декомпозиции мы получим систему следующих отношений с заданными на них первичными и внешними ключами

Сотрудники (**№ табельный**, Фамилия, Имя, Отчество, Код должности, Дата приема, Дата увольнения);

Телефоны (**Телефон**, **№ табельный**);

Определение

Базовое отношение находится во **второй нормальной форме** относительно заданного множества функциональных зависимостей тогда и только тогда, когда оно находится в первой нормальной форме и, кроме того, каждый неключевой атрибут полностью функционально зависит от каждого ключа.

Замечание

- Если потенциальный ключ отношения является простым, то отношение автоматически находится в 2НФ.

Таким образом, теперь при нормализации отношения мы должны следить и за выполнением условий пребывания отношения в первой нормальной форме, т. е. следить, чтобы его атрибуты были простыми и однозначными, а также за выполнением второго условия, касающегося ограничений функциональных зависимостей.

Пример

Аудитории (*№ корпуса, № аудитории,*
Площадь кв. м, № табельный коменданта
корпуса);

Primary key (№ корпуса, № аудитории);

Кроме того, определена следующая
система функциональной зависимости:

{№ корпуса} > {№ табельный коменданта
корпуса};

Все условия пребывания отношения «Аудитории» в первой нормальной форме выполнены - все атрибуты этого отношения однозначны и просты.

Но условие, что каждый неключевой элемент должен полностью функционально зависеть от ключа, не выполняется.

Почему?

Потому что атрибут «№ табельный коменданта корпуса» функционально зависит не от составного ключа «№ корпуса, № аудитории», а от части этого ключа, т. е. от атрибута «№ корпуса».

Таким образом, основной задачей нашей нормализации становится задача добиться того, чтобы ключи распределялись таким образом, чтобы, в частности, атрибут «№ табельный коменданта корпуса» полностью функционально зависел от всего ключа, а не от его какой-то части.

Решение

Корпуса (***№ корпуса***, № табельный
коменданта корпуса);

Аудитории (***№ аудитории***, № корпуса,
Площадь кв. м);

В отношении «Корпуса» неключевой атрибут «№ табельный коменданта корпуса» полностью функционально зависит от первичного ключа «№ корпуса».

В отношении «Аудитории» атрибут первичного ключа «№ корпуса» является внешним ключом, ссылающимся на первичный ключ отношения «Корпуса». В этом отношении неключевой атрибут «Площадь кв. м» полностью зависит от всего составного первичного ключа «№ корпуса, № аудитории» и не зависит, даже не может зависеть ни от какой из его частей.

Определение

- Базовое отношение находится в **третьей нормальной форме** относительно заданного множества функциональных зависимостей тогда и только тогда, когда оно находится во второй нормальной форме и каждый неключевой атрибут полностью функционально зависит только от ключей.

- Фактически в третьей нормальной форме каждый неключевой атрибут зависит от ключа, причем от всего ключа целиком и ни от чего другого, кроме как от ключа.

Пример

- Сотрудники (*№ табельный*, Фамилия, Имя, Отчество, Код должности, Оклад);

Кроме того, над данным отношением «Сотрудники» задана следующая система функциональных зависимостей:

- {Код должности} > {Оклад};

Отношение «Сотрудники» не находится в третьей нормальной форме, ведь получается, что неключевой атрибут «Оклад» полностью функционально зависит от атрибута «Код должности», хотя этот атрибут и не является ключевым.

Решение

Должности (***Код должности***, Оклад);

- Сотрудники (***№ табельный***, Фамилия, Имя, Отчество, **Код должности**);

