

**ТЕМА: ОСНОВНЫЕ ЭТАПЫ
ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ**

ДИСЦИПЛИНА: ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

ДЛЯ СТУДЕНТОВ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ
6B06105 «DATA SCIENCE»

СТАРШИЙ ПРЕПОДАВАТЕЛЬ КЛЮЕВА Е.Г.

ПЛАН ЛЕКЦИИ

1. Внешний уровень проектирования БД;
2. Концептуальное моделирование. ER-диаграмма;
3. Этапы физического проектирования;
4. Бизнес-модель этапа физического проектирования;
5. Файловые структуры, используемые для хранения информации в БД;
6. Аспекты классификации архитектур ИС на основе БД.

ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Целью проектирования на внешнем уровне является разработка *внемашинного* информационного обеспечения, которое включает систему входной (первичной) документации, характеризующую определенную предметную область, систему классификации и кодирования технико-экономической информации, а также перечень соответствующих выходных сообщений, которые нужно формировать с помощью БД.

ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Существуют два подхода к проектированию баз данных на внешнем уровне:

- 1) «от предметной области»;
- 2) «функциональный - от запроса».



ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Подход *«от предметной области»* состоит в том, что формируется внешнее информационное обеспечение всей предметной области без учета потребностей пользователей и прикладных программ.

Иногда этот подход называют еще *объектным* или *непроцессным*.



ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

При подходе «от запроса» основным источником информации о предметной области является изучение запросов пользователей и потребностей прикладных программ. Этот подход также называется *процессным* или *функциональным*.

При таком подходе БД проектируется для выполнения текущих задач управления без учета возможности расширения системы и возникновения новых задач управления.

ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Преимущество подхода «от предметной области» - его объективность, системность при отображении в ПО, стойкость информационной модели, возможность реализации большого количества прикладных программ и запросов, в том числе незапланированных при создании БД.

Недостатком этого подхода является значительный объем работ, которые необходимо выполнить при определении информации, подлежащей хранению в БД, что, соответственно, усложняет и увеличивает срок разработки проекта.

ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Недостатки функционального подхода заключаются в том, что он ориентирован на реализацию текущих требований пользователей и прикладных программ без учета перспектив развития системы. При его использовании могут возникнуть сложности в агрегации требований разных пользователей и прикладных программ.

Достоинства. При функциональном подходе значительно уменьшается трудоемкость проектирования, и поэтому возможно создать систему с высокими эксплуатационными характеристиками.

ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Процесс проектирования БД на внешнем уровне состоит из следующих работ:

- 1) определение функциональных задач предметной области, которые подлежат автоматизированному решению и для которой разрабатывается БД. Изучение и анализ предусматривают также выявление информационных потребностей и определения информационных потоков. Результатом такого изучения может быть перечень функциональных задач, которые должны решаться автоматизированным способом с использованием БД;

ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Процесс проектирования БД на внешнем уровне состоит из следующих работ:

2) изучение и анализ оперативных первичных документов. Изучив и проанализировав все оперативные документы (как внешние, так и внутренние), которые используются на входе каждой задачи, определяют, какие реквизиты этих документов нужно сохранять в БД;



ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Процесс проектирования БД на внешнем уровне состоит из следующих работ:

3) изучение нормативно-справочных документов. К такой документации принадлежат различные классификаторы, сметы, договоры, нормативы, законодательные акты по налоговой политике, плановая документация и т.п. Распределение и отдельный анализ оперативной и нормативно-справочной информации обусловлены технологически. В БД различаются технологии создания и ведения файлов условно-постоянной информации, размещенной в нормативно-справочной;

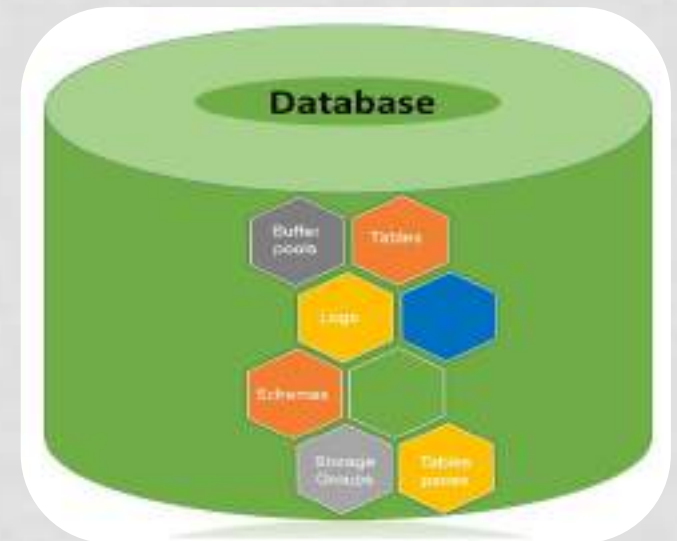
ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Процесс проектирования БД на внешнем уровне состоит из следующих работ:

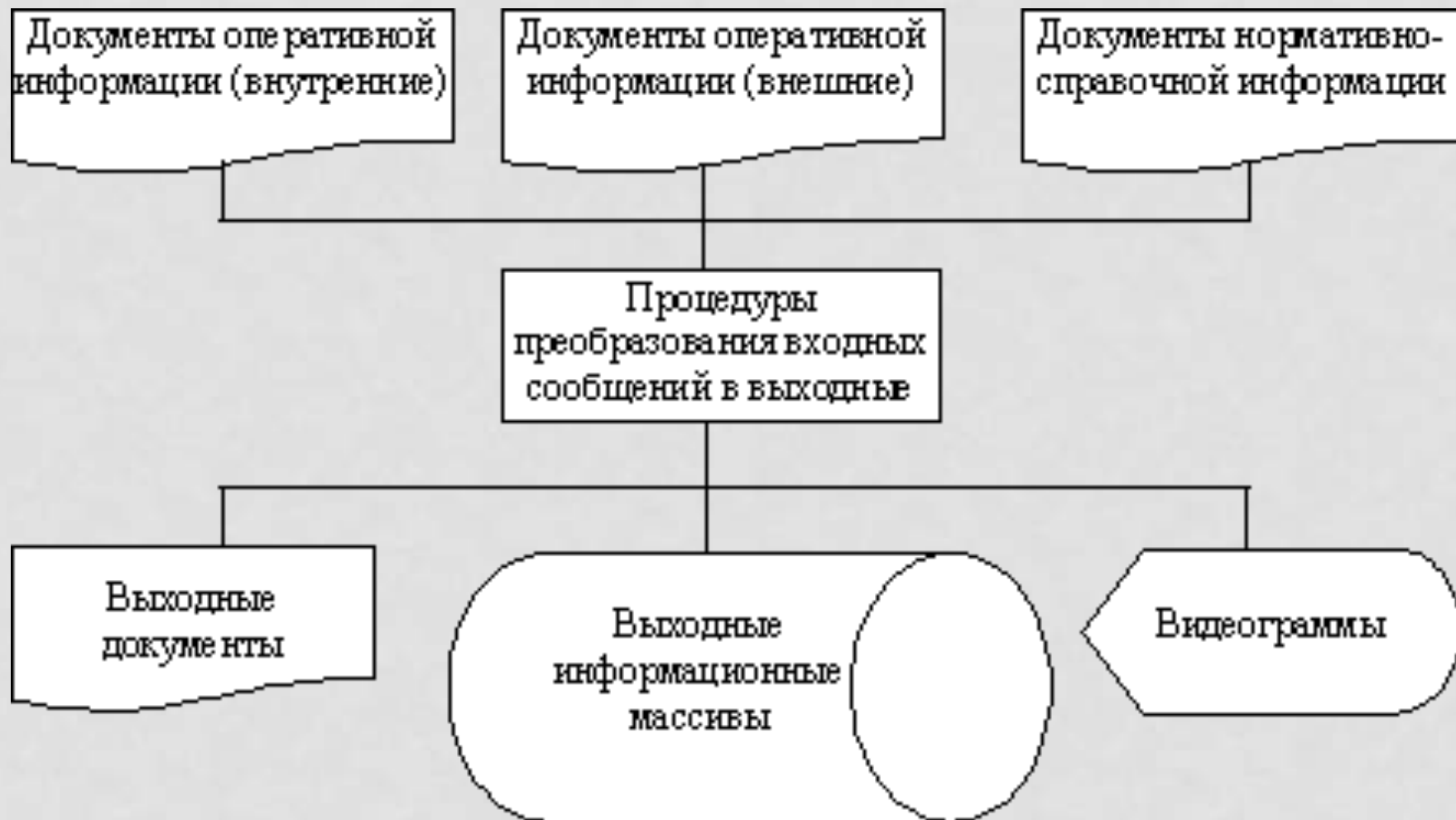
4) изучение процессов преобразования входных сообщений в выходные. Это необходимо для того, чтобы определить, которые из атрибутов входных сообщений нужно сохранять в БД для получения выходных сообщений. Кроме того, на этом этапе определяются те показатели, которые получают во время решения задачи в результате выполнения определенных вычислений. По каждому расчетному показателю следует определить алгоритм его формирования.

ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

При проведении проектных работ на внешнем уровне надо учитывать то, что для выполнения определенных функций в БД необходимо сохранять дополнительные данные, которые не отображены в документах (данные календаря, статистические данные и т.п.).



ОБОБЩЕННАЯ СХЕМА ПРОЕКТИРОВАНИЯ БД НА ВНЕШНЕМ УРОВНЕ



ВНЕШНИЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ БД

Результатом проектирования на внешнем уровне будет перечень атрибутов (реквизитов) оперативной и условно-постоянной информации, которые необходимо хранить в БД, с указанием источников их получения и формы представления.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Информационная и функциональная модели предметной области создаются на этапе анализа требований к БД и не содержат предположений о технологии реализации БД.

Они строятся независимо от выбираемой модели данных (сетевой, иерархической, реляционной, объектно-ориентированной, многомерной и т.д.), поддерживаемой СУБД, модели вычислений, программно-аппаратной платформы для БД.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Совокупность реалий (объектов) внешнего мира - объектов, о которых можно задавать вопросы и сведения о которых надо хранить в БД, - образует объектное *ядро предметной области*.

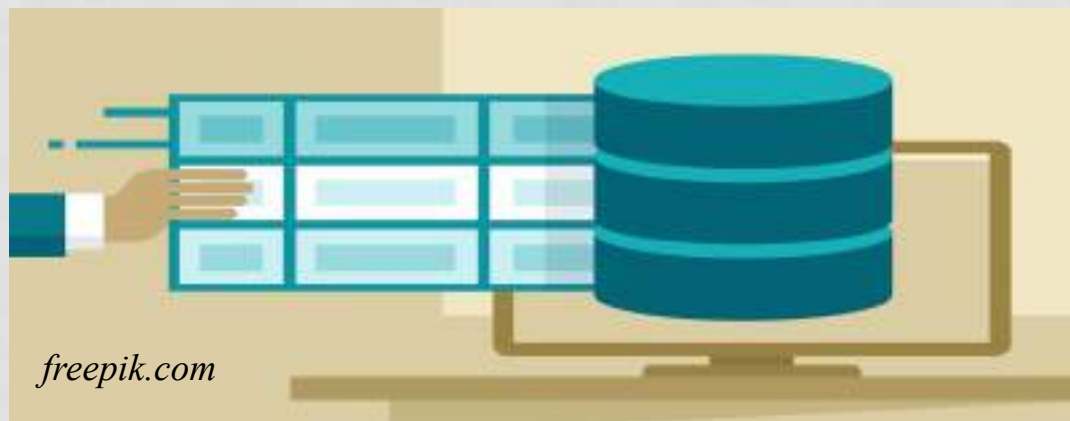
Примерами *сущностей* (с точки зрения ИС) или *объектов* (с точки зрения внешнего мира) являются отдельный студент, группа студентов, аудитория, время занятий, слова, числа, символы.

Обычно считается, что быть объектом - это значит быть дискретным и различимым.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Инфологическая модель представляет собой формализованное описание предметной области с использованием стандартных языковых средств (текстовых и графических).

Инфологическая модель строится без ориентации на какую-либо конкретную СУБД.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

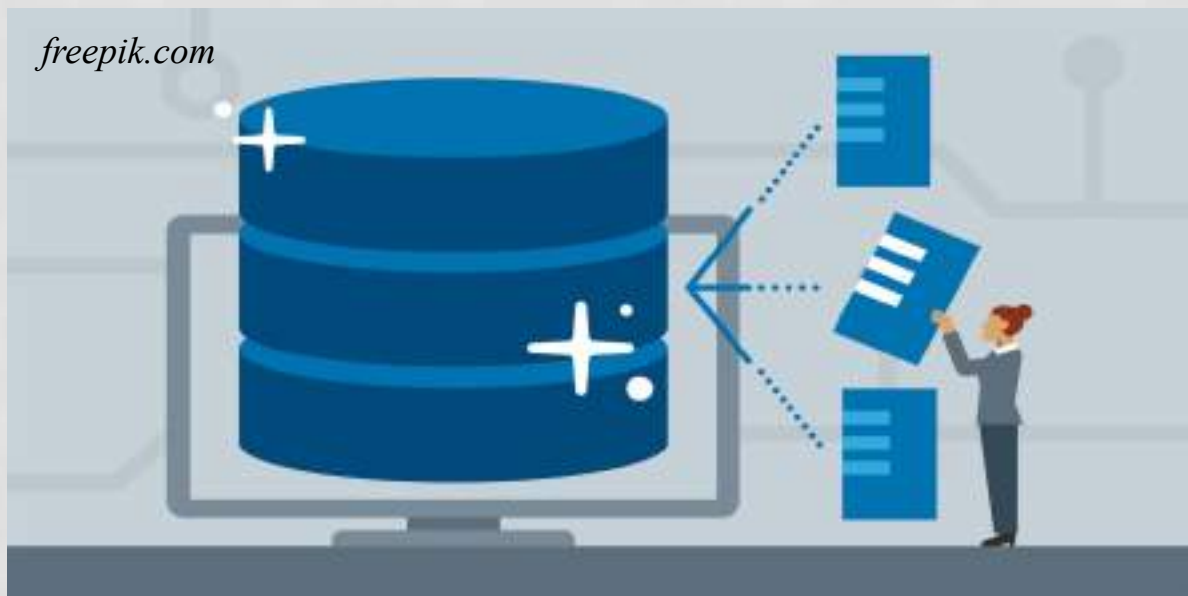
Существует несколько разновидностей инфологических моделей данных.

К ним можно отнести:

- семантическую модель данных *Хаммера и Мак-Леона*;
- функциональную модель данных *Шипмана*;
- модель «сущность-связь», предложенную *Ченом*.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

В настоящий момент модель Чена «сущность-связь», или «Entity-Relationship», стала фактическим стандартом при инфологическом моделировании баз данных.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

В основе ER-модели лежат следующие базовые понятия:

- сущность;
- атрибут;
- связь между сущностями.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Сущность — любой конкретный или абстрактный объект в рассматриваемой предметной области сведения о котором накапливаются в БД.

Сущность имеет имя уникальное в пределах проектируемой БД.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Так как сущность соответствует некоторому классу однотипных объектов, то предполагается, что в системе существует множество экземпляров данной сущности.

