КАФЕДРА: КИИ

ТЕМА: МОДЕЛИ И СТРУКТУРЫ ДАННЫХ

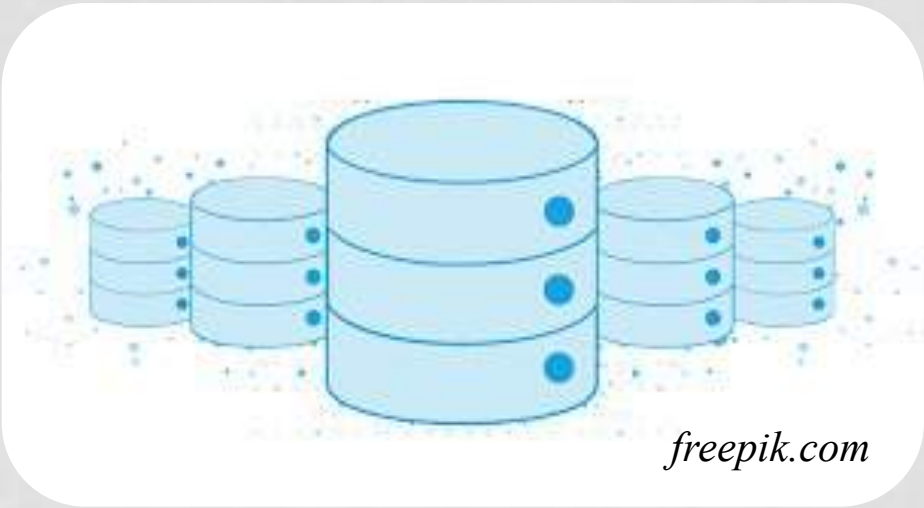
ДИСЦИПЛИНА: ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

ДЛЯ СТУДЕНТОВ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ
6B06105 «DATA SCIENCE»

СТАРШИЙ ПРЕПОДАВАТЕЛЬ КЛЮЕВА Е.Г.

ПЛАН ЛЕКЦИИ

1. Классификация структур данных;
2. Принцип независимости от данных;
3. Классификация моделей данных.



СТРУКТУРЫ ДАННЫХ

Понятие ***«данные»*** в концепции баз данных - это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы.

Примеры данных: Петров Николай Степанович, $100 и т.д. Данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру, то есть осознает их смысловое содержание.

СТРУКТУРЫ ДАННЫХ

Структура данных - это организационная схема данных, в соответствии с которой они упорядочены, с тем, чтобы их можно было интерпретировать и выполнять над ними определенные операции.

freepik.com

СТРУКТУРЫ ДАННЫХ

Структуры данных могут быть:

- однородными (все элементы данных представлены записями одного типа);
- неоднородными (элементами одной структуры могут являться записи разных типов).

freepik.com

СТРУКТУРЫ ДАННЫХ

Пример неоднородной записи - совокупность сведений о некотором студенте.

Объект "студент" обладает свойствами:

- "личный номер" - характеризуется целым положительным числом;
- "фамилия-имя-отчество" - характеризуется строкой символов;
- "группа" - характеризуется строкой символов и т.д.

СТРУКТУРЫ ДАННЫХ

Структуры данных предоставляют различный доступ к своим элементам:

- в одних структурах доступ возможен к любому элементу;

- в других структурах доступ возможен к строго определенному элементу.

freepik.com

ПРЕДСТАВЛЕНИЕ ДАННЫХ В ПАМЯТИ ПК

Различают структуры хранения, использующие в памяти компьютера:

- последовательное представление данных;
- связанное представление данных.

ПРЕДСТАВЛЕНИЕ ДАННЫХ В ПАМЯТИ ПК

При *последовательном представлении* данные в памяти компьютера размещаются в соседних последовательно расположенных ячейках.

При этом физический порядок следования записей полностью соответствует логическому порядку, определяемому логической структурой, то есть логическая структура поддерживается физическим порядком следования данных.

ПРЕДСТАВЛЕНИЕ ДАННЫХ В ПАМЯТИ ПК

@rec	+0	+1	+21	+29	+37	+38	+39
	24	Иванов В.И.	АП	54	4	5	5

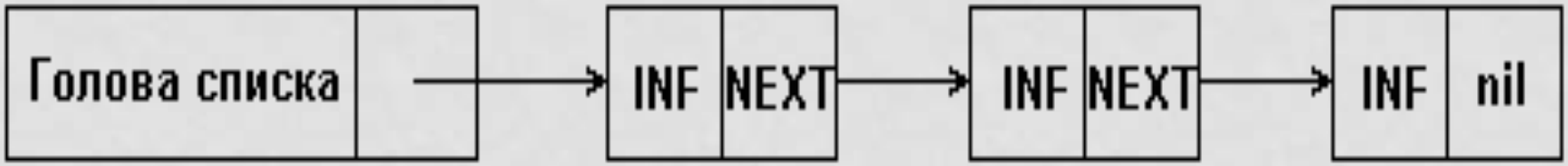
Представление записи в виде последовательности полей

ПРЕДСТАВЛЕНИЕ ДАННЫХ В ПАМЯТИ ПК

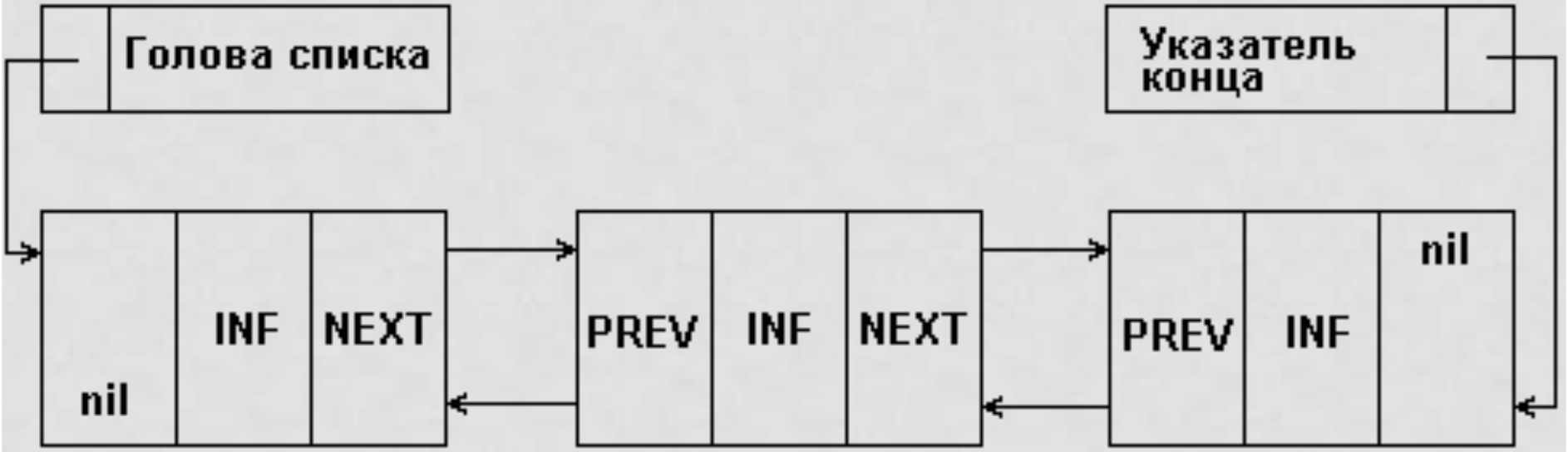
При *связанном представлении* в каждой записи предусматривается дополнительное поле, в котором размещается указатель (ссылка).

В памяти компьютера записи располагаются в любых свободных ячейках и связываются между собой указателями, указывающими на место расположения записи, логически следующей за данной.

СТРУКТУРА ОДНОСВЯЗНОГО СПИСКА



СТРУКТУРА ДВУХСВЯЗНОГО СПИСКА



ПРЕДСТАВЛЕНИЕ ЗАПИСИ В ВИДЕ СВЯЗНОГО СПИСКА

СТРУКТУРЫ ДАННЫХ

Структуры данных делятся на:
- линейные;
- нелинейные.

СТРУКТУРЫ ДАННЫХ

К линейным структурам относят:

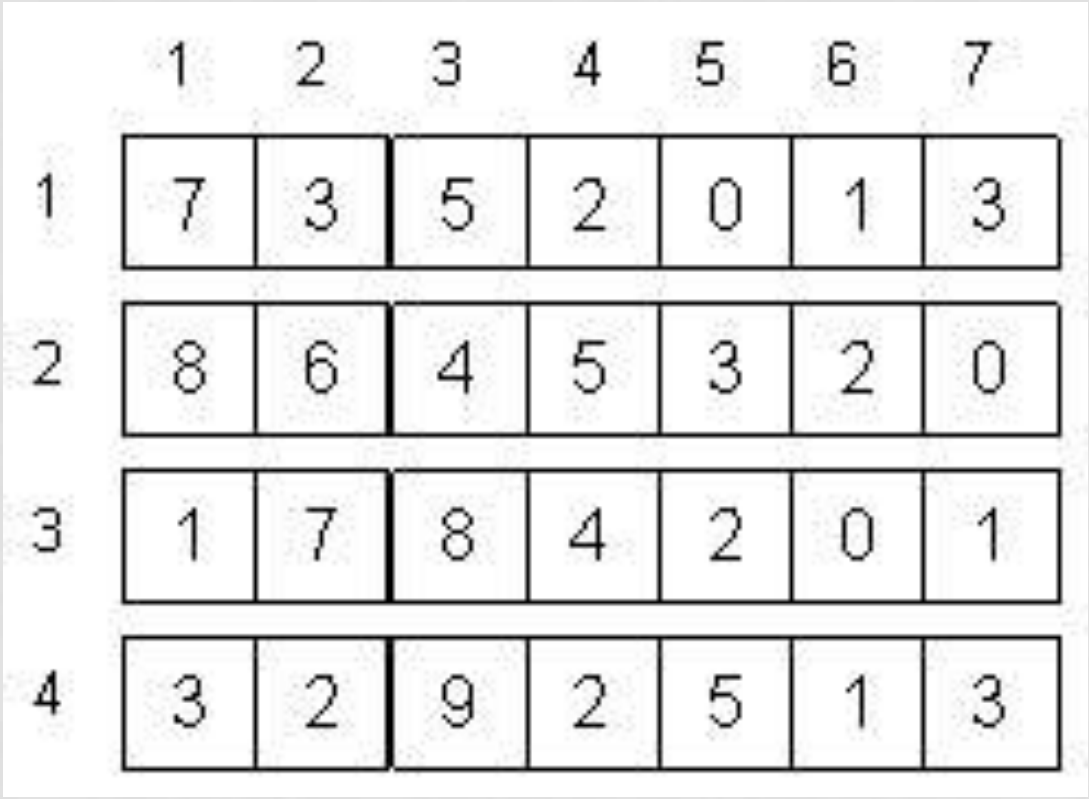
- массив;
- стек;
- очередь;
- таблица.