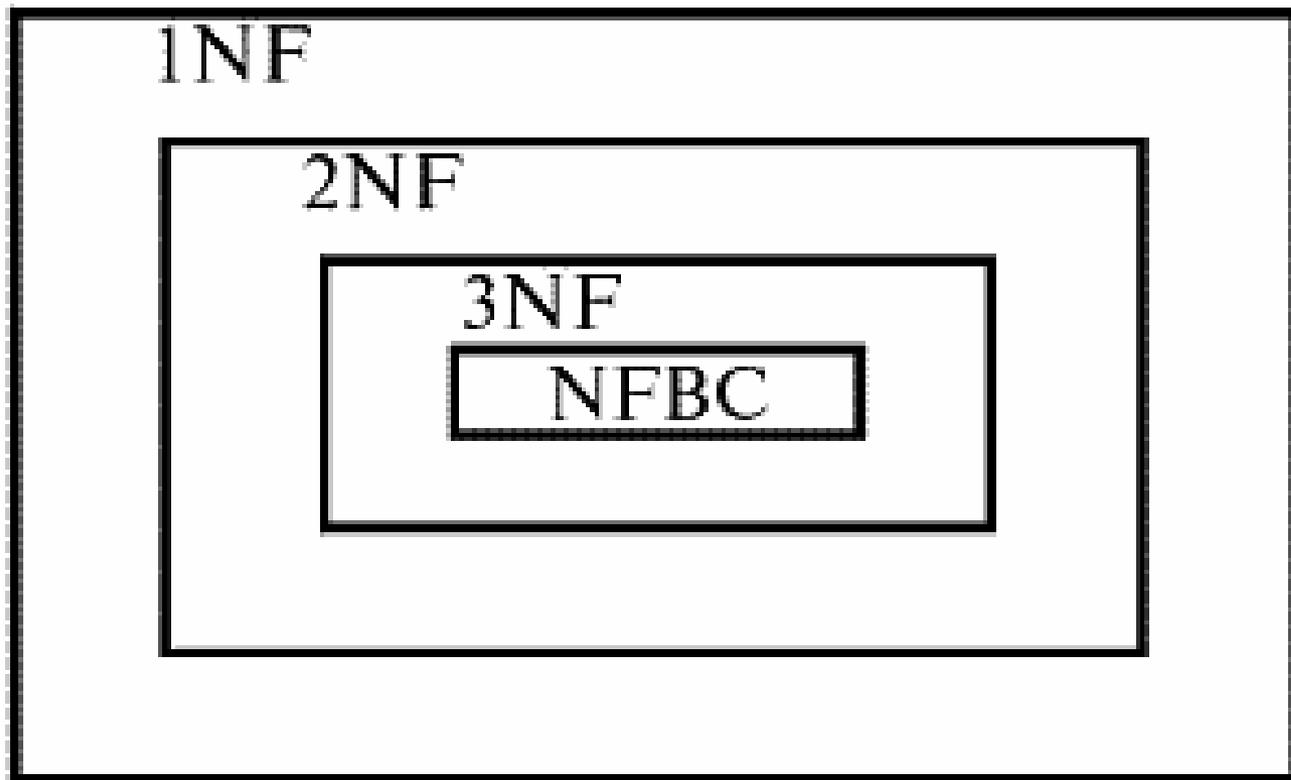
Нормальная форма Бойса – Кодда (NFBC)

- Базовое отношение находится в **нормальной форме Бойса – Кодда** тогда и только тогда, когда она находится в третьей нормальной форме, и при этом не только любой неключевой атрибут полностью функционально зависит от любого ключа, но и любой ключевой атрибут должен полностью функционально зависеть от любого ключа.

- Таким образом, требование о фактической зависимости неключевых атрибутов от всего ключа целиком и ни от чего другого, кроме как от ключа, распространяется и на ключевые атрибуты.
- Практика проектирования показала, что не всегда удастся привести базовое отношение к NFBC.

Вложенность нормальных форм

Вложенность нормальных форм – это отношение понятий ослабленной и усиленной формы по отношению друг к другу.



Сравнение нормализованных и ненормализованных моделей

| Критерий | Отношения слабо нормализованы (1НФ, 2НФ) | Отношения сильно нормализованы (3НФ) |
|---|---|---|
| Адекватность базы данных предметной области | ХУЖЕ (-) | ЛУЧШЕ (+) |
| Легкость разработки и сопровождения базы данных | СЛОЖНЕЕ (-) | ЛЕГЧЕ (+) |
| Скорость выполнения вставки, обновления, удаления | МЕДЛЕННЕЕ (-) | БЫСТРЕЕ (+) |
| Скорость выполнения выборки данных | БЫСТРЕЕ (+) | МЕДЛЕННЕЕ (-) |

Практическое задание – Привести в ЗНФ

ПРЕПОДАВАТЕЛЬ

| ФИО | Должн | Оклад | Стаж | Д_Стаж | Каф | Предм | Группа | ВидЗан |
|--------------|---------|-------|------|--------|-----|---------|--------|--------|
| Иванов И.М. | преп | 500 | 5 | 100 | 25 | СУБД | 256 | Практ |
| Иванов И.М. | преп | 500 | 5 | 100 | 25 | ПЛ/1 | 123 | Практ |
| Петров М.И. | ст.преп | 800 | 7 | 100 | 25 | СУБД | 256 | Лекция |
| Петров М.И. | ст.преп | 800 | 7 | 100 | 25 | Паскаль | 256 | Практ |
| Сидоров Н.Г. | преп | 500 | 10 | 150 | 25 | ПЛ/1 | 123 | Лекция |
| Сидоров Н.Г. | преп | 500 | 10 | 150 | 25 | Паскаль | 256 | Лекция |
| Егоров В.В. | преп | 500 | 5 | 100 | 24 | ПЭВМ | 244 | Лекция |

Выполнение практической работы:

- 1) Привести базу данных из примера в 3 НФ;
- 2) Построить ER-модель базы данных;
- 3) Обосновать каждый этап приведения базы данных в 3НФ;
- 4) Рассмотреть свой пример приведения базы данных в 3НФ;
- 5) Составить отчет по результатам работы;
- 6) Отправить отчет на проверку преподавателю.

Модели представления данных

Хранимые в базе данные имеют определенную логическую структуру – описываются некоторой **моделью представления данных**, поддерживаемой СУБД.

