Типы моделей данных. Взаимосвязи в моделях данных.

Виды моделей данных

Ядром любой БД является модель данных.

Модель данных – это совокупность структуры данных и операций их обработки.

Иерархическая модель данных

Представляет собой совокупность элементов, связанных по строго определенным правилам.

Объекты, связанные иерархическими отношениями, образуют ориентированный граф. Основными понятиями иерархической модели данных являются: ***уровень, узел*** (или элемент) и ***связь.***

Свойства иерархической модели:

- каждый узел связан только с одним вышестоящим узлом, кроме вершины;
- иерархическая модель данных имеет только одну вершину, узел не подчинен более никаким узлам;
- от каждого узла существует единственный путь к вершине;
- связь не может быть установлена между объектами, находящимися через уровень;
- связь между узлами первого уровня не определяется.

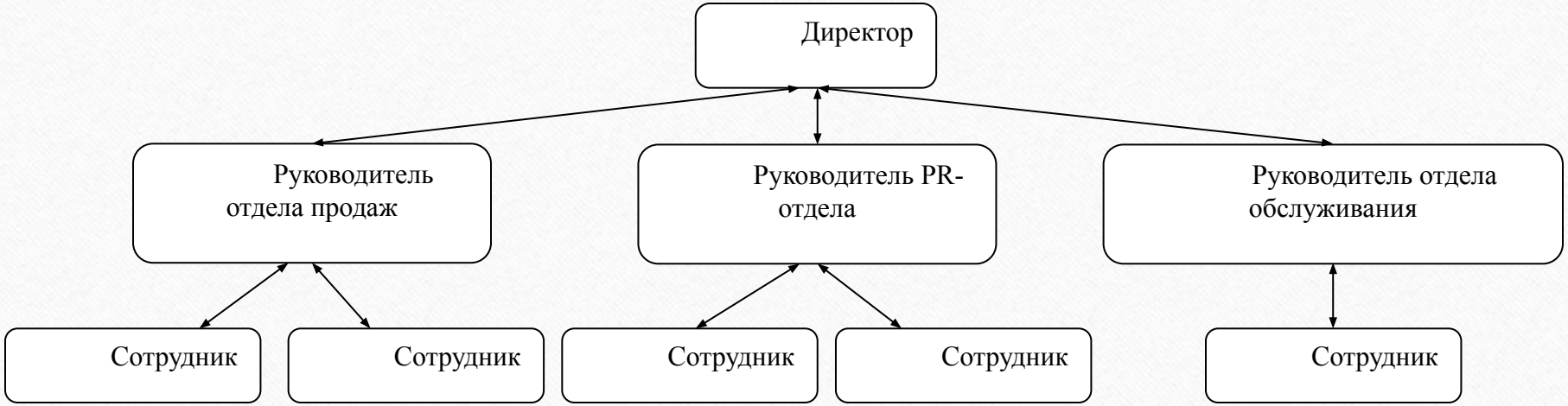

Рисунок 3.1 – Иерархическая структура данных

Преимущества:

- Простота.
- Минимальный расход памяти.

Недостатки:

- Отсутствие универсальности – не всякую информацию можно выразить в иерархической модели данных.
- Исключительно навигационный принцип доступа к данным.
- Доступ к данным только через корневой элемент.

Сетевая модель данных

Элементами этой модели являются: ***уровень, узел, связь.*** Отличия в том, что элемент одного уровня может быть связан с любым количеством элементов соседнего уровня, и не существует подчиненности уровней друг другу.

Свойства сетевой модели:

- связь не может быть установлена между объектами, находящимися через уровень;
- связь между узлами первого уровня не определяется.