freepik.com

ЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ

Одномерный массив

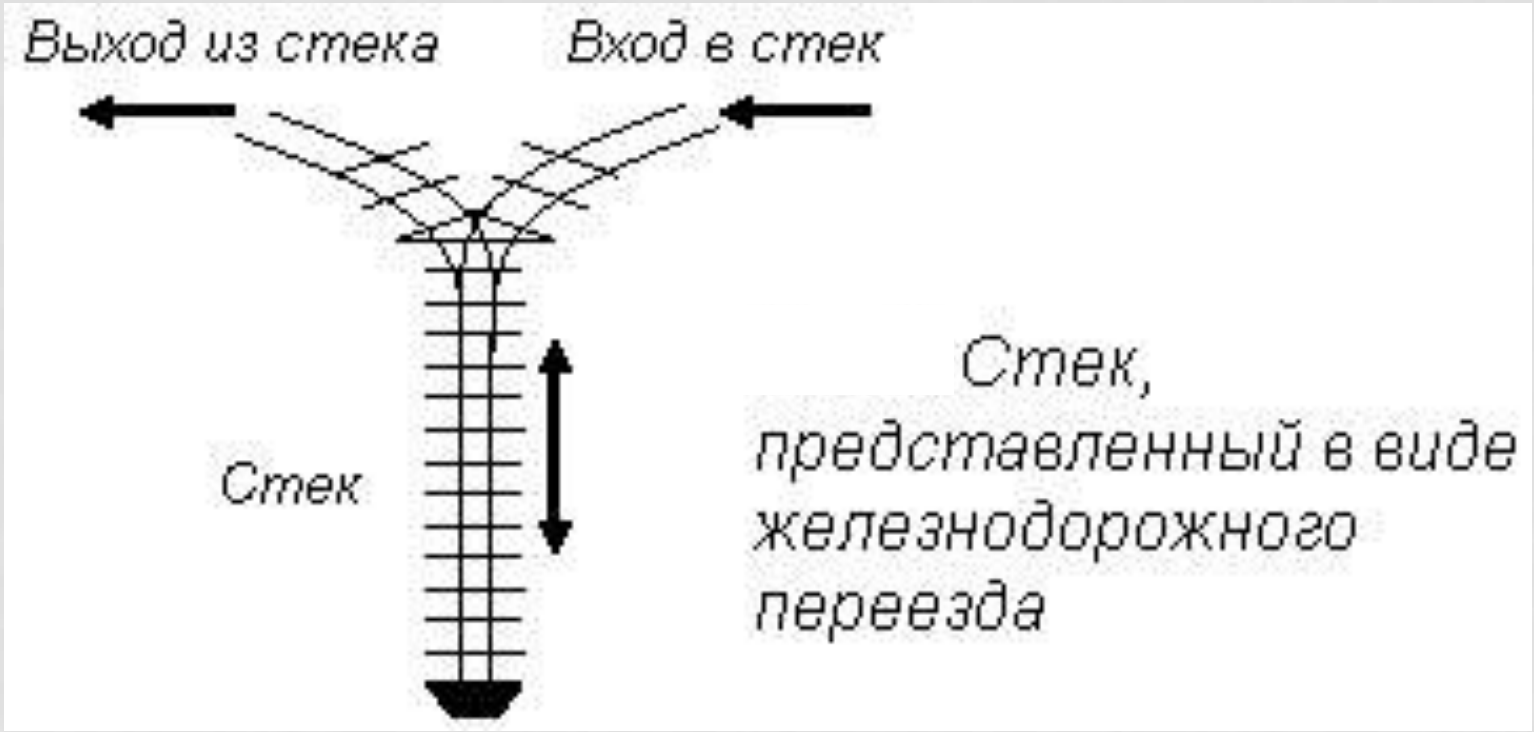
ЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ



	1	2	3	4	5	6	7
1	7	3	5	2	0	1	3
2	8	6	4	5	3	2	0
3	1	7	8	4	2	0	1
4	3	2	9	2	5	1	3

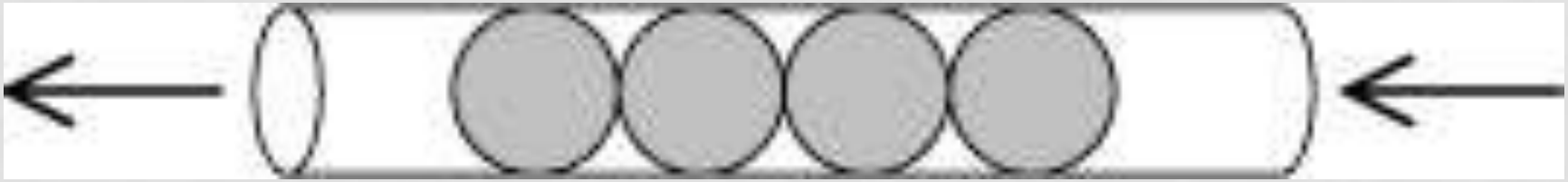
Двумерный массив

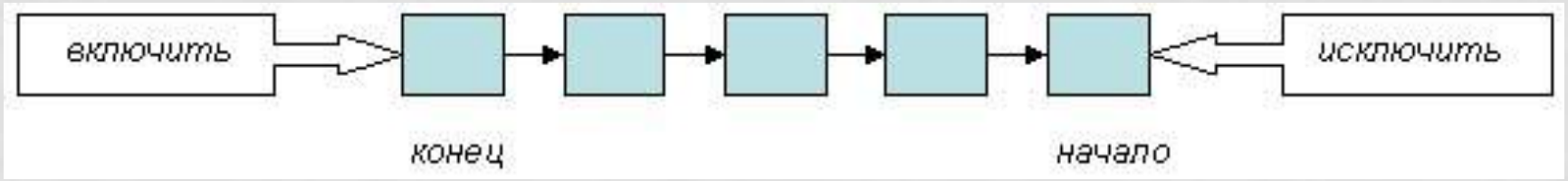
ЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ



Стек, представленный в виде железнодорожного переезда

ЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ





Очередь

ЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ

Номер	Автор	Название	Год	Полка
0001	Беляев А.Р.	Человек-амфибия	1987	5
0002	Кервуд Д.	Бродяги Севера	1991	7
0003	Тургенев И.С.	Повести и рассказы	1982	1
0004	Олеша Ю.К.	Избранное	1987	5
0005	Беляев А.Р.	Звезда КЭЦ	1990	5
0006	Тынянов Ю.Н.	Кюхля	1979	1
0007	Толстой Л.Н.	Повести и рассказы	1986	1
0008	Беляев А.Р.	Избранное	1994	7

Таблица

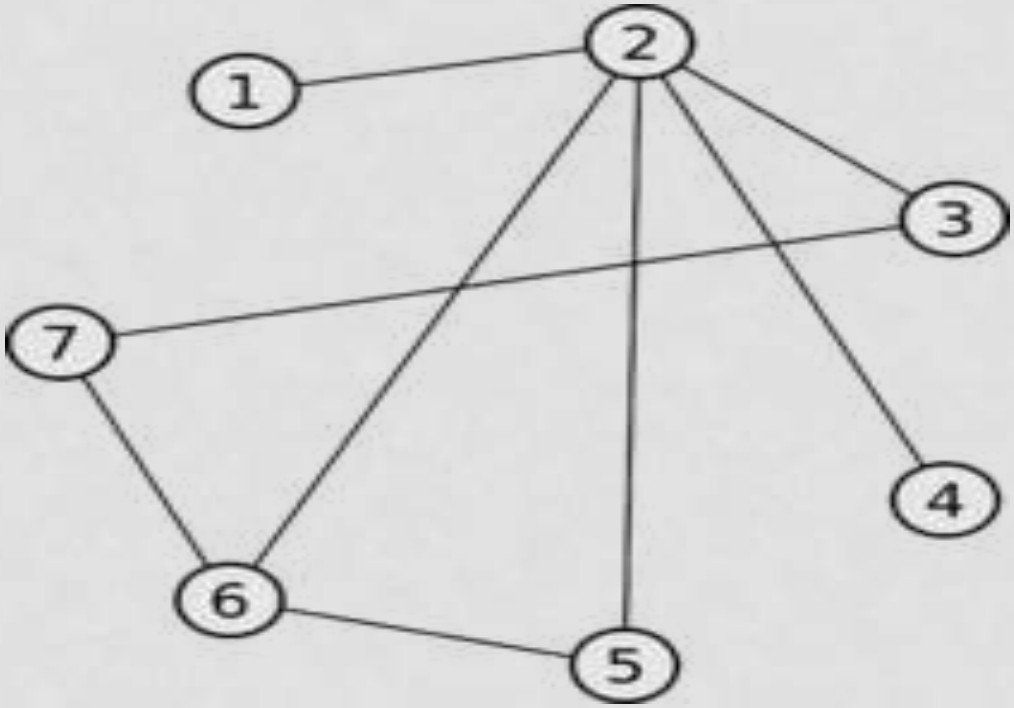
НЕЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ

В нелинейных структурах связь между элементами структуры (записями) определяется отношениями подчинения или какими-либо логическими условиями.

К нелинейным структурам принадлежат:

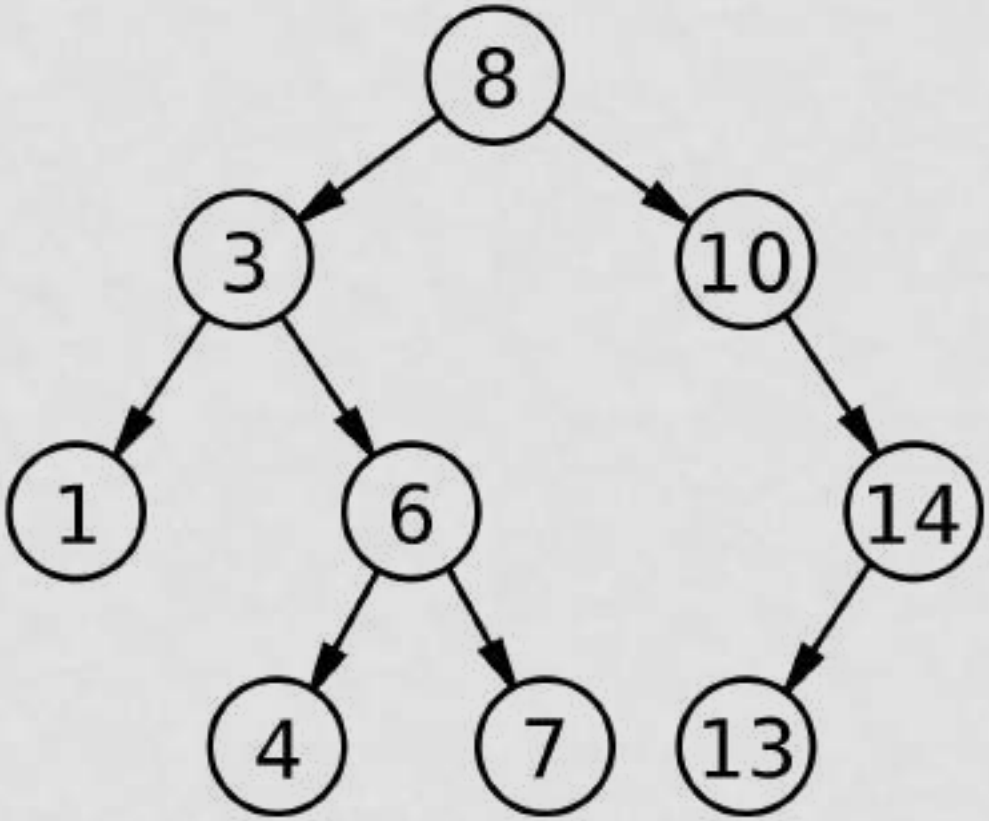
- деревья;
- графы;
- многосвязные списки и списковые структуры.

НЕЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ



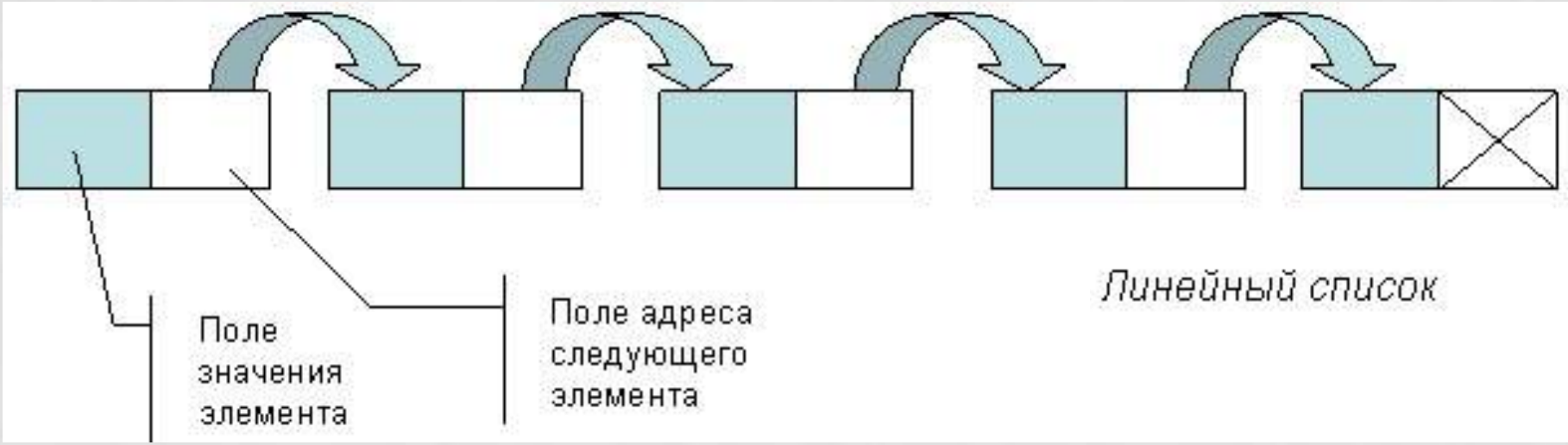
Граф

НЕЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ



Дерево

НЕЛИНЕЙНЫЕ СТРУКТУРЫ ДАННЫХ



ПРИНЦИП НЕЗАВИСИМОСТИ ОТ ДАННЫХ

При разработке структур данных должен обеспечиваться *принцип независимости от данных*.

Различают два типа независимости от данных:

- логическую;
- физическую.

freepik.com

ПРИНЦИП НЕЗАВИСИМОСТИ ОТ ДАННЫХ

Физическая независимость от данных означает, что любые изменения в физическом расположении данных (переход на новую файловую систему, структуру хранения, изменения индекса или способа хэширования) или техническом обеспечении БД (переход на другое устройство хранения) не должны отражаться на логических структурах и прикладных программах, то есть не должны вызывать в них изменений.

Пользователи могут заметить изменения только за счет изменения общей производительности системы.

ПРИНЦИП НЕЗАВИСИМОСТИ ОТ ДАННЫХ

Логическая независимость от данных означает, что изменения в структурах хранения (добавление или удаление новых объектов БД, их атрибутов или связей) не должны вызывать изменения в логических структурах данных и в прикладных программах.

Пользователи, для которых эти изменения предназначались, должны о них знать, но важно, чтобы другие пользователи даже и не подозревали о них.

ОСОБЫЕ ВИДЫ ДАННЫХ

Соблюдение принципа независимости данных позволяет использовать особые виды данных:

- виртуальные;
- прозрачные.

freepik.com

ОСОБЫЕ ВИДЫ ДАННЫХ

Виртуальные данные существуют лишь на логическом уровне.

Пользователю эти данные представляются реально существующими, и он оперирует ими при необходимости.

Каждый раз при обращении к этим данным система определенным образом их генерирует на основании других данных, физически существующих в системе.

Примером могут служить представления.

ОСОБЫЕ ВИДЫ ДАННЫХ

Прозрачные данные представляются несуществующими на логическом уровне.

Это позволяет скрыть от пользователя многие сложные механизмы, используемые при преобразовании логических структур данных в физические (примером могут служить индексы).

freepik.com

МОДЕЛИ ДАННЫХ

Под **моделью данных** понимают некоторую формальную теорию представления и обработки данных, включающую методы описания типов и логических структур данных (аспект структуры), методы манипулирования данными (аспект манипуляции) и методы описания и поддержки целостности данных (аспект целостности).

МОДЕЛИ ДАННЫХ

Модель данных – это способ моделировать, инструмент.

Модель базы данных – это результат использования этого способа для проектирования базы данных.

МОДЕЛИ ДАННЫХ

Любая модель данных содержит три компоненты:

1. структурная составляющая;
2. манипуляционная составляющая;
3. целостная составляющая.

МОДЕЛИ ДАННЫХ

Структурная составляющая описывает точку зрения пользователя на представление данных или набор правил, по которым может быть построена база данных.

МОДЕЛИ ДАННЫХ

Манипуляционная составляющая определяет набор допустимых операций, выполняемых на структуре данных.

Модель данных предполагает, как минимум, наличие *языка определения данных* (DDL), описывающего структуру хранения данных, и *языка манипулирования данными* (DML), включающего операции извлечения и модификации (вставка, удаление, изменение) данных.

МОДЕЛИ ДАННЫХ

Целостная составляющая определяет механизм поддержания соответствия данных предметной области на основе формально описанных правил или набор ограничений поддержки целостности данных, гарантирующих корректность используемых данных.

freepik.com

МОДЕЛИ ДАННЫХ

Цель построения модели данных заключается в представлении данных в понятном для разработчиков и пользователей БД виде.

freepik.com

МОДЕЛИ ДАННЫХ

Элемент данных (поле) – наименьшая поименованная единица данных. Используется для представления значения атрибута.

Запись – поименованная совокупность полей. Используется для представления совокупности атрибутов сущности (записи о сущности).

Экземпляр записи – запись с конкретными значениями полей.

МОДЕЛИ ДАННЫХ

Агрегат данных – поименованная совокупность элементов данных внутри записи, которую можно рассматривать как единое целое.

Файл – поименованная совокупность экземпляров записей одного типа. Используется для представления однородного набора сущностей.

Набор файлов – поименованная совокупность файлов, обрабатываемых в системе. Используется для представления нескольких наборов сущностей.

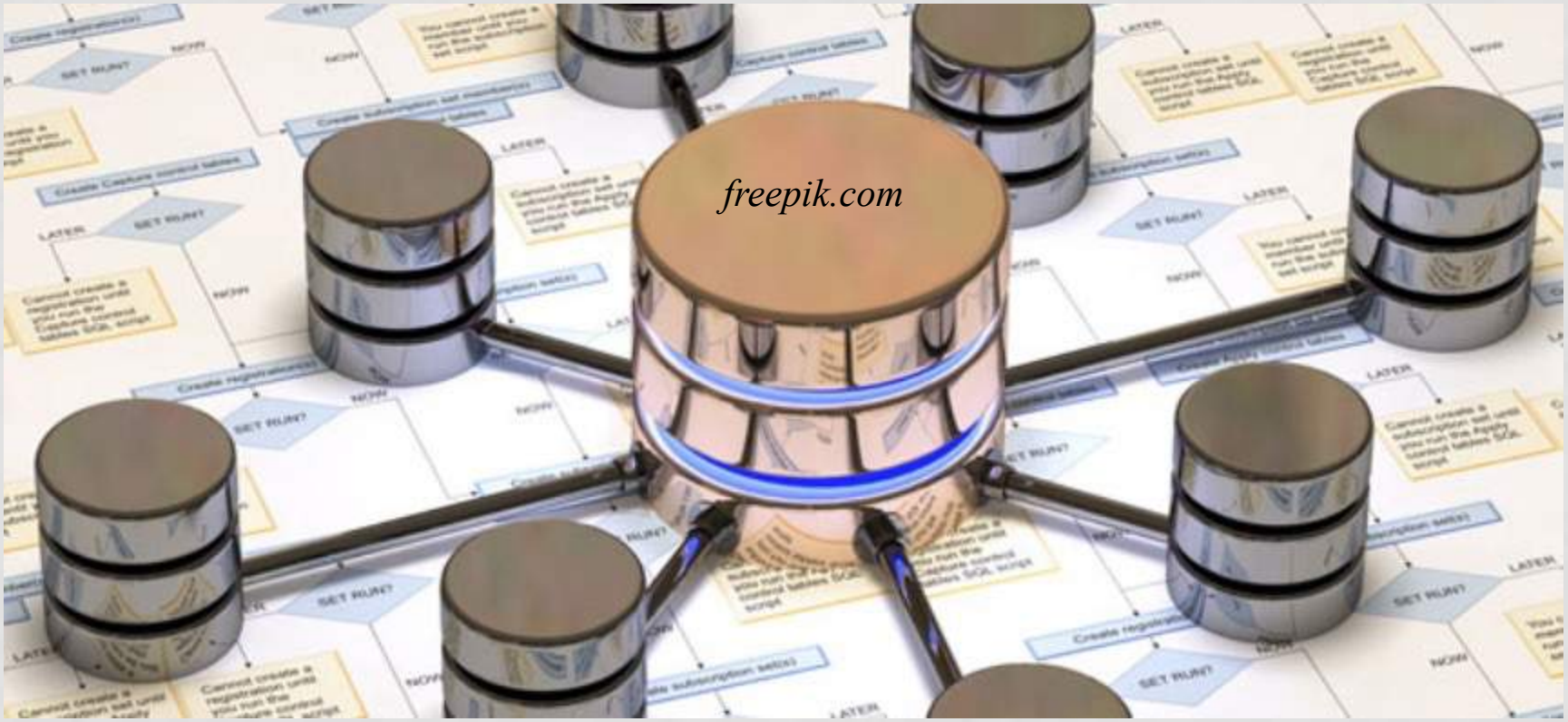
МОДЕЛИ ДАННЫХ

Группа – это поименованная совокупность элементов данных или элементов данных и других групп.

Групповое отношение – поименованное бинарное отношение, заданное на двух множествах экземпляров рассматриваемых групп. По характеру бинарных связей различают групповые отношения вида 1:1, 1:M, M:1, M:N. Пары чисел называют ***коэффициентами группового отношения***. В групповом отношении один член группы назначается владельцем отношения, другой – членом.

МОДЕЛИ ДАННЫХ

База данных – поименованная совокупность экземпляров групп и групповых отношений.



МОДЕЛИ ДАННЫХ

Модель данных описывается следующим образом:

- определяются типы и характеристики логических структур данных (полей, записей, файлов);
- описываются правила составления структур более общего типа из структур более простых типов;

МОДЕЛИ ДАННЫХ

- описываются возможные действия над структурами и правила их выполнения, включающие:
  - основные элементарные операции над данными;
  - обобщенные операции (процедуры);
  - средства контроля относительно простых условий корректности ввода данных (ограничения);
  - средства контроля сколь угодно сложных условий корректности выполнения определенных действий (правила).

МОДЕЛИ ДАННЫХ

В качестве основных элементарных операций обычно рассматриваются следующие: поиск записи с заданным значением ключа, чтение нужной записи, добавление записи, корректировка, удаление.

В моделях данных также предусматриваются специальные операции для установления групповых отношений.

МОДЕЛИ ДАННЫХ

В процессе исторического развития в СУБД использовались следующие модели данных:

- иерархическая;
- сетевая;
- реляционная.

В последние годы появились и стали внедряться следующие модели данных:

- пост-реляционная;
- многомерная;
- объектно-ориентированная;
- документоориентированная.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Является первой и появилась вместе с наиболее известной СУБД, ее реализующей – системой IMS (Information Management System) разработки компании IBM.

Эта СУБД была выпущена в 1968 году и существует и развивается до сих пор.

Данные организуются в виде иерархии. Основными понятиями данной модели являются *база данных, сегмент и поле*.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Поле (атрибут, элемент данных) – минимальная единица данных, которая представляет собой неделимое целое.

То есть, например, если нужно хранить в поле адрес, то выделить в нем элементы (например, регион, город, улицу) и получать доступ к ним средствами СУБД невозможно, это можно реализовывать только на уровне прикладной программы.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Набор полей образует ***сегмент (или запись)***.

В сегменте должен быть определен ключ – поле или набор полей, значение которого может однозначно идентифицировать конкретный экземпляр сегмента.

Сегменты образуют между собой связи типа «родитель-потомок». При этом у сегмента может быть произвольное количество потомков, но только один родитель. В вершине иерархии корневой сегмент - без родителя. Все сегменты в совокупности образуют древовидную структуру, которая и является базой данных.

Схема базы данных – это набор таких деревьев.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Реализация групповых отношений в иерархической модели, как и в сетевой, может осуществляться с помощью указателей и представляется в виде графа.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

1. Групповые отношения являются отношениями соподчиненности.

Группа (запись) – владелец отношения имеет подчиненные группы – члены отношений. Исходная группа называется предком, подчиненная – потомком.

2. Групповые отношения образуют иерархическую структуру, которую можно описать как ориентированный граф следующего вида:

– имеется единственная особая вершина (соответствующая группе), называемая корнем, в которую не заходит ни одно ребро (группа не имеет предков);

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

– во все остальные вершины входит только одно ребро (все остальные группы имеют одного предка), а исходит произвольное количество ребер (группы имеют произвольное количество потомков);

– отсутствуют циклы.

3. Иерархическая модель данных может представлять совокупность нескольких деревьев. В терминологии иерархической модели деревья, описывающие структуру данных, называются деревьями описания данных, а сами структурированные данные (база данных) – деревьями данных.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Для групповых отношений в иерархической модели обеспечивается *автоматический режим включения и фиксированное членство*.

Это означает, что для запоминания любой некорневой записи в БД должна существовать ее родительская запись.

При удалении родительской записи автоматически удаляются все подчиненные.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

ДОБАВИТЬ в базу данных новую запись. Для корневой записи обязательно формирование значения ключа.

ИЗМЕНИТЬ значение данных предварительно извлеченной записи. Ключевые данные не должны подвергаться изменениям.

УДАЛИТЬ некоторую запись и все подчиненные ей записи.

ИЗВЛЕЧЬ:

- извлечь корневую запись по ключевому значению, допускается также последовательный просмотр корневых записей;

- извлечь следующую запись (следующая запись извлекается в порядке левостороннего обхода дерева).

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Поддерживается только целостность связей между владельцами и членами группового отношения (никакой потомок не может существовать без предка).

Как уже отмечалось, не обеспечивается автоматическое поддержание соответствия парных записей, входящих в разные иерархии.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Особенностью реализации *операций поиска* в иерархической модели является то, что операция всегда начинает поиск с корневой вершины и специфицирует иерархический путь (последовательность связанных вершин) от корня до вершины, экземпляры которой удовлетворяют условиям поиска.

Необходимо отметить, что программы, реализующие операции иерархической модели, существенно проще, чем аналогичные программы для сетевой модели, т.к. здесь много легче осуществлять навигацию по структуре.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

К достоинствам относятся эффективное использование памяти ЭВМ и неплохие показатели времени выполнения основных операций над данными. Иерархическая модель удобна для работы с иерархически упорядоченной информацией.

Недостатком является ее громоздкость для обработки информации с достаточно сложными логическими связями, а также сложность понимания для обычного пользователя.

Примеры иерархических СУБД: IMS, TDMS, Mark IV MultiAccess Retrieval System, System 2000, Caché .

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

СУБД, реализующие сетевую модель данных, появились почти одновременно с иерархическими СУБД в 1971 году. Наиболее известная СУБД, реализующая архитектуру CODASYL, IDMS (принадлежит компании Computer Associates и продается под наименованием CA IDMS/DB).

Сетевую модель данных можно считать расширением иерархической модели.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Реализация групповых отношений в сетевой модели осуществляется с использованием указателей (адресов связи или ссылок), которые устанавливают связь между владельцем и членом группового отношения.

Запись может состоять в отношениях разных типов (1:1, 1:N, M:N).

freepik.com

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Если один из вариантов установления связи 1:1 очевиден (в запись – владелец отношения, поля которой соответствуют атрибутам сущности, включается дополнительное поле – указатель на запись – член отношения), то возможность представления связей 1:N и M:N таким же образом весьма проблематична.

Поэтому наиболее распространенным способом организации связей в сетевых СУБД является введение дополнительного типа записей, полями которых являются указатели.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Группа может быть членом более чем одного группового отношения. В этом случае вводится несколько дополнительных групп-указателей, а в группе – владельце отношений вводится несколько полей – указателей на дополнительные группы.

Тогда множество записей (групп) и связей между ними образует некую сетевую структуру (ориентированный граф общего вида). Вершинами графа являются группы; дугами графа, направленными от владельца к члену группового отношения, – связи между группами.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

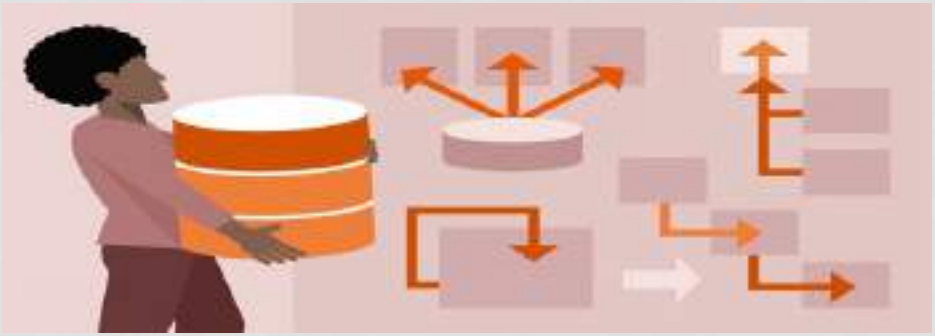
Сетевая модель данных поддерживает все необходимые операции над данными, реализованные как действия со списковыми структурами.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Режим включения подчиненных записей:

- ***автоматический*** - невозможно занести в БД запись без того, чтобы она была сразу же закреплена за неким владельцем;

- ***ручной*** - позволяет запомнить в БД подчиненную запись и не включать ее немедленно в экземпляр группового отношения. Эта операция позже инициируется пользователем).



freepik.com

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Режим исключения. Принято выделять три класса членства подчиненных записей в групповых отношениях:

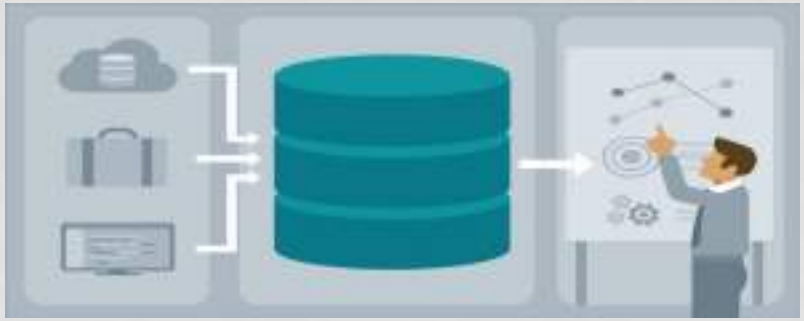
- ***фиксированное***. Подчиненная запись жестко связана с записью владельцем и ее можно исключить из группового отношения только удалив. При удалении записи-владельца все подчиненные записи автоматически тоже удаляются;

freepik.com

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Режим исключения. Принято выделять три класса членства подчиненных записей в групповых отношениях:

- ***обязательное***. Допускается переключение подчиненной записи на другого владельца, но невозможно ее существование без владельца. Для удаления записи-владельца необходимо, чтобы она не имела подчиненных записей с обязательным членством;



freepik.com

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Режим исключения. Принято выделять три класса членства подчиненных записей в групповых отношениях:

- ***необязательное***. Можно исключить запись из группового отношения, но сохранить ее в базе данных не прикрепляя к другому владельцу. При удалении записи-владельца ее подчиненные записи - необязательные члены сохраняются в базе, не участвуя более в групповом отношении такого типа.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

ДОБАВИТЬ - внести запись в БД и, в зависимости от режима включения, либо включить ее в групповое отношение, где она объявлена подчиненной, либо не включать ни в какое групповое отношение.

ВКЛЮЧИТЬ В ГРУППОВОЕ ОТНОШЕНИЕ - связать существующую подчиненную запись с записью-владельцем.

ПЕРЕКЛЮЧИТЬ - связать существующую подчиненную запись с другой записью-владельцем в том же групповом отношении.

ОБНОВИТЬ - изменить значение элементов предварительно извлеченной записи.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

ИЗВЛЕЧЬ - извлечь записи последовательно по значению ключа, а также используя групповые отношения - от владельца можно перейти к записям - членам, а от подчиненной записи к владельцу набора.

УДАЛИТЬ - убрать из БД запись. Если эта запись является владельцем группового отношения, то анализируется класс членства подчиненных записей. Обязательные члены должны быть предварительно исключены из группового отношения, фиксированные удалены вместе с владельцем, необязательные останутся в БД.

ИСКЛЮЧИТЬ ИЗ ГРУППОВОГО ОТНОШЕНИЯ - разорвать связь между записью-владельцем и записью-членом.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Как и в иерархической модели обеспечивается только поддержание целостности по ссылкам (владелец отношения - член отношения).

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Достоинством является возможность эффективной реализации по показателям затрат памяти и оперативности. Сетевая модель предоставляет большие возможности в смысле допустимости образования произвольных связей.

Недостатком является высокая сложность и жесткость схемы БД, построенной на ее основе, а также сложность для понимания и выполнения обработки информации в БД обычным пользователем. В сетевой модели данных ослаблен контроль целостности связей вследствие допустимости установления произвольных связей между записями.

Пример сетевых СУБД: IDS, IDMS, db VistaIII, IMAGE/3000, Caché, DBMS-10, СООБЗ Cerebrum.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Основными используемыми понятиями здесь также являются *поле*, *запись* и *файл*.

Структура записи определяет структуру таблицы, содержащей экземпляры соответствующей записи.

Столбцы таблицы представляют собой имена полей записи, строки таблицы – экземпляры записи.

Таким образом, понятие «таблица» здесь соответствует понятию «файл» модели данных.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Групповое отношение может представляться двумя способами.

При первом способе в таблицы, соответствующие группам – членам отношения, добавляются столбцы ключевых полей (атрибутов) другого члена отношения (связь описывается через ключевые атрибуты).

При втором способе групповое отношение определяется как дополнительная группа (дополнительная таблица). Столбцами этой дополнительной таблицы являются ключи групп – членов отношения. Таким образом, при любом способе соответствующая модель данных представляет собой совокупность структур таблиц.

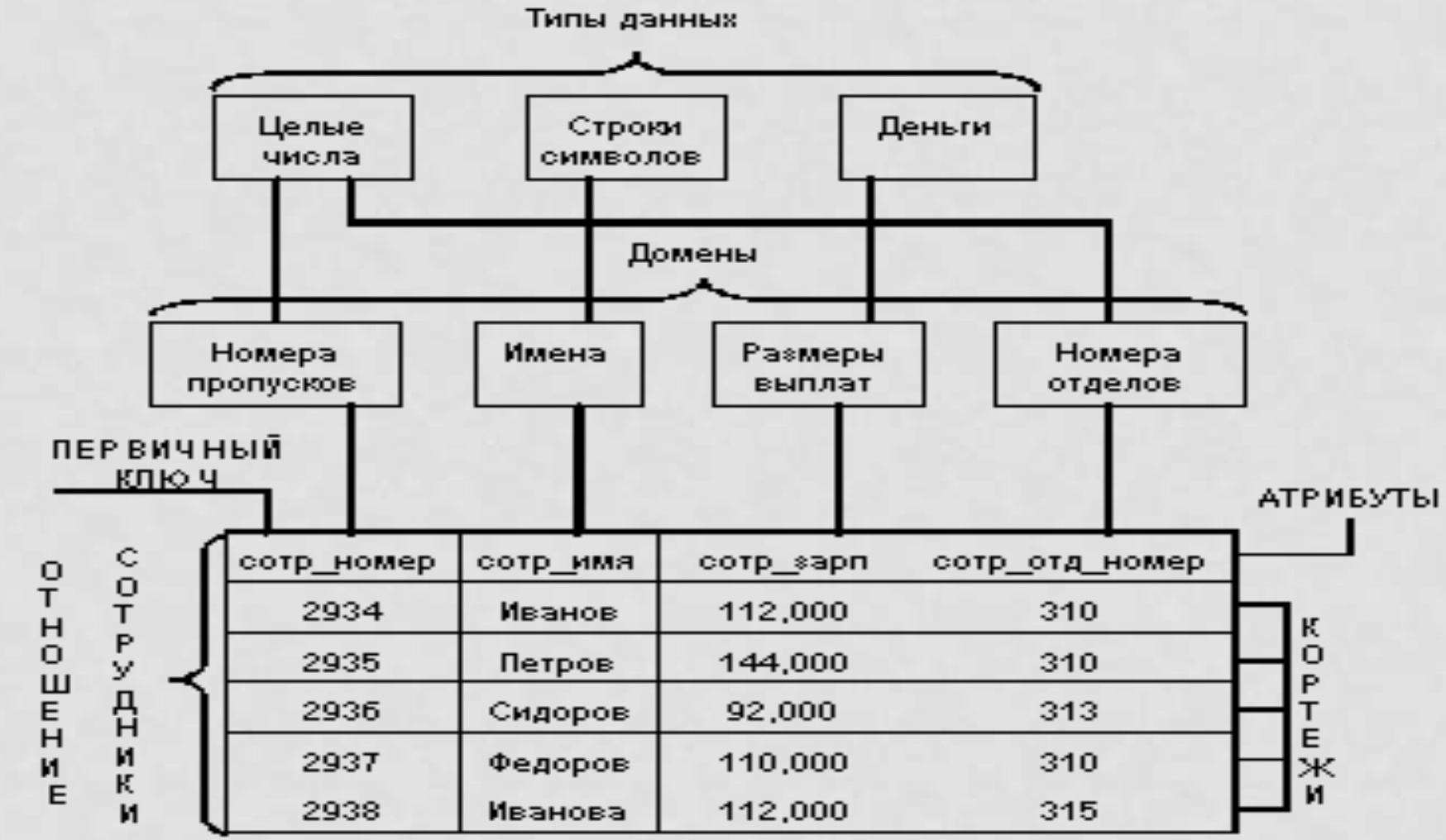
РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Для формального описания таблицы используется теоретико-множественное понятие отношения.

Список названий столбцов таблицы (имен полей записи, соответствующих атрибутам) именуют ***схемой отношения***.

Совокупность схем отношений, используемых для представления концептуальной модели, называется ***схемой реляционной базы данных***, а текущие значения соответствующих отношений – ***реляционной базой данных***.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ



РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Достоинство заключается в простоте, понятности и удобстве физической реализации на ЭВМ.

Недостатком является отсутствие стандартных средств идентификации отдельных записей и сложность описания иерархических и сетевых связей.

Примеры реляционных СУБД: IBM DB2 Universal Database, Oracle Database, Microsoft SQL Server, IBM Informix Dynamic Server, Sybase Adaptive Server, Borland InterBase, PostgreSQL, MySQL, MariaDB, SQLight.

ПОСТ-РЕЛЯЦИОННАЯ МОДЕЛЬ

Представляет собой расширенную реляционную модель, снимающую ограничение неделимости данных, хранящихся в записях таблиц.

Постреляционная модель данных допускает многозначные поля - поля, значения которых состоят из подзначений.

Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную таблицу.

В качестве языка запросов используется несколько расширенный SQL, позволяющий извлекать сложные объекты из одной таблицы без операций соединения.

ПОСТ-РЕЛЯЦИОННАЯ МОДЕЛЬ

Накладные

Номер_накладной	Номер_покупателя
0373	8723
8374	8232
7364	8723

Накладные_товары

Номер_накладной	Название_товара	Количество_товара
0373	Сыр	3
0373	Рыба	2
8374	Лимонад	1
8374	Сок	6
8374	Печенье	2
7364	Йогурт	1

Реляционная модель

Накладные

Номер_накладной	Номер_покупателя	Название_товара	Количество_товара
0373	8723	Сыр	3
		Рыба	2
8374	8232	Лимонад	1
		Сок	6
		Печенье	2
7364	8723	Йогурт	1

Пост-реляционная модель

ПОСТ-РЕЛЯЦИОННАЯ МОДЕЛЬ

Достоинства: по сравнению с реляционной моделью данные хранятся более эффективно, а при обработке не требуется выполнять операцию соединения данных из двух таблиц. На длину полей и количество полей в записях таблицы не накладывается требование постоянства. Это означает, что структура данных и таблиц имеют большую гибкость.

Недостатком является сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.

Примеры постреляционных СУБД: Bubba, Dasdb, Adabas, Pick и Universe.

МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ

Многомерность модели данных означает здесь многомерное логическое представление структуры информации и, вообще говоря, не связана с многомерностью визуализации.

Многомерные структуры представляются как гиперкубы данных.

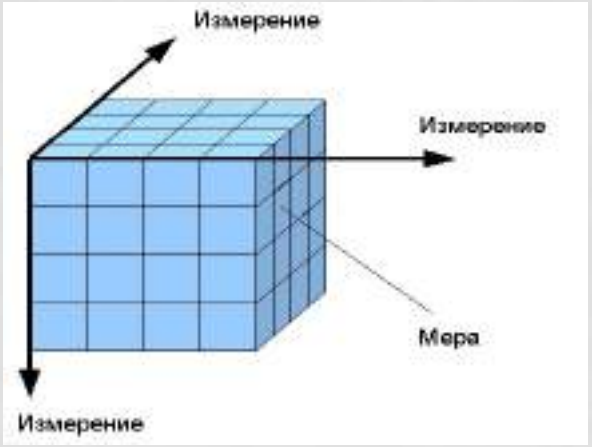
Каждая грань куба является размерностью.

Основными понятиями, используемыми в многомерных моделях данных, являются «*измерение*» (dimension) и «*ячейка*» (cell).

МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ

Измерение – упорядоченный набор значений, принимаемых конкретным параметром, соответствующий одной из граней гиперкуба.

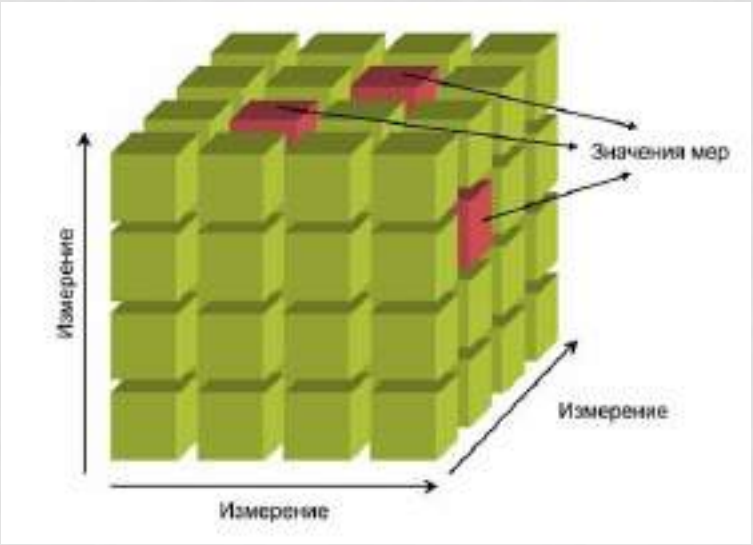
Например, год – 2018, 2019, 2020; область – Алматинская, Карагандинская, Южно-Казахстанская и т.д.



https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ

Ячейка или ***показатель*** – это поле, соответствующее атрибуту сущности, значение которого однозначно определяется фиксированным набором значений параметров (значениями «измерений», *например*, 2019 г., Карагандинская область).



https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

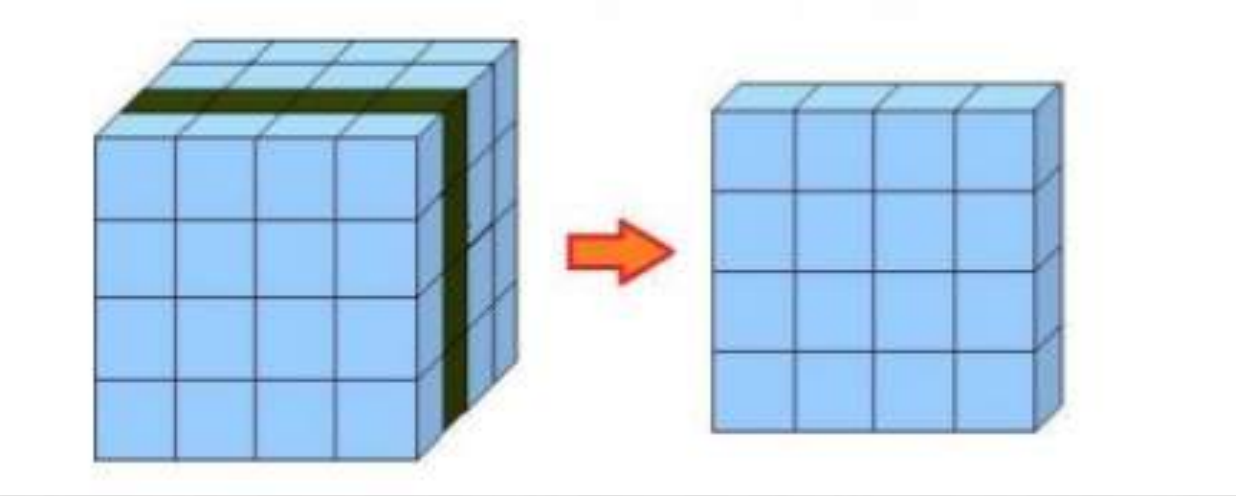
МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ

В многомерной модели данных определяется ряд дополнительных операций, среди которых можно выделить операции формирование среза, агрегации и поворота.

https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ

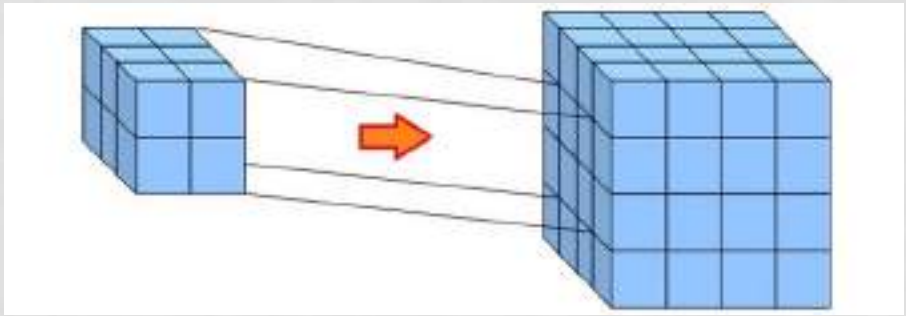
При ***формировании среза*** пользователю по его запросу предоставляется некоторое подмножество гиперкуба, полученное в результате фиксаций пользователем одного или нескольких значений параметров.

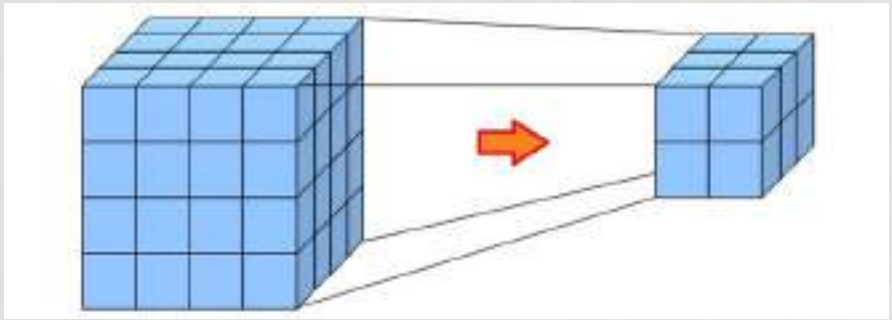


https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ

Операция «***агрегация***» обеспечивает переход к более общему представлению информации из гиперкуба пользователю, например суммируя значения показателей по всем значениям одного из параметров, допустим, по всем областям.

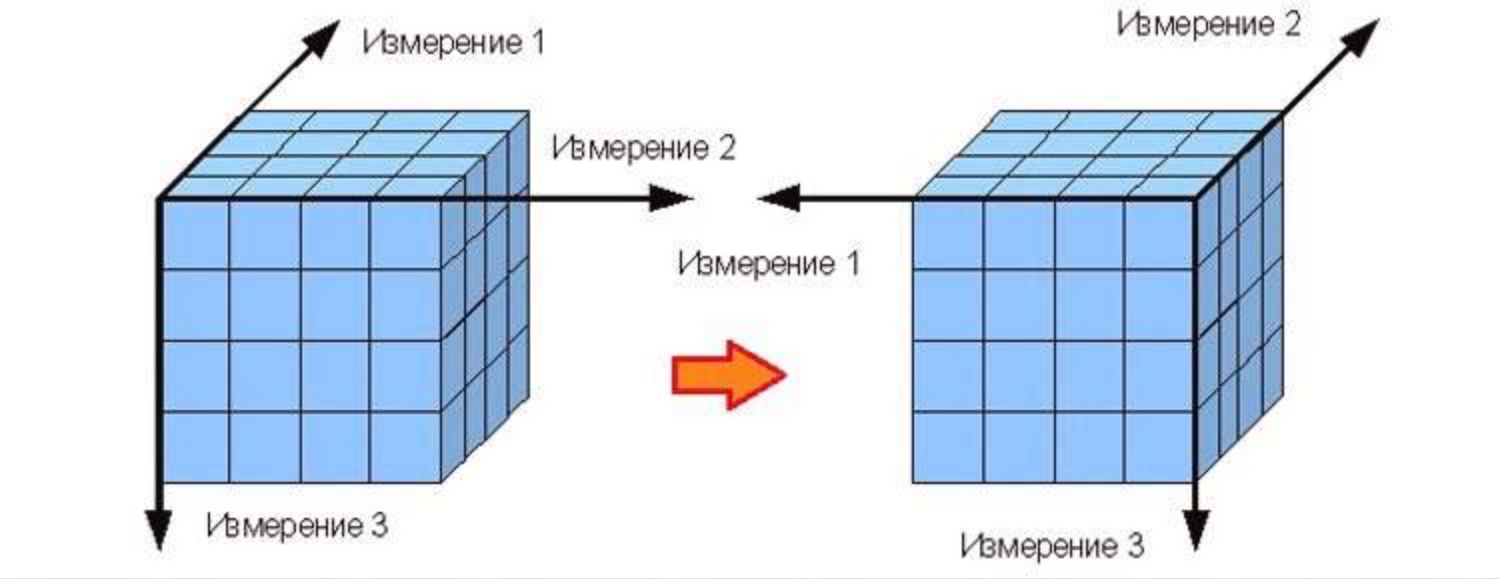




https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ

Поворот (вращение) куба дает пользователям возможность увидеть данные, сгруппированные по другим измерениям.



https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ

Достоинством является удобство и эффективность аналитической обработки больших объемов данных, связанных со временем.

Недостатком многомерной модели данных является ее громоздкость для простейших задач обычной оперативной обработки информации.

Примеры многомерных СУБД: Cache, Teradata, Essbase, Media Multi-matrix, Oracle Express Server, MS SQL Server.

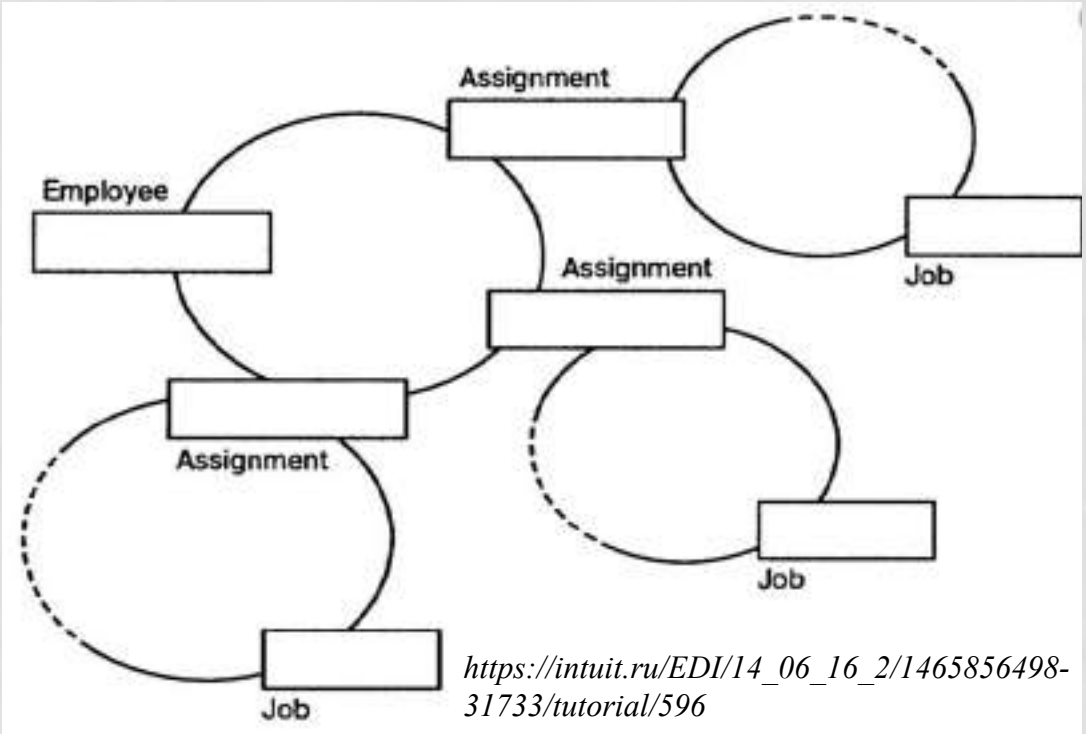
ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ

Объектно-ориентированные базы данных обычно рекомендованы для тех случаев, когда требуется высокопроизводительная обработка данных, имеющих сложную структуру.

ООСУБД позволяет работать с объектами баз данных так же, как с объектами в программировании в ООЯП. ООСУБД расширяет языки программирования, прозрачно вводя долговременные данные, управление параллелизмом, восстановление данных, ассоциированные запросы и другие возможности.

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ

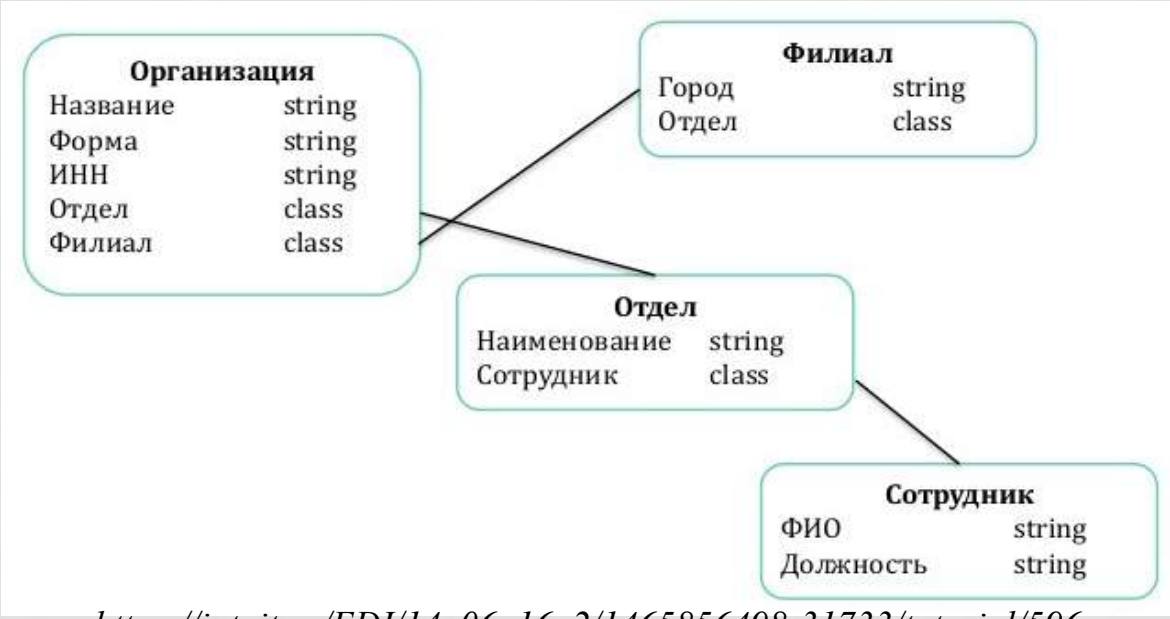
Структура объектно-ориентированной БД графически представима в виде дерева, узлами которого являются объекты.



https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ

Свойства объектов описываются некоторым стандартным типом данных (например, строковым - string) или типом, конструируемым пользователем (определяется как class).



https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ

Достоинством является возможность отображения информации о сложных связях объектов. Объектно-ориентированная модель данных позволяет идентифицировать отдельную запись базы данных и определять функции их обработки.

Недостатками являются высокая понятийная сложность, неудобство обработки данных и низкая скорость выполнения запросов.

Примеры ООСУБД: O2, ORION, GemStone и Iris.

ДОКУМЕНТООРИЕНТИРОВАННАЯ МОДЕЛЬ

Документоориентированная модель специально предназначена для хранения иерархических структур данных (документов) и обычно реализуемая с помощью подхода NoSQL.

В основе документоориентированных СУБД лежат документные хранилища (document store), имеющие структуру дерева (иногда леса).

ДОКУМЕНТООРИЕНТИРОВАННАЯ МОДЕЛЬ

Структура дерева начинается с корневого узла и может содержать несколько внутренних и листовых узлов.

Листовые узлы содержат данные, которые при добавлении документа заносятся в индексы, что позволяет даже при достаточно сложной структуре находить место (путь) искомых данных.

API для поиска позволяет находить по запросу документы и части документов.

ДОКУМЕНТООРИЕНТИРОВАННАЯ МОДЕЛЬ

В отличие от хранилищ типа ключ-значение, выборка по запросу к документному хранилищу может содержать части большого количества документов без полной загрузки этих документов в оперативную память.

Документы могут быть организованы (сгруппированы) в коллекции. Их можно считать отдалённым аналогом таблиц реляционных СУБД, но коллекции могут содержать другие коллекции. Хотя документы коллекции могут быть произвольными, для более эффективного индексирования лучше объединять в коллекцию документы с похожей структурой.

ДОКУМЕНТООРИЕНТИРОВАННАЯ МОДЕЛЬ

Примеры документоориентированных СУБД: CouchDB, Couchbase, MarkLogic, Mongo DB, eXist.

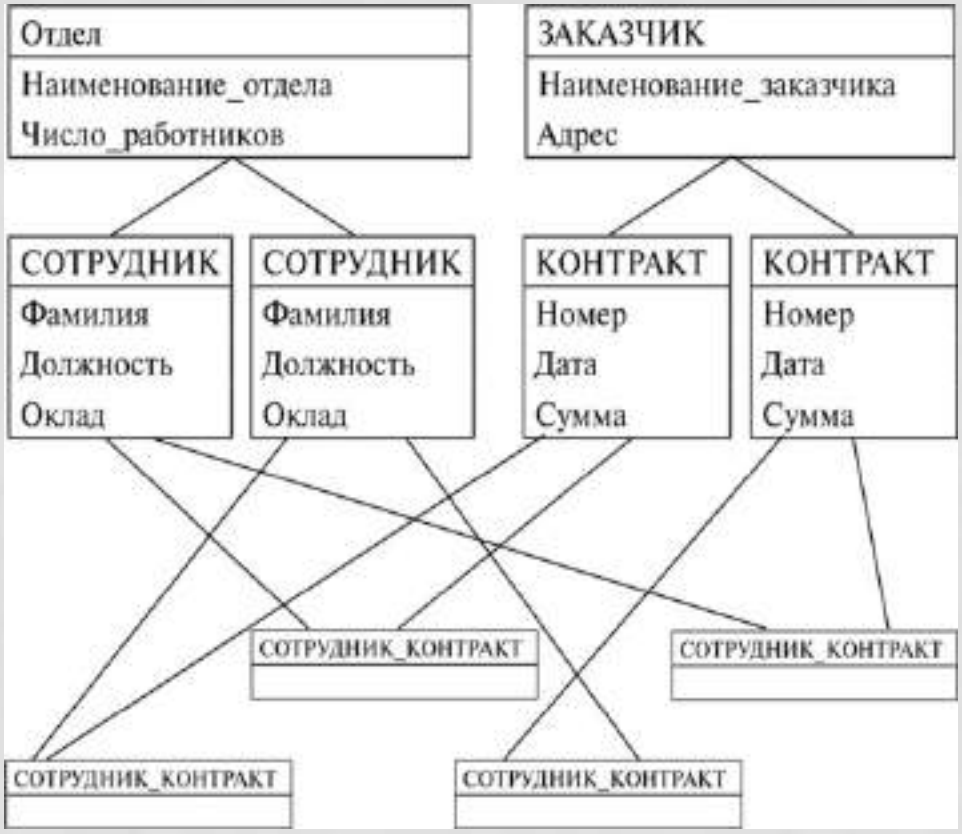
https://intuit.ru/EDI/14_06_16_2/1465856498-31733/tutorial/596

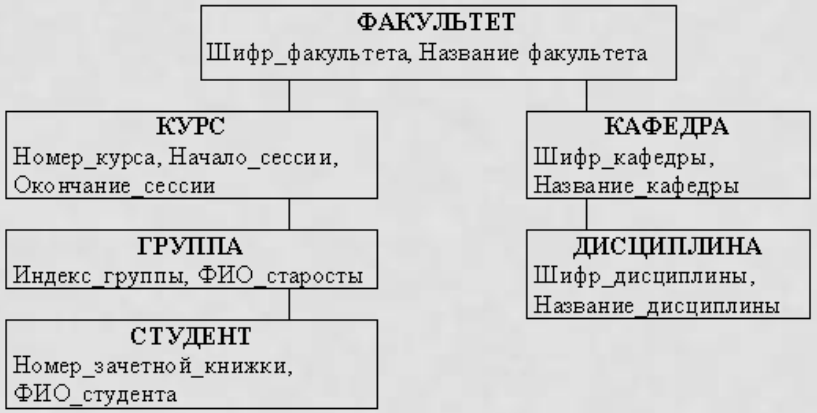
ЗАДАНИЯ ДЛЯ СРС

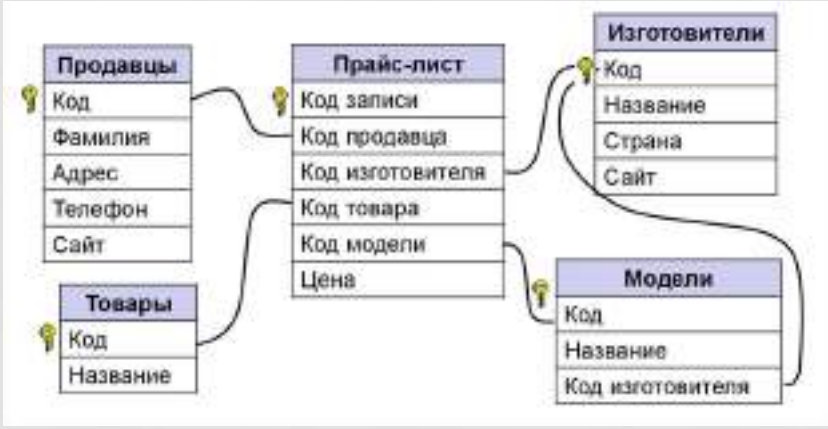
1. Что представляет собой структура данных?
2. Какие структуры данных были использованы вами ранее в программировании?
3. Объясните принципы независимости от данных.
4. Перечислите компоненты системы баз данных.
5. Что такое модель данных?
6. Перечислите составляющие модели данных.

ЗАДАНИЯ ДЛЯ СРС

7. Охарактеризуйте модели данных, представленные на рисунках



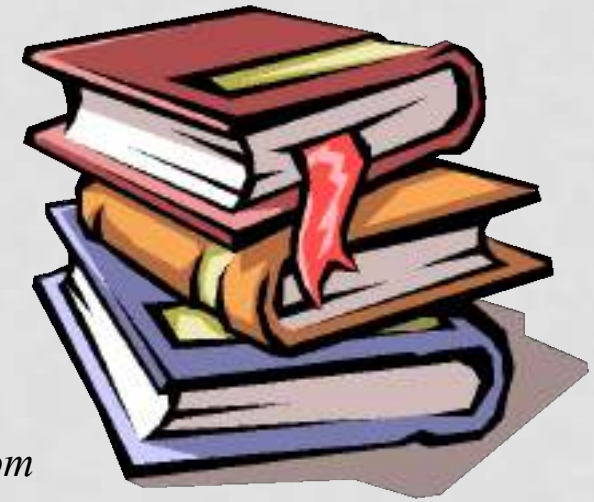




РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Т. Конноли, К. Бегг. Базы данных: Проектирование, реализация и сопровождение. Теория и практика [Текст] : учебное пособие / 3-е изд. - М. ; СПб. ; Киев : Вильямс, 2018. - 1440 с.

2. К.Дж. Дейт. Введение в системы баз данных. М., 2018. - 1328 с.



freepik.com

ЛЕКЦИЯ ОКОНЧЕНА.
БЛАГОДАРЮ ЗА ВНИМАНИЕ!