Модели данных

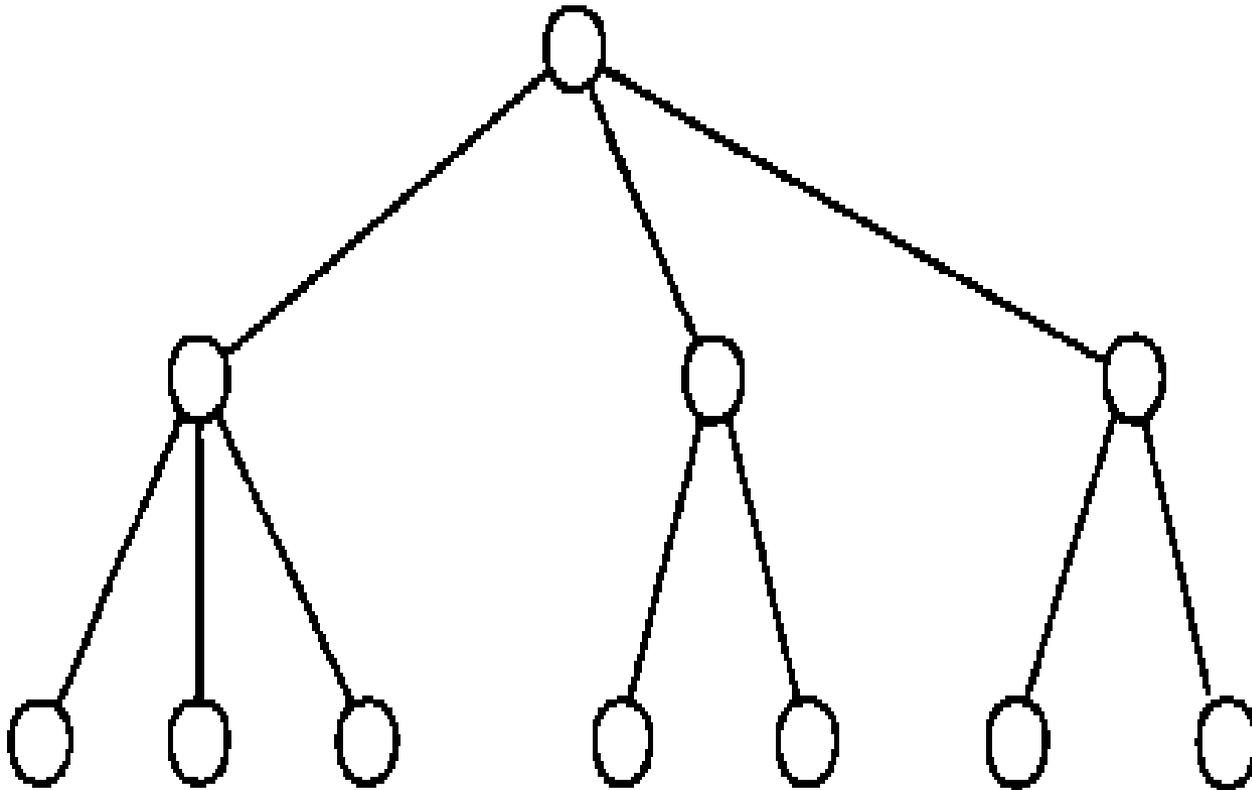
Классические модели данных:

- Иерархическая;
- Сетевая;
- Реляционная.

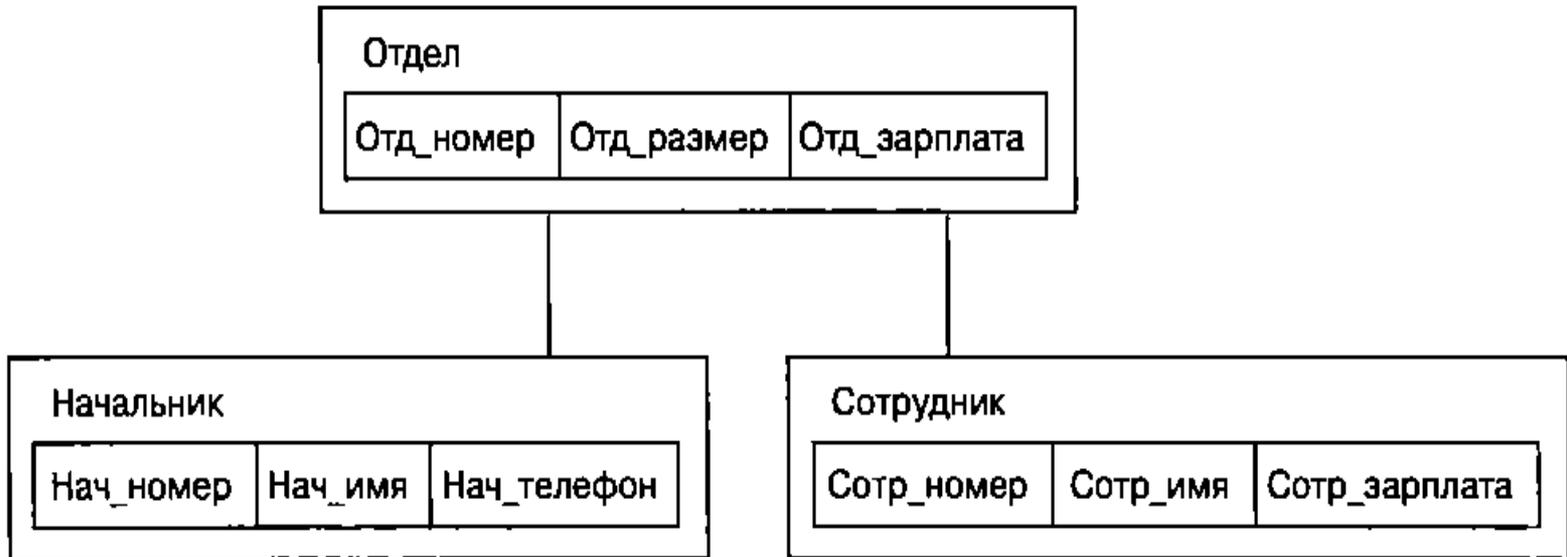
Новые модели данных:

- Постреляционная;
- Многомерная;
- Объектно-ориентированная;
- Объектно-реляционные, дедуктивно-объектно-ориентированные, семантические, концептуальные и ориентированные модели.

Иерархическая модель



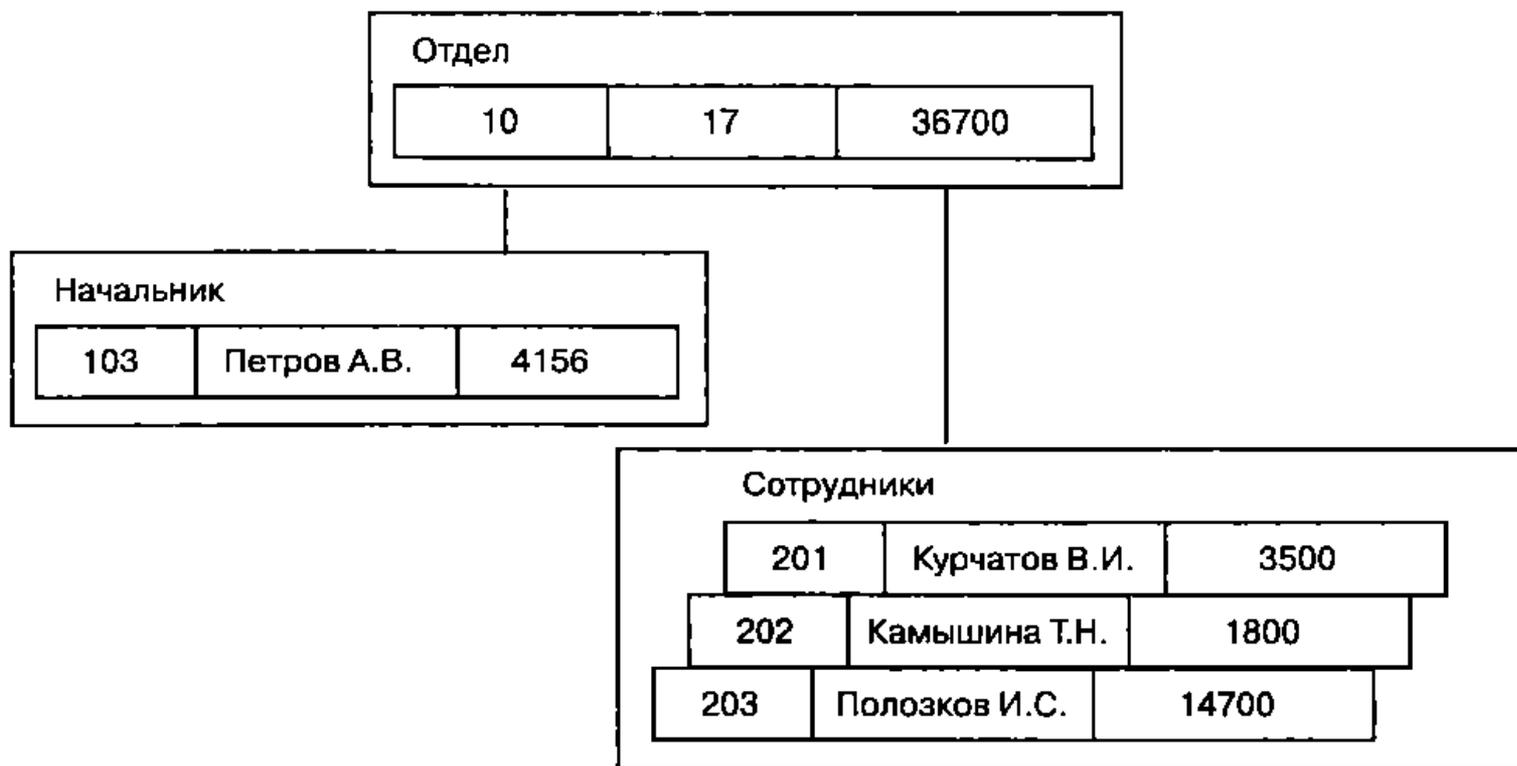
Иерархическая модель



Определения

- **Корневым** называется тип, который имеет подчиненные типы и сам не является подтипом.
- **Подчиненный** тип (подтип) является потомком по отношению к типу, который выступает для него в роли **предка** (родителя).
- Потомки одного и того же типа являются **близнецами** по отношению друг к другу.

Данные в иерархической базе



Основные операции манипулирования

- Поиск указанного экземпляра БД (например, дерева со значением 10 в поле отд_номер);
- Переход от одного дерева к другому;
- Переход от одной записи к другой внутри дерева (например, к следующей записи типа Сотрудники);
- Вставка новой записи в указанную позицию;
- Удаление текущей записи и т.д.

- Основное правило контроля целостности – потомок не может существовать без родителя, а у некоторых родителей может не быть потомков.

Иерархическая модель

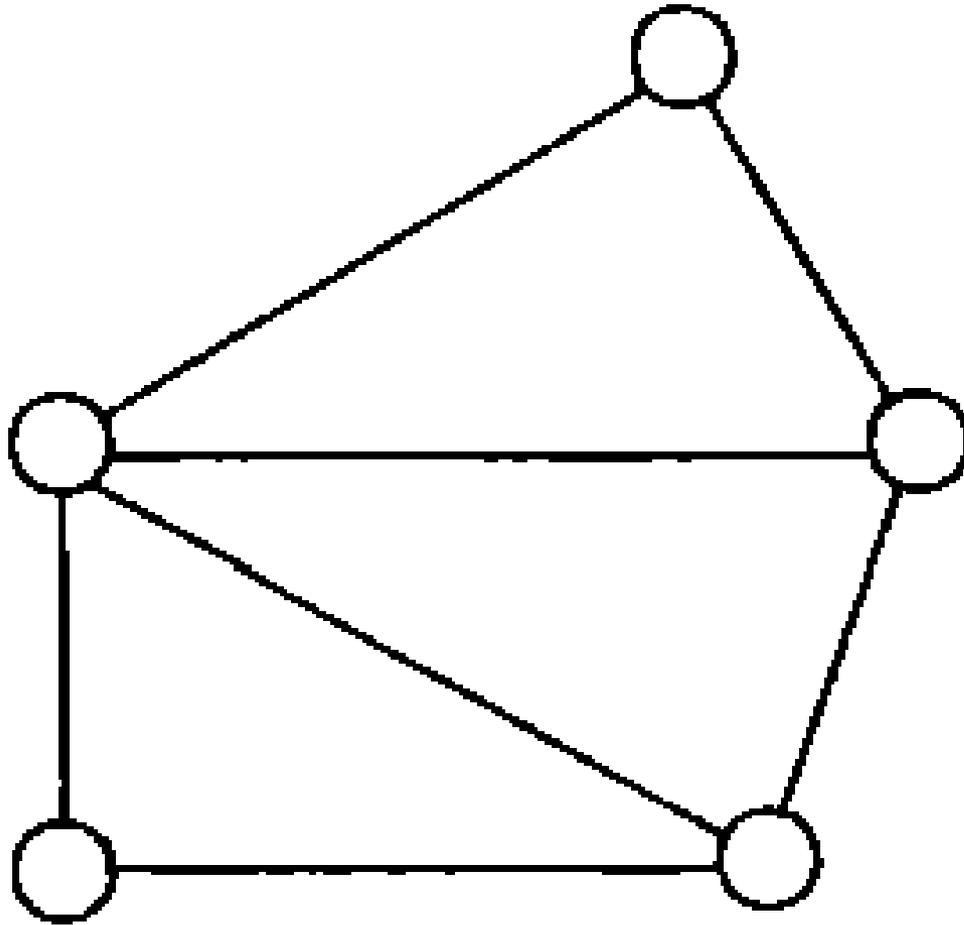
Достоинства:

- Эффективное использование памяти ЭВМ;
- Неплохие показатели времени выполнения основных операций над данными;
- Модель удобна для работы с иерархически упорядоченной информацией.

Недостатки:

- Громоздкость для обработки информации с достаточно сложными логическими связями;
- Сложность понимания для обычного пользователя.