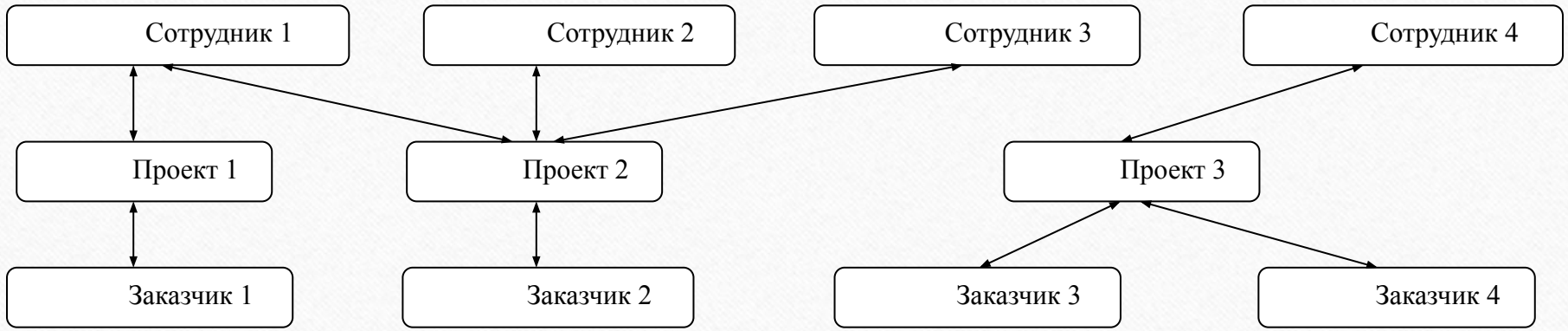

Рисунок 3.2 – Сетевая структура данных

Преимущества:

- Универсальность.
- Возможность доступа к данным через значения нескольких отношений.

Недостатки:

- Сложность – обилие понятий, вариантов их взаимосвязей и способов реализации.
- Допустимость только навигационного принципа доступа к данным.

Реляционная модель данных (табличная)

Это способ представления данных в виде таблиц. Элементы: ***поле*** (столбец), ***запись*** (строка) и ***таблица*** (отношение).

Под ***реляционной системой*** понимается система, для которой характерны следующие свойства:

- данные пользователя представлены только в виде таблиц;
- пользователю предоставляются операторы, генерирующие новые таблицы из старых (для выборки данных).

Students:

StudentID	LastName	FirstName	MiddleName	GroupID
1	Казаков	Петр	Владимирович	1
2	Васильев	Иван	Аркадьевич	2
4	Шишкина	Дарья	Сергеевна	1

Groups:

GroupID	Supervisor
1	Царев С.М.
2	Пестов Д.Н.

Рисунок 3.3 – Реляционная структура данных

Преимущества:

- Простота. В такой модели всего одна информационная конструкция, формализующая табличное представление. Она наиболее привычна для пользователя.
- Теоретическое обоснование. Существуют строгие методы нормализации данных в таблицах.
- Независимость данных. При изменении БД, ее структуры необходимы бывают лишь минимальные изменения прикладных программ.

Недостатки:

- Низкая скорость, т.к. требуются операции соединения.
- Большой расход памяти в силу организации *всех* данных в виде таблиц.

Система инвертированных списков

Система инвертированных списков – система индексов. Систему инвертированных списков можно рассматривать как частный случай сетевой модели данных, которая имеет два уровня. Основные элементы: ***основной файл***, ***инвертированный список*** (файл), ***список связей***.

В такой системе имеется несколько основных файлов, имеющих единую сквозную нумерацию.

Сотрудники:

Сотрудник	Должность
01 Иванов	программист
02 Сидоров	лаборант
03 Шишкин	преподаватель
04 Васильев	преподаватель

Зарплата:

Сотрудник	Дата	Сумма
05 Иванов	1.10.2008	5000
06 Сидоров	5.10.2008	7500
07 Иванов	3.12.2008	10000
08 Шишкин	3.12.2008	8000
09 Васильев	25.01.2009	5000
10 Васильев	27.01.2009	8750

Должность:

- программист – 01;
- лаборант – 02;
- преподаватель – 03, 04

- *Сотрудники – Зарплата*:
  - 01 – 05, 07
  - 02 – 06
  - 03 – 08
  - 04 – 09, 10
- *Зарплата – Сотрудники*:
  - 05 – 01
  - 06 – 02
  - 07 – 01
  - 08 – 03
  - 09 – 04
  - 10 – 04

Инвертированные списки являются основой для создания *информационно-поисковых систем* (ИПС). В ИПС ключевые атрибуты соответствуют ключевым словам, определяющим тематику поиска.

Взаимосвязи в моделях данных

На практике часто используются связи, устанавливающие различные виды соответствия между объектами «связанных» типов, — это один к одному (1:1), один ко многим (1:M), многие ко многим (M: M).

Один к одному

Связь один к одному означает, что каждому экземпляру первого объекта (A) соответствует только один экземпляр второго объекта (B) и, наоборот, каждому экземпляру второго объекта (B) соответствует только один экземпляр первого объекта (A).

Один ко многим

Связь один ко многим означает, что каждому экземпляру одного объекта (A) может соответствовать несколько экземпляров другого объекта (B), а каждому экземпляру второго объекта (B) может соответствовать только один экземпляр первого объекта (A).