Поэтому следует различать понятия *тип сущности* и *экземпляр сущности*.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Понятие *тип сущности* относится к набору однородных предметов, объектов, событий, личностей, выступающих как единое целое в конкретной предметной области, имеющее независимое существование.

Экземпляр сущности относится к конкретной вещи в наборе и представляет собой однозначно идентифицируемый объект, который относится к сущности определенного типа.

Например, типом сущности может быть ГОРОД, а экземпляром – Караганда, Астана и т.д.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Каждая сущность характеризуется набором *атрибутов* (характеристики, определяющие свойства сущности). Набор атрибутов сущности должен быть таким, чтобы можно было различать конкретные экземпляры сущности.

Каждый атрибут имеет имя, уникальное в пределах сущности.

Например, у сущности Сотрудник может быть следующий набор атрибутов: Табельный номер, Фамилия, Имя, Отчество, Дата рождения, Место жительства, Количество детей.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Атрибуты подразделяются на:

- простые и составные;
- однозначные и многозначные;
- производные.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Простой (элементарный) атрибут — атрибут, состоящий из одного компонента с независимым существованием, они не могут быть разделены на более мелкие компоненты.

Пример такого атрибута: пол, цвет. . . .



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Составной атрибут – атрибут, состоящий из нескольких компонентов, каждый из которых характеризуется независимым существованием.

Например, атрибут адрес может быть разбит на более мелкие компоненты – город, улица, номер дома, номер квартиры.

Выбор формы представления атрибута (простой или сложный) определяется требованиями, которые пользователи предъявляют к приложению.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Однозначный атрибут может принимать только одно значение для каждого экземпляра сущности. Большинство атрибутов являются однозначными.

Многозначный атрибут может принимать несколько значений для каждого экземпляра сущности определенного типа.

Например, у компании или человека может быть несколько телефонных номеров, или у книги может быть несколько авторов.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Атрибуты, значения которых могут быть установлены с помощью значений других атрибутов, называются *производными*.

Например, некоторая сущность Договор имеет атрибуты: Дата начала договора, Дата окончания договора, тогда атрибут Срок действия договора может быть производным, значение которого будут вычисляться как Дата окончания договора - Дата начала договора.

Часто подобные атрибуты вообще не отражаются в концептуальной модели.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Потенциальным ключом называется атрибут или набор атрибутов сущности, позволяющий однозначно идентифицировать каждый ее экземпляр.

Для некоторых сущностей возможно наличие нескольких потенциальных ключей. В этом случае из них нужно выбрать один, который станет первичным, а другие будут называться альтернативными.

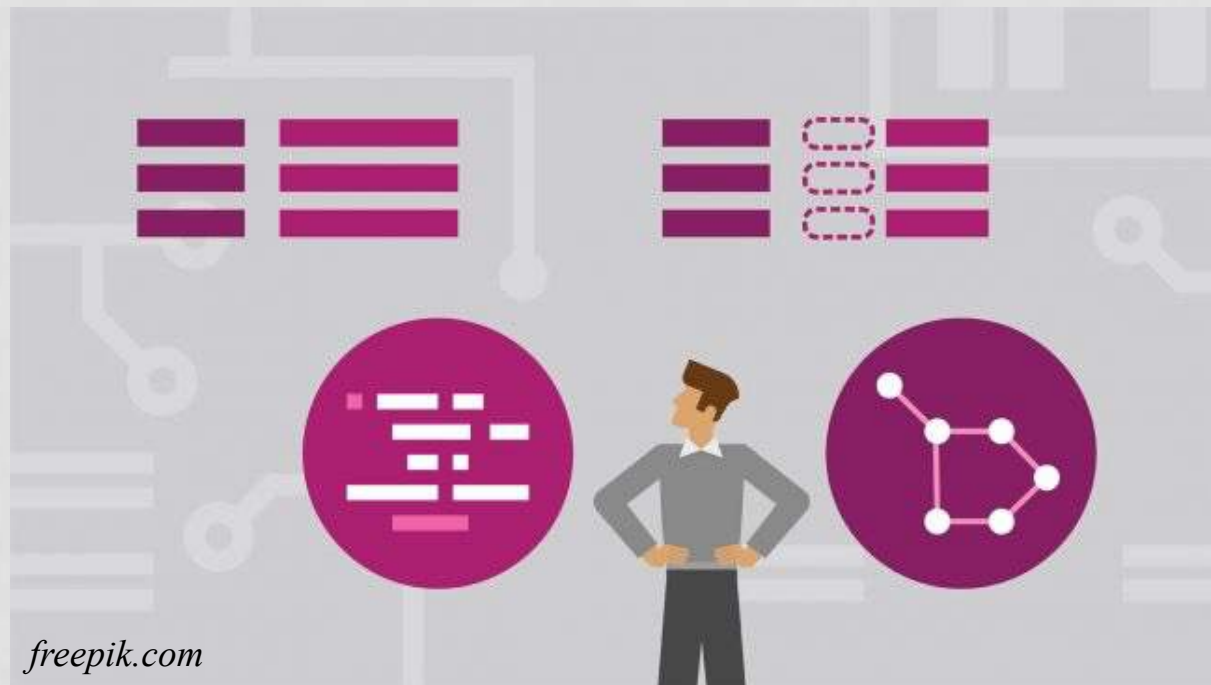
КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

При выборе первичного ключа придерживаются следующих правил:

- используйте потенциальный ключ с минимальным набором атрибутов;
- используйте тот потенциальный ключ, вероятность изменения которого минимальна;
- используйте потенциальный ключ, значения которого имеют минимальную длину, либо наименьшую максимальную длину.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Если для какой-либо сущности можно установить первичный ключ, то ее называют *сильной сущностью*, если нет, то *слабой*.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Пример определения
сущности в ER- модели

Одно из общепринятых графических обозначений сущности - прямоугольник, в верхней части которого записано имя сущности, а ниже перечисляются атрибуты, *причем ключевые атрибуты помечаются, например, подчеркиванием или специальным шрифтом*

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Между сущностями предметной области могут быть установлены *связи* - ассоциации, показывающие, каким образом сущности соотносятся или взаимодействуют между собой.

Связь может существовать между разными сущностями или между сущностью и ей же самой (*рекурсивная связь*).

Пример: поскольку каждый сотрудник работает в каком-либо отделе, между сущностями СОТРУДНИК и ОТДЕЛ существует связь "работает в" или ОТДЕЛ-РАБОТНИК;

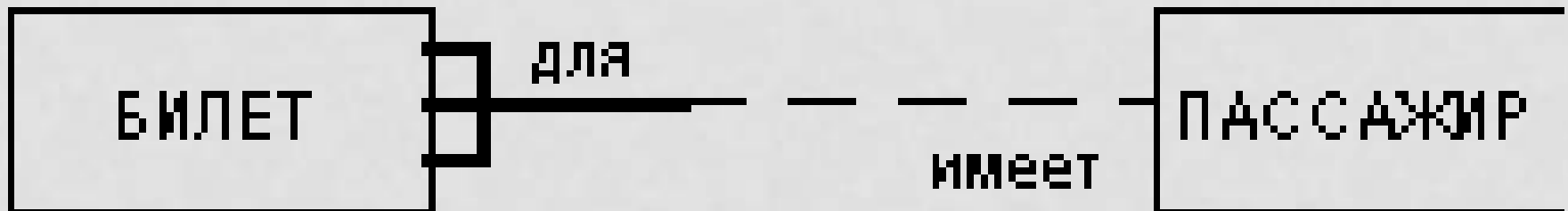
КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

В диаграммах связь обозначается в виде линии, связывающей сущности и дополненной именем связи.

На схеме может быть показано направление действия каждой связи, поскольку обычно имеет смысл только одно направление связи.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Лаконичной трактовкой изображенной диаграммы является следующая:

- каждый БИЛЕТ предназначен для одного и только одного ПАССАЖИРА;
- каждый ПАССАЖИР может иметь один или более БИЛЕТОВ.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Лаконичной трактовкой изображенной диаграммы является следующая:

- каждый ЧЕЛОВЕК является сыном одного и только одного ЧЕЛОВЕКА;
- каждый ЧЕЛОВЕК может являться отцом для одного или более ЛЮДЕЙ ("ЧЕЛОВЕКОВ").

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Степень связи — количество типов сущностей, которые охвачены данной связью.

Между сущностями предметной области могут иметь место:

- бинарные связи (между двумя сущностями или между сущностью и ей же самой - рекурсивная связь);
- тренарные связи (между тремя сущностями);
- в общем случае - n-арные связи.

На практике чаще всего встречаются связи со степенью два, то есть бинарные связи.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

При установке связей необходимо определить кратность каждой из связей и зафиксировать это.

Кратность связи — количество возможных экземпляров сущности некоторого типа, которые могут быть связаны с экземплярами сущности другого типа с помощью определенной связи.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

По множественности (кратности) связи делятся на три типа:

- *один-к-одному* (1:1);
- *один-ко-многим* (1:M);
- *многие-ко-многим* (M:M).



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Связь *один-к-одному* означает, что экземпляр одной сущности связан только с одним экземпляром другой сущности.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Связь *один ко многим* (1:M) существует, когда один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан только с одним экземпляром первой сущности

Например, преподаватель может не иметь дипломников, руководить одним или несколькими дипломными проектами.



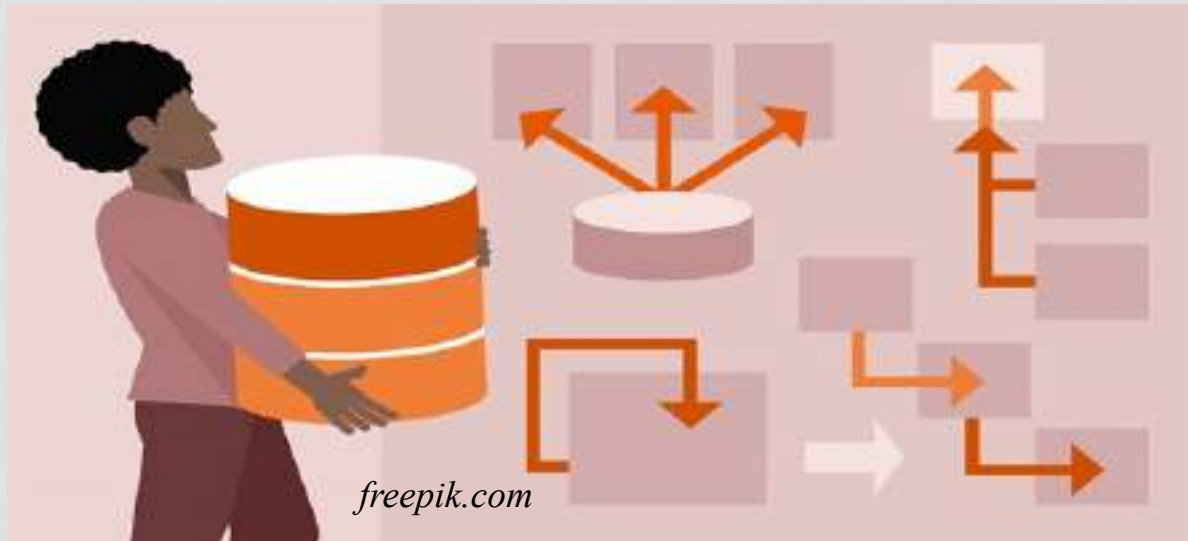
КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Пример отношения «один-ко-многим» при связывании сущностей «Студент» и «Преподаватель»

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Связь *многие ко многим* (M:N) существует, когда один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан с одним или более экземпляром первой сущности.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Пример моделирования связи «многие-ко-многим»

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Связь любого из перечисленных выше типов может быть *обязательной*, если в данной связи должен участвовать каждый экземпляр сущности, и *необязательной* - если не каждый экземпляр сущности должен участвовать в данной связи.

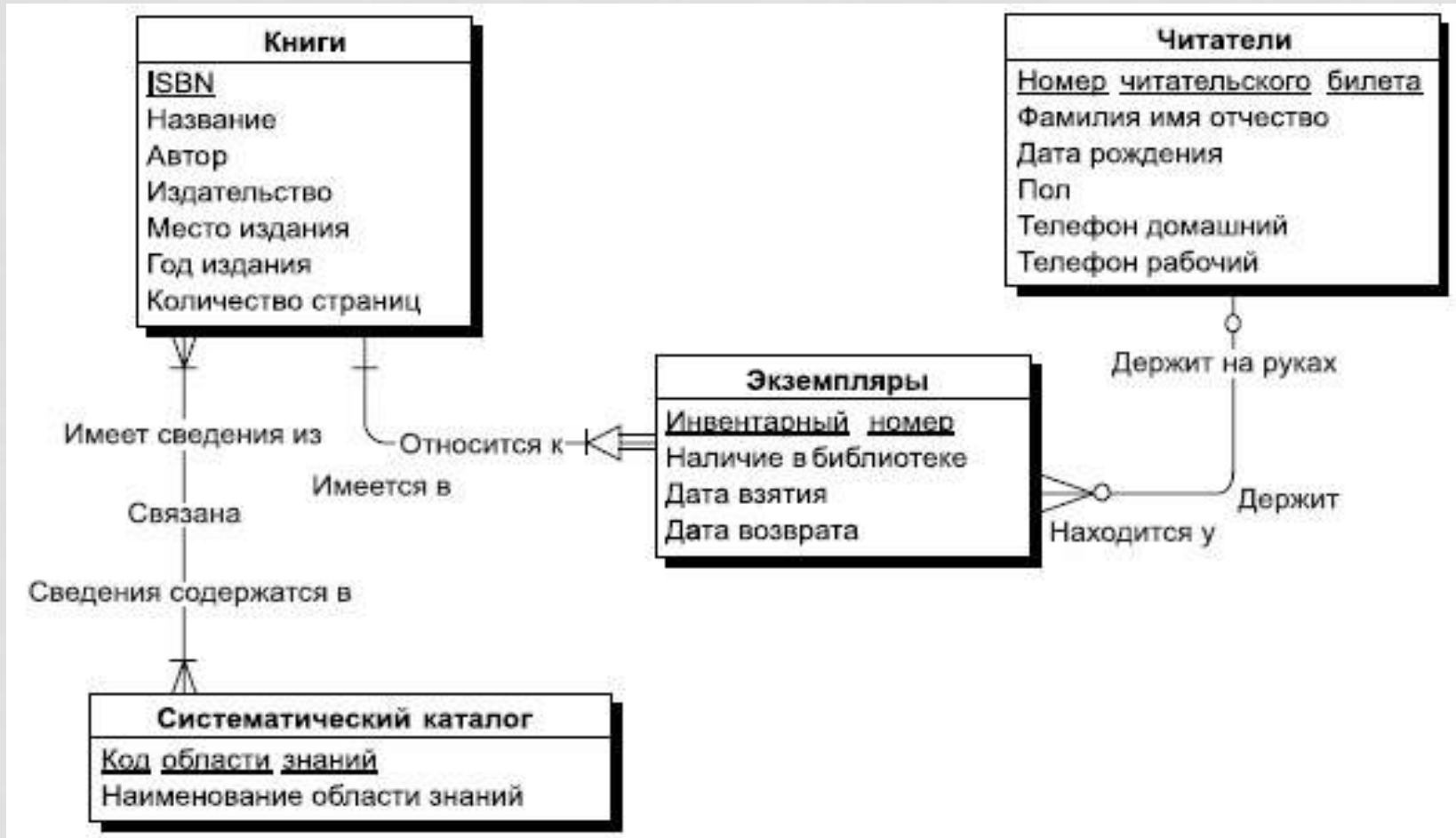
При этом связь может быть *обязательной с одной стороны* и *необязательной с другой стороны*.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Пример обязательной и необязательной связи между сущностями

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Пример инфологической модели

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

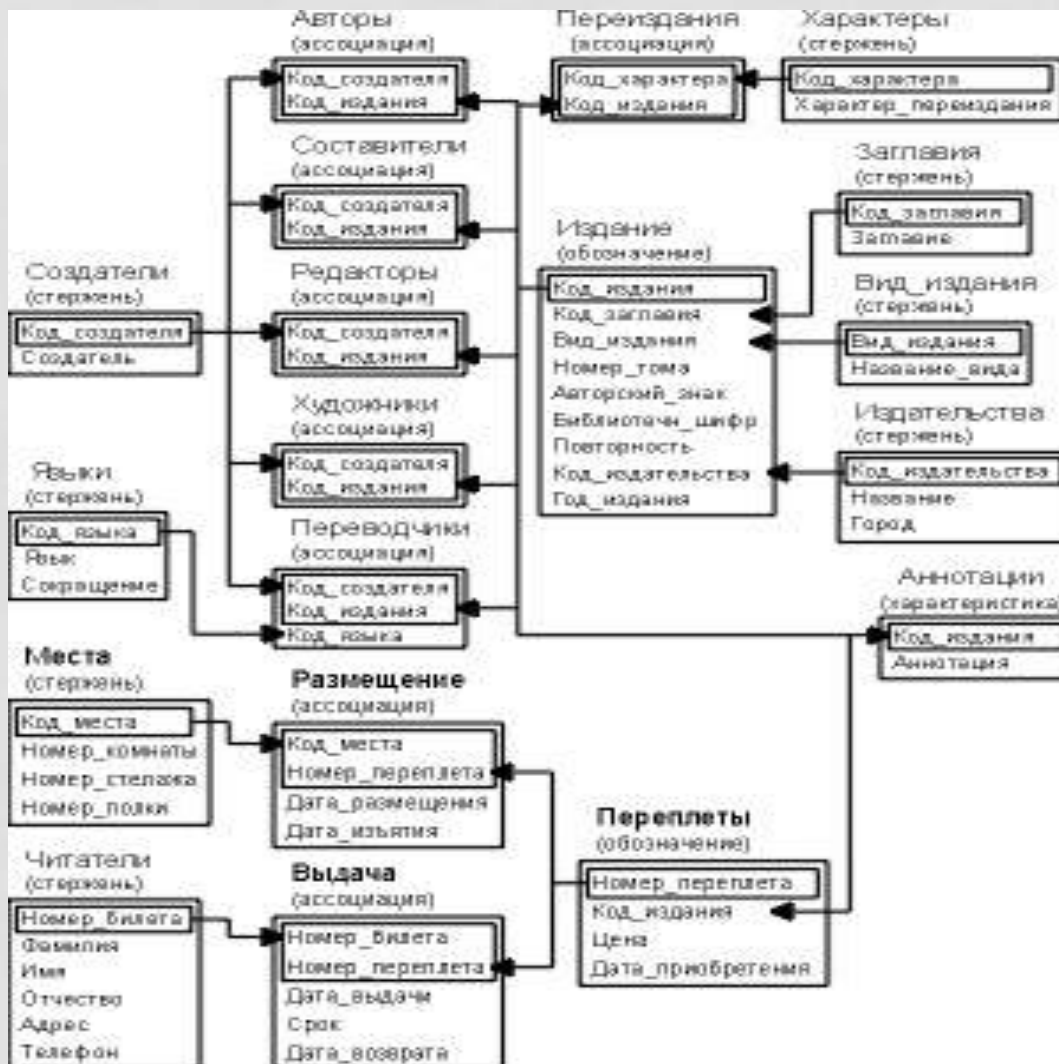
Для инфологической модели используется еще один способ описания сущностей.

Сущности изображаются одностолбцовыми таблицами с заголовками, состоящими из имени сущности.

Строки таблицы — это перечень атрибутов сущности. (Атрибут первичного ключа обозначается как (PK). За сложным атрибутом принято приводить список простых составляющих его атрибутов. Производные атрибуты отмечаются префиксом в виде косой черты (/)).

Связи между сущностями указываются стрелками, направленными от первичных ключей к внешним.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Пример описания
сущностей на языке
«таблицы-связи»

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

В ER-модели выделяют три вида сущностей:

- стержневая (стержень);
- ассоциативная (ассоциация);
- характеристическая (характеристика).

Кроме этого во множестве ассоциативных сущностей также определяют подмножество *обозначений*.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Стержневая (сильная) сущность —
независящая от других сущность.

Для описания сущности используется предложение, имеющее в общем случае вид:

СУЩНОСТЬ_ИМЯ (атрибут 1, атрибут 2 , ..., атрибут n)

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Ассоциативная сущность выражает собой связь «многие ко многим» между двумя сущностями. Является самостоятельной сущностью, может иметь не только набор ключевых атрибутов, необходимых для указания связей, но и любое число других атрибутов, характеризующих связь.

Для описания ассоциации используется предложение, имеющее в общем случае вид:

***АССОЦИАЦИЯ_ИМЯ [СУЩНОСТЬ_ S1, СУЩНОСТЬ_ S2,
...]/(атрибут 1, атрибут 2, .., атрибут n)***

где S – степень связи.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Характеристическую сущность (слабую сущность) выражает связь вида «один ко многим» и «один к одному».

Характеристическая сущность описывает или уточняет другую более сильную сущность. Она полностью зависит от нее и исчезает с исчезновением последней.

Для описания характеристики используется предложение, имеющее в общем случае вид:

***ХАРАКТЕРИСТИКА (атрибут 1, атрибут 2, ...)
{СПИСОК ХАРАКТЕРИЗУЕМЫХ
СУЩНОСТЕЙ}***

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Обозначающая сущность или *обозначение* – это связь вида "многие-к-одному" или "один-к-одному" между двумя сущностями и отличается от характеристики тем, что не зависит от обозначаемой сущности.

Для описания обозначения используется предложение, имеющее в общем случае вид:

***ОБОЗНАЧЕНИЕ (атрибут 1, атрибут 2, ...)
[СПИСОК ОБОЗНАЧАЕМЫХ СУЩНОСТЕЙ].***

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

а)

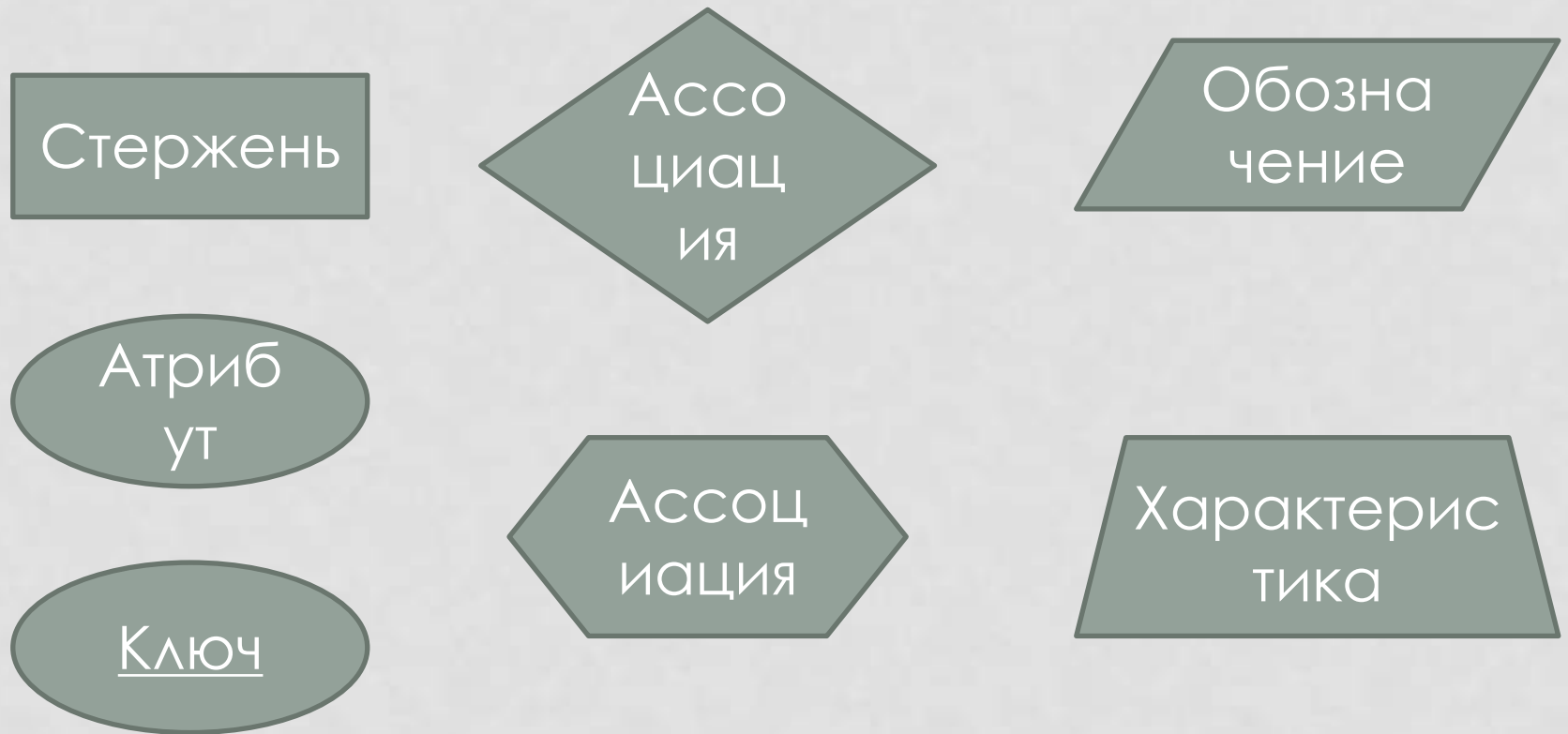


б)

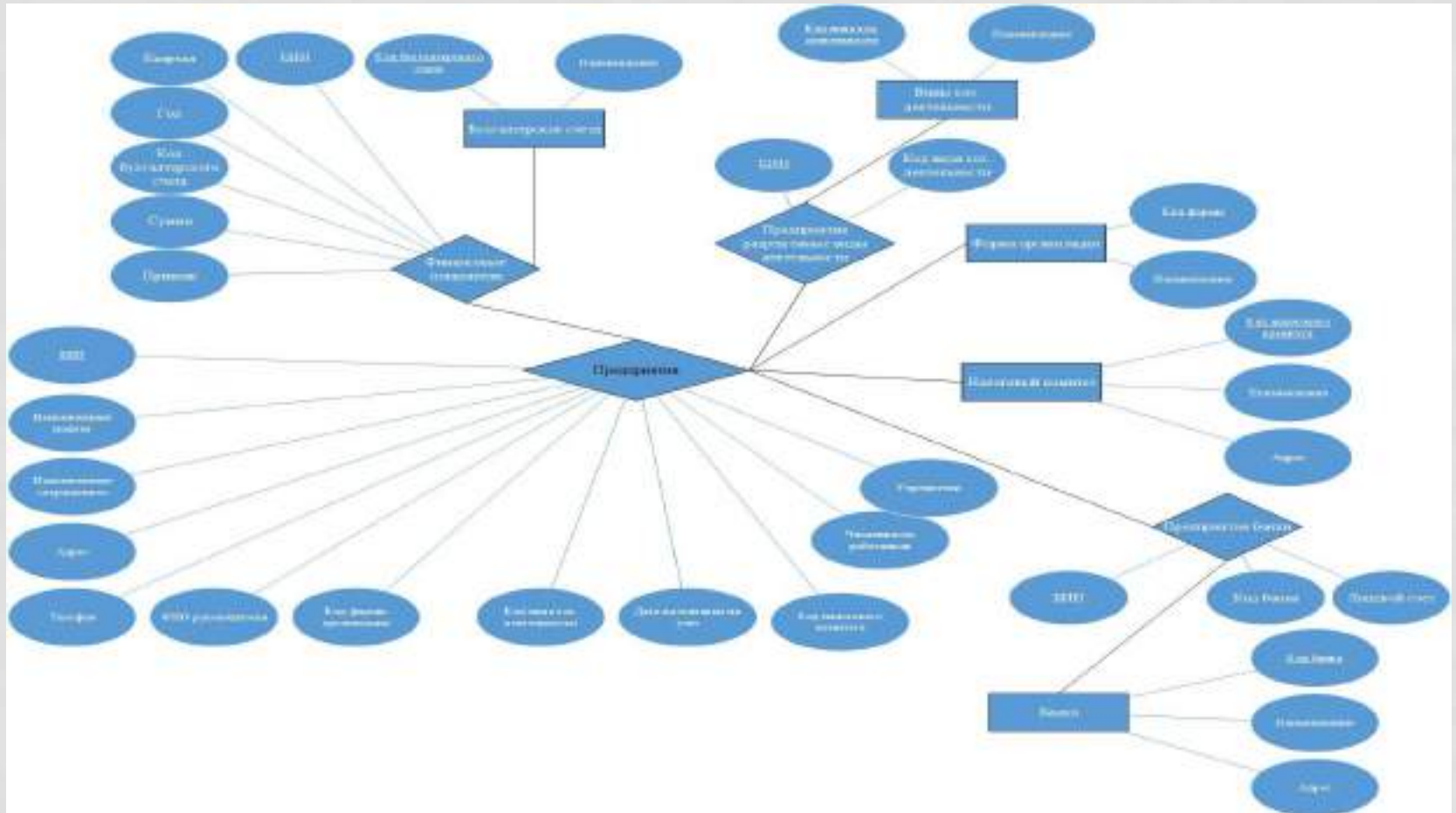


КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

В диаграммах ER-модели принято следующее представление сущностей



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА



Инфологическая модель базы данных на языке ER-диаграмм

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

В ER-модели допускается принцип *категоризации сущностей*.

Это значит, что вводится понятие подтипа сущности, то есть сущность может быть представлена в виде двух или более своих подтипов, каждая из которых может иметь общие атрибуты и отношения и/или атрибуты и отношения, которые определяются однажды на верхнем уровне и наследуются на нижнем уровне.

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

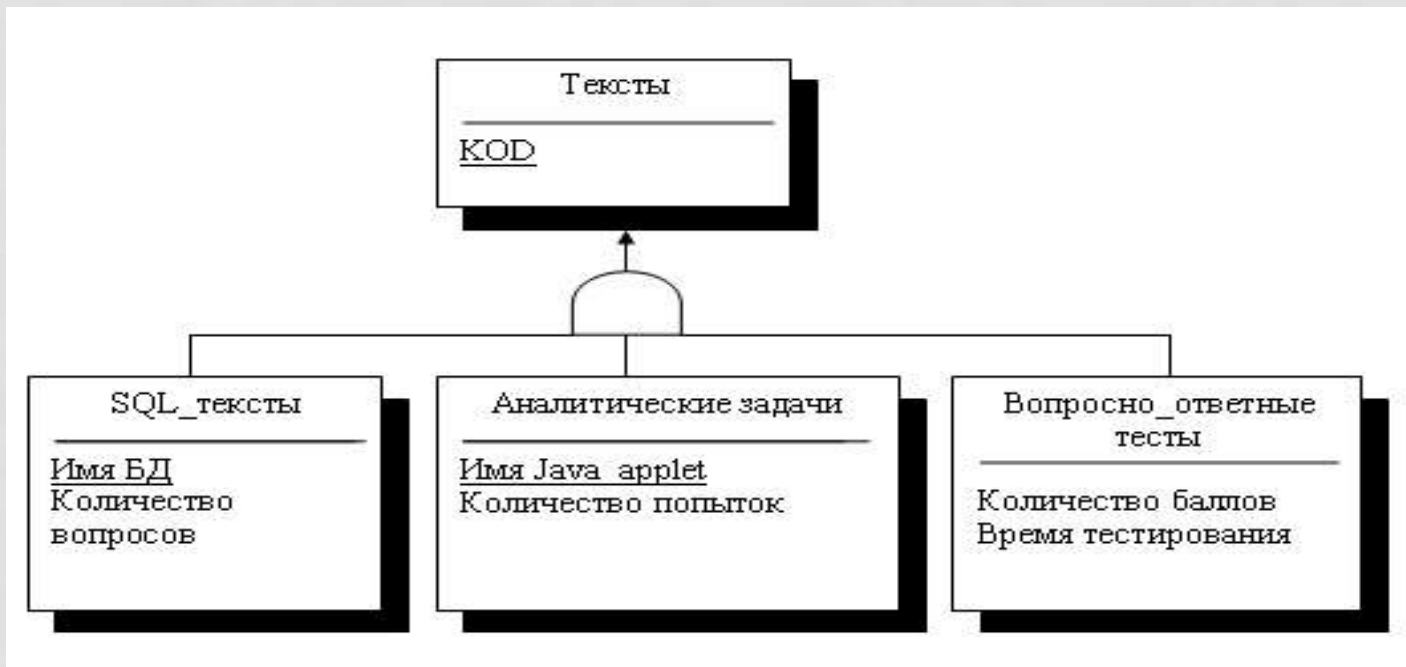
Все подтипы одной сущности рассматриваются как взаимоисключающие, и при разделении сущности на подтипы она должна быть представлена в виде полного набора взаимоисключающих подтипов.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Сущность, на основе которой строятся подтипы, называется *супертипом*.

Любой экземпляр супертипа должен относиться к конкретному подтипу.



КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ. ER-ДИАГРАММА

Приведенную выше диаграмму можно расшифровать следующим образом.

Каждый тест в некоторой системе тестирования является либо тестом проверки знаний языка SQL, либо некоторой аналитической задачей, которая выполняется с использованием заранее написанных Java-апплетов, либо тестом по некоторой области знаний, состоящим из набора вопросов и набора ответов, предлагаемых к каждому вопросу.

МЕТОДОЛОГИЯ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Физическое проектирование представляет собой процесс подготовки описания реализации базы данных во внешней памяти компьютера.



МЕТОДОЛОГИЯ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Физическое проектирование является третьим и последним этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных. Во время предыдущего этапа проектирования была определена логическая структура базы данных (которая описывает отношения и ограничения в рассматриваемой прикладной области). Хотя эта структура не зависит от конкретной целевой СУБД, она создается с учетом выбранной модели хранения данных, например реляционной, сетевой или иерархической.

МЕТОДОЛОГИЯ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Однако, приступая к физическому проектированию базы данных, прежде всего необходимо выбрать конкретную целевую СУБД. Поэтому физическое проектирование неразрывно связано с конкретной СУБД.

Между логическим и физическим проектированием существует постоянная обратная связь, так как решения, принимаемые на этапе физического проектирования с целью повышения производительности системы, способны повлиять на структуру логической модели данных.

МЕТОДОЛОГИЯ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Физический проект:

- описывает базовые отношения;
- определяет организацию файлов и состав индексов, применяемых для обеспечения эффективного доступа к данным;
- регламентирует ограничения целостности данных;
- регламентирует меры защиты.



МЕТОДОЛОГИЯ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Основная задача - преобразовать логическую модель базы данных в последовательность команд SQL для создания объектов базы данных.

Эта задача включает выполнение ряда обязательных последовательных процедур.



ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

1) Проектирование базовых отношений в среде СУБД

Для каждого отношения определяются следующие элементы:

- имя отношение;
- список простых атрибутов;
- определяется первичный ключ;
- определяются требования ссылочной целостности для любых внешних ключей.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

1) Проектирование базовых отношений в среде СУБД

Для каждого атрибута отношения должна присутствовать следующая информация:

- определение его домена, включающая указания типа данных;
- размерность внутреннего представления данных;
- требуемые ограничения на допустимые значения;
- принимаемые по умолчанию значения атрибута;
- допустимость Null значения для данного атрибута.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

2) Проектирование отношений, содержащих производные данные

Производными или расчетными называются атрибуты, значения которых можно определить с использованием значений других атрибутов.



ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

2) Проектирование отношений, содержащих производные данные

Проектировщик должен рассчитать следующее:

- дополнительные затраты на хранение производных данных и поддержание их согласованности с реальными данными, на основе которых они вычисляются;
- затраты на вычисление производных данных, если их вычисление выполняется по мере необходимости.

Решение, предусматривающее хранение производных атрибутов, является более приемлемым в том случае, если язык запросов целевой СУБД не позволяет легко реализовать алгоритм вычисления производных атрибутов.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

3) Реализация ограничений предметной области

Способ реализации ограничений зависит от возможностей выбранной СУБД:

- если СУБД поддерживает стандарт языка SQL, то ограничения реализуются этими средствами;
- альтернативным методом реализации ограничений является механизм триггеров.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

4) Проектирование физического представления БД

Одной из важнейших целей физического проектирования БД является организация эффективного хранения данных.



ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

4) Проектирование физического представления БД

Существует несколько показателей достигнутой эффективности:

- **производительность выполнения транзакций** (представляет собой количество транзакций, которые могут быть обработана за заданный интервал времени);

- **время ответа** (характеризует временной промежуток, необходимый для выполнения одной транзакции;

- **дисковая память.** Характеризует объем дискового пространства для размещения БД.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

4) Проектирование физического представления БД

Наибольшее влияние на производительность системы влияют следующие компоненты аппаратного обеспечения:

- объем оперативной памяти;
- быстродействие процессора;
- быстродействие дисковой системы;
- пропускная способность сети.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

5) Анализ транзакций

Для успешного планирования каждой транзакции необходимо знать следующее:

- транзакции, выполняемые наиболее часто и оказывающие существенное влияние на производительность;
- транзакции, наиболее важные для работы организации;
- периоды времени на протяжении суток/недель, в которые нагрузка БД возрастает до максимума.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

б) Выбор файловой структуры

В каждой СУБД по-разному организованы хранение и доступ к данным, однако существуют некоторые файловые структуры, которые имеют общепринятые способы организации и широко применяются практически во всех СУБД.



ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

б) Выбор файловой структуры

К наиболее распространенным файловым структурам используемым в СУБД относят:

- неупорядоченные (последовательные) файлы;
- упорядоченные файлы;
- хешированные файлы;
- индексно-последовательные файлы;
- сбалансированные деревья;
- кластеризованные таблицы.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

7) *Определение индексов*

Индекс – это вспомогательная структура данных:

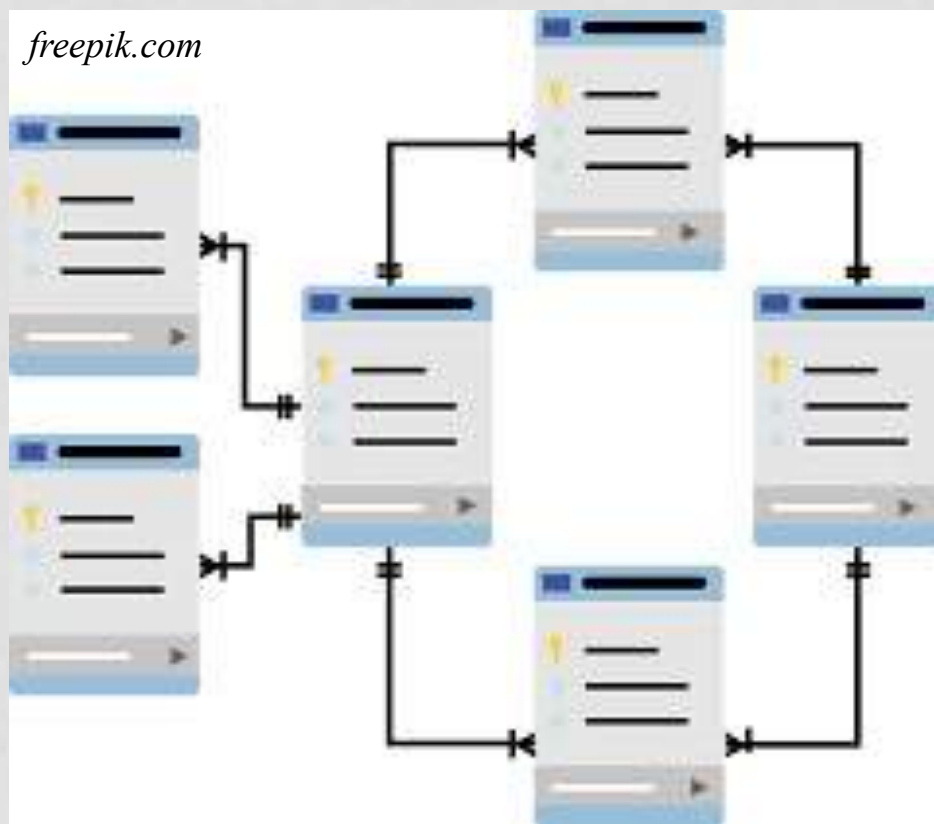
- которая помогает СУБД быстрее обнаружить отдельные записи в файле и сократить время выполнения запросов пользователей;
- связанная с файлом, позволяет избежать проведения последовательного или пошагового просмотра файла в поисках нужных данных.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

7) Определение индексов

Типы индексов:

- первичный;
- уникальный;
- внешний.



ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

8) Разработка механизмов защиты

Как правило, реляционные СУБД предоставляют средства защиты БД следующих типов:

- защита системы;
- защита данных.



ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

8) Разработка механизмов защиты

Средства защиты системы регламентируют доступ и эксплуатацию БД на уровне системы; к ним, в частности, относится аутентификация пользователя по имени и паролю.

Средства защиты данных регламентируют доступ и использование объектов БД (таких как отношение и представление), а также действия, которые могут быть выполнены пользователями с конкретными объектами.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

8) Разработка механизмов защиты

GRANT ALL PRIVILEGES ON Сотрудник
TO Manager
WITH GRANT OPTION;

REVOKE UPDATE, DELETE ON Сотрудник
TO Manager;

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

9) Анализ необходимости введения контролируемой избыточности

Денормализация иногда применяется как средство повышения производительности системы.

Если производительность системы не удовлетворяет выдвигаемым требованиям и проектируемое отношение имеет низкую скорость обновления при большой частоте запросов, денормализация реляционной модели может быть оправданной.

ЭТАПЫ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ

10) Организация мониторинга и настройка функционирования ОС

Цель этого этапа - обеспечить текущий контроль функционирования операционной системы и добиться повышения ее производительности для устранения последствий принятия неоптимальных проектных решений или учета изменившихся требований.

АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

В самом широком смысле *информационная система* представляет собой программный комплекс, функции которого состоят в поддержке надежного хранения информации в памяти компьютера, выполнении специфических для данного приложения преобразований информации и/или вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса.

АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Обычно объемы информации, с которыми приходится иметь дело таким системам, достаточно велики, а сама информация имеет достаточно сложную структуру.



АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Архитектура ИС является логическим построением, или моделью, и влияет на совокупную стоимость владения через набор связанных с ней решений по выбору средств реализации, СУБД, операционной платформы, телекоммуникационных средств и т.п., т.е. через то, что мы принято называть инфраструктурой ИС.



АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Информационные системы на основе БД *по способу организации хранения и доступа к данным* классифицируются на:

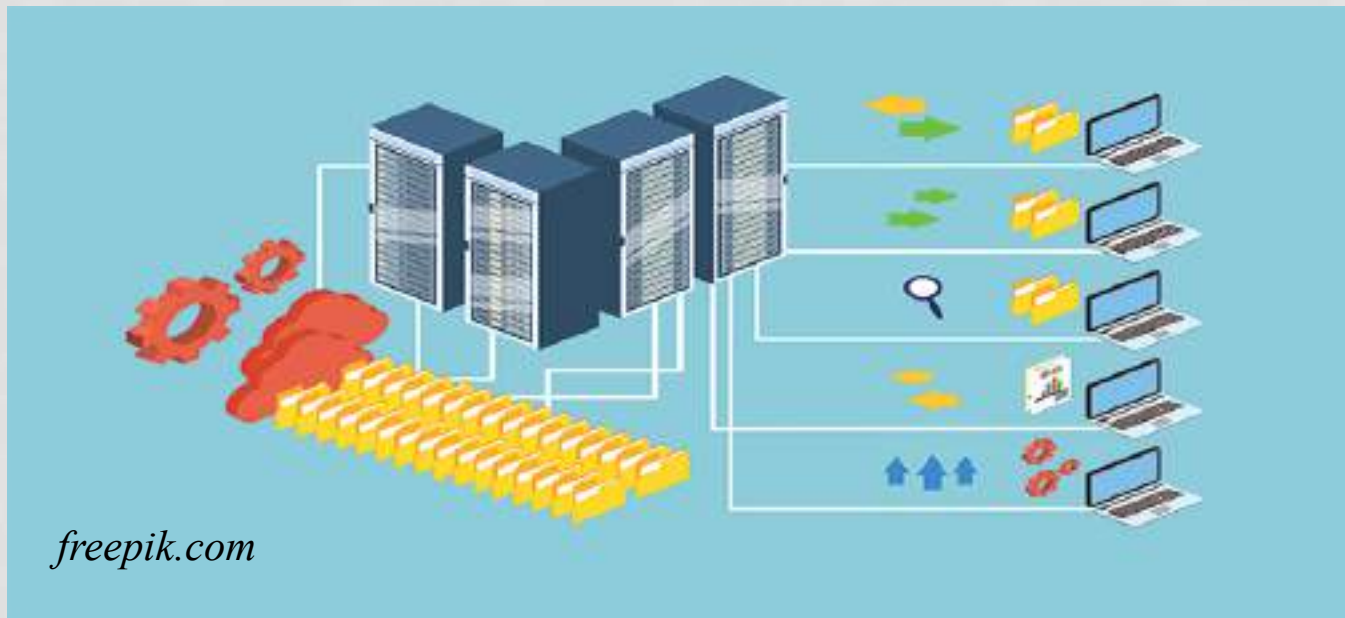
- локальные (централизованные);
- распределенные.



АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

К *централизованным* информационным системам принято относить:

- персональные;
- частично групповые.



АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Локальные ИС реализуются на автономном компьютере, как правило, ПК.

Такая система может содержать несколько простых приложений, связанных