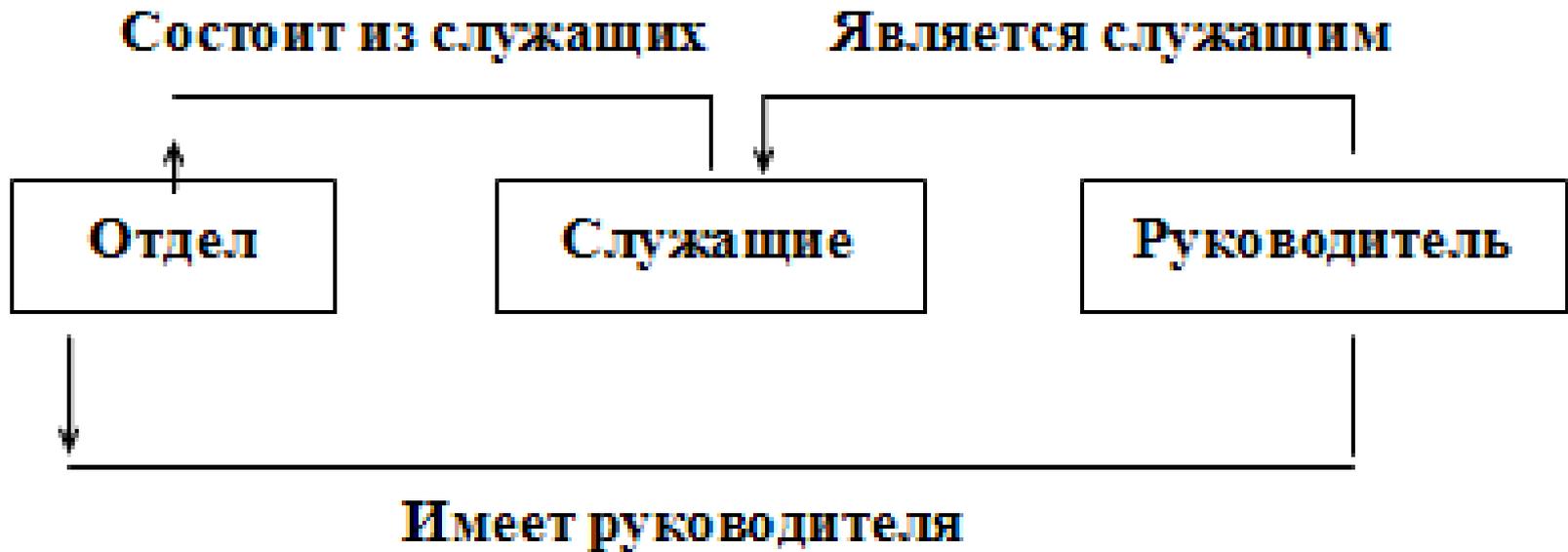
Сетевая модель



Сетевая модель

- Для описания схемы сетевой БД используется две группы типов: «запись» и «связь».
- Тип «связь» определяется для двух типов «запись» – предка и потомка.
- Сетевая БД состоит из набора записей и набора соответствующих связей.
- Если в иерархической модели запись-потомок могла иметь только одну запись-предка, то в сетевой модели данных запись-потомок может иметь произвольное число записей-предков (сводных родителей).

Сетевая модель



Основные операции манипулирования

- Поиск записи в БД;
- Переход от предка к первому потомку;
- Переход от потомка к предку;
- Создание новой записи;
- Удаление текущей записи;
- Обновление текущей записи;
- Включение записи в связь;
- Исключение записи из связи;
- Изменение связей и т.д.

Достоинства и недостатки

- **Достоинства:**
- Возможность эффективной реализации по показателям затрат памяти и оперативности.
- В сравнении с иерархической моделью сетевая модель предоставляет большие возможности в смысле допустимости образования произвольных связей.
- **Недостатки:**
- Высокая сложность и жесткость схемы БД;
- Сложность для понимания и выполнения обработки информации в БД обычным пользователем.
- В сетевой модели ослаблен контроль целостности связей вследствие допустимости установления произвольных связей между записями.

Реляционная модель

- **Достоинства:**
- Простота, понятность и удобство физической реализации на ЭВМ.
- **Недостатки:**
- Сложность описания иерархических и сетевых связей.

Постреляционная модель

- Классическая реляционная модель предполагает неделимость данных, хранящихся в полях записей таблиц (информация представляется в 1НФ).
- Постреляционная модель представляет собой расширенную реляционную модель, снимающую ограничение неделимости данных, хранящихся в записях таблиц.
- Постреляционная модель допускает многозначные поля (значения которых состоят из подзначений).

Сравнение моделей

а)

INVOICES

| INVNO | CUSTNO |
|-------|--------|
| 0373 | 8723 |
| 8374 | 8232 |
| 7364 | 8723 |

INVOICE.ITEMS

| INVNO | GOODS | QTY |
|-------|---------|-----|
| 0373 | Сыр | 3 |
| 0373 | Рыба | 2 |
| 8374 | Лимонад | 1 |
| 8374 | Сок | 6 |
| 8374 | Печенье | 2 |
| 7364 | Йогурт | 1 |

б)

INVOICES

| INVNO | CUSTNO | GOODS | QTY |
|-------|--------|---------|-----|
| 0373 | 8723 | Сыр | 3 |
| | | Рыба | 2 |
| 8374 | 8232 | Лимонад | 1 |
| | | Сок | 6 |
| | | Печенье | 2 |
| 7364 | 8723 | Йогурт | 1 |

Достоинства и недостатки

- **Достоинства:**
- Возможность представления совокупности связанных реляционных таблиц одной постреляционной таблицей. Это обеспечивает высокую наглядность представления информации и повышение эффективности ее обработки.
- **Недостатки:**
- Сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.

Многомерная модель

- **Многомерные СУБД** – узкоспециализированные СУБД, предназначенные для интерактивной аналитической обработки информации.
- Многомерность модели данных означает многомерное логическое представление структуры информации при описании и в операциях манипулирования данными.

Определения

- Агрегируемость данных – рассмотрение информации на различных уровнях ее обобщения.
- Историчность данных – обеспечение высокого уровня статичности (неизменности) данных и их взаимосвязей, а также обязательность привязки данных ко времени.
- Прогнозируемость данных – задание функций прогнозирования и применение их к различным временным интервалам.

Сравнение с реляционной моделью

а)

| Модель | Месяц | Объем |
|-----------|--------|-------|
| «Жигули» | июнь | 12 |
| «Жигули» | июль | 24 |
| «Жигули» | август | 5 |
| «Москвич» | июнь | 2 |
| «Москвич» | июль | 18 |
| «Волга» | июль | 19 |

б)

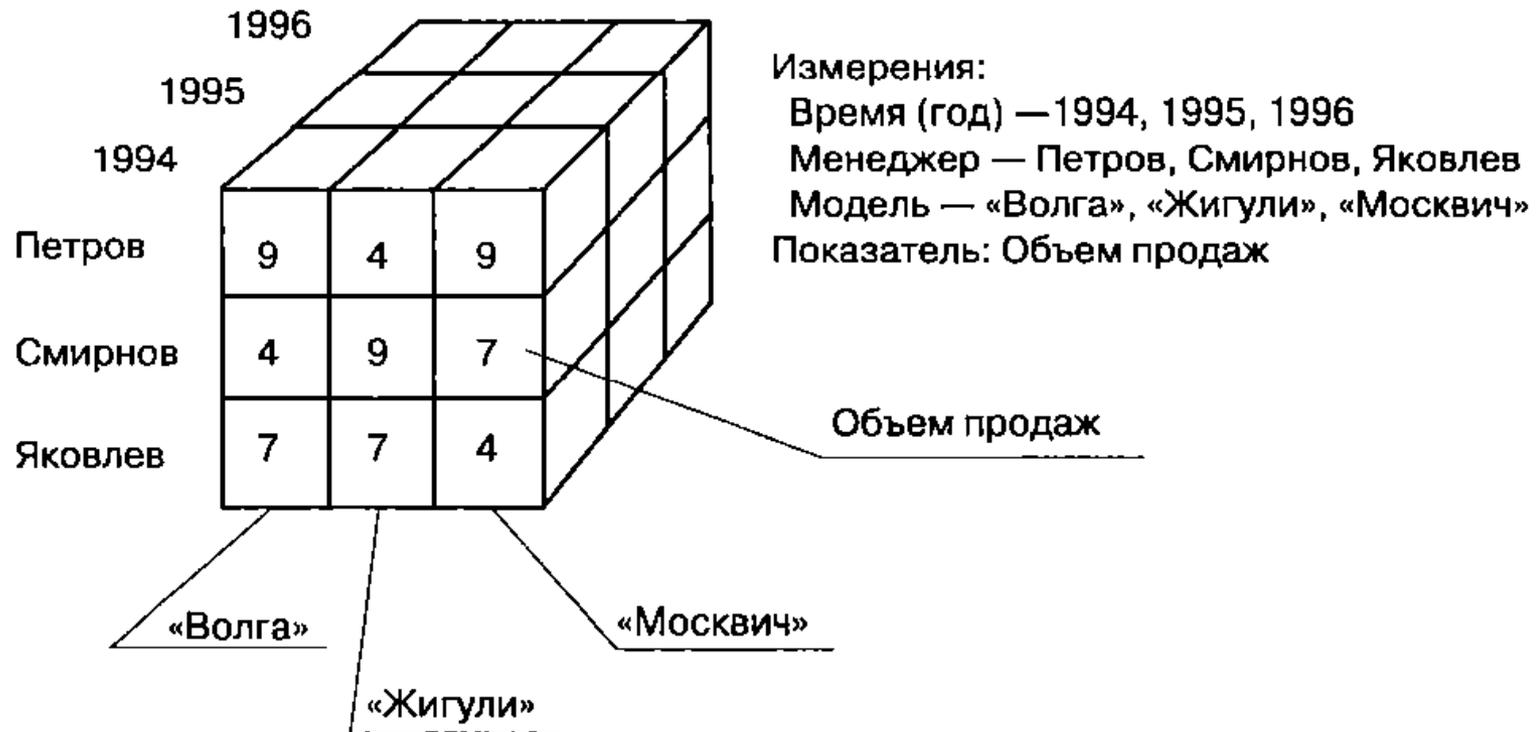
| Модель | Июнь | Июль | Август |
|-----------|------|------|--------|
| «Жигули» | 12 | 24 | 5 |
| «Москвич» | 2 | 18 | No |
| «Волга» | No | 19 | No |

- Если речь идет о многомерной модели с мерностью больше двух, то не обязательно представлять информацию в виде многомерных объектов (трех-, четырех- и более мерных гиперкубов).
- Пользователю в этих случаях более удобно иметь дело с двухмерными таблицами. Данные при этом представляют собой **«срезы»** из многомерного хранилища данных.

Определения

- **Измерение** – множество однотипных данных, образующих одну из граней гиперкуба. Измерения играют роль индексов, служащих для идентификации конкретных значений в ячейках гиперкуба.
- **Ячейка (показатель)** – поле, значение которого однозначно определяется фиксированным набором измерений. Ячейка может быть переменной или формулой.
- **«Срез»** - подмножество гиперкуба, полученное в результате фиксации одного или нескольких измерений.

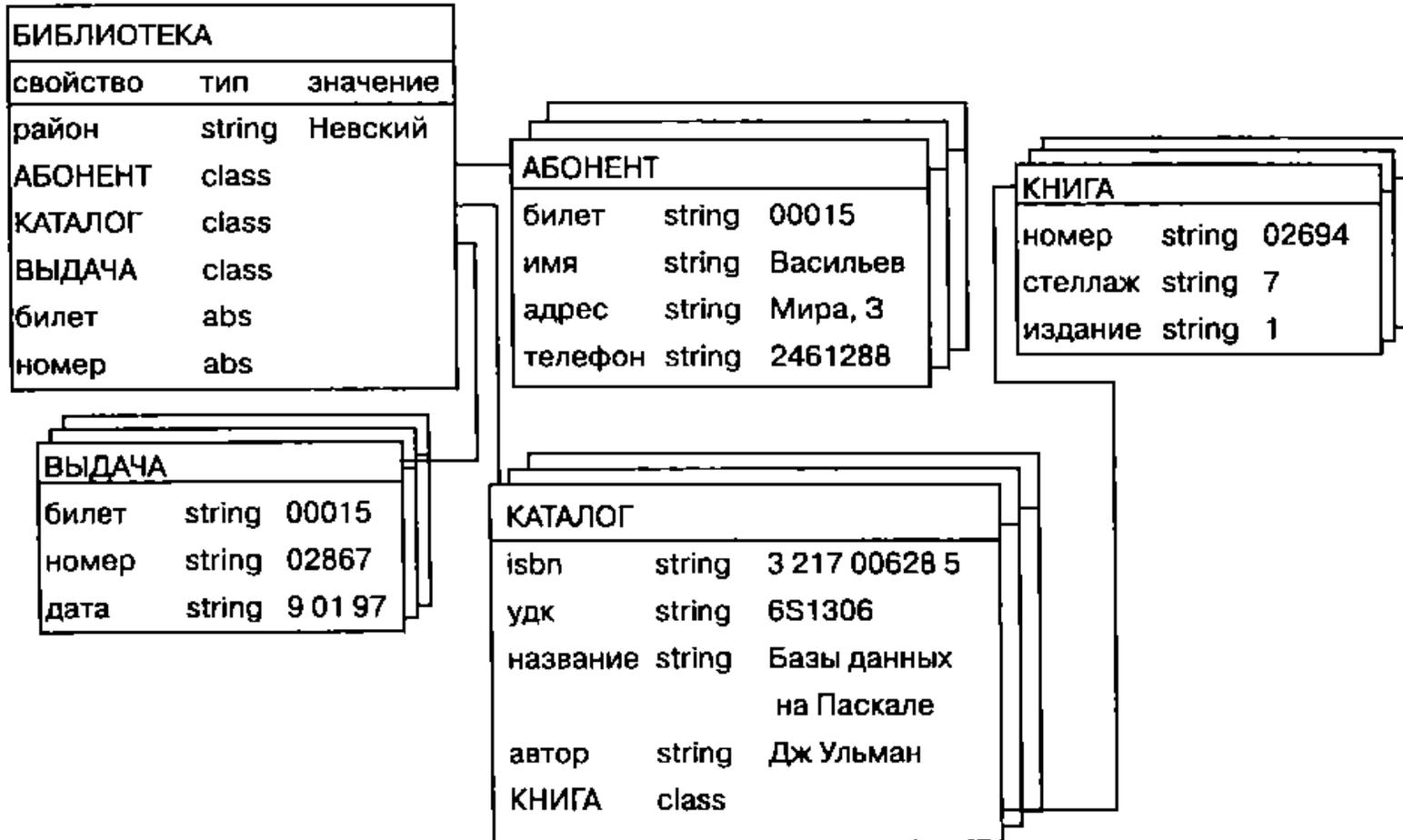
Пример трехмерной модели



Достоинства и недостатки

- **Достоинства:**
- Удобство и эффективность аналитической обработки больших объемов данных, связанных со временем;
- **Недостатки:**
- Громоздкость для простейших задач обычной оперативной обработки информации.

Объектно-ориентированная модель



Достоинства и недостатки

- **Достоинства:**
- Возможность отображения информации о сложных взаимосвязях объектов.
- ООМД позволяет идентифицировать отдельную запись данных и определять функции их обработки.
- **Недостатки:**
- Высокая понятийная сложность, неудобство обработки данных, низкая скорость выполнения запросов.

**Удачи в выполнении работы
и изучении курса!**