Многие ко многим

Связь многие ко многим означает, что каждому экземпляру одного объекта (А) могут соответствовать несколько экземпляров второго объекта (В) и, наоборот, каждому экземпляру второго объекта (В) могут соответствовать тоже несколько экземпляров первого объекта (А).

Обеспечение непротиворечивости и целостности данных в базе

Выделяют два основных типа ограничений по условию целостности данных в базе.

- 1. Каждая строка таблицы должна отличаться от остальных ее строк значением хотя бы одного столбца.
- 2.Внешний ключ не может быть указателем на несуществующую строку той таблицы, на которую он ссылается. Это ограничение называется ограничением целостности данных в базе по ссылкам.

Архитектура БД

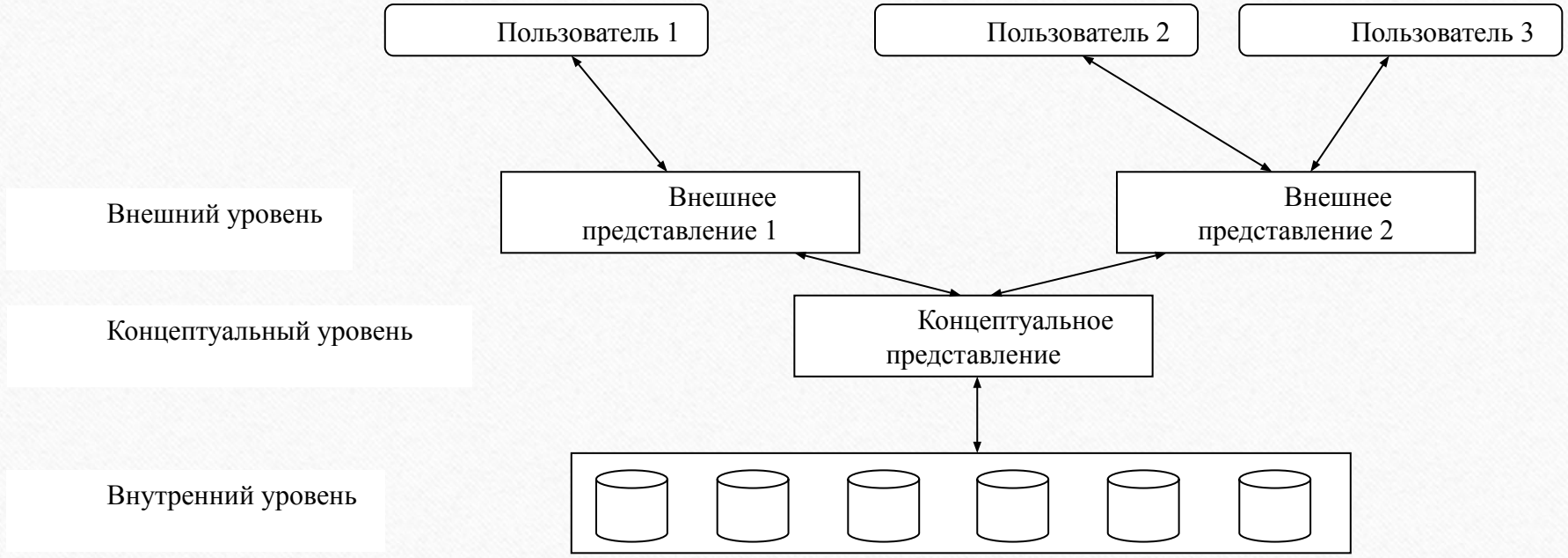

Рисунок 3.4 – Уровни архитектуры систем баз данных

Внутренний уровень

Внутренний уровень (называемый также *физическим*) наиболее близок к физическому хранилищу информации, т.е. связан со способами хранения информации на физических устройствах. Внутренний уровень отображает физические элементы для хранения информации. Он представляет собой бесконечное адресное пространство, из которого информация проецируется на внешние носители.

Внешний уровень

Внешний уровень (называемый также *пользовательским*) наиболее близок к пользователям, т.е. связан со способами представления данных для отдельных пользователей (прикладной программист или конечный пользователь). Для каждого пользователя может существовать свой язык СУБД. Для прикладного программиста – это язык программирования, а для конечного пользователя – это язык, основанный на меню, формах и т.д.

Концептуальный уровень

Концептуальный уровень (называемый также *логическим*) является «промежуточным» уровнем между двумя первыми. Это представление всей информации БД в более абстрактной форме. На этом уровне хранятся собственно данные, независимые от формы их представления. Концептуальное представление состоит из множества экземпляров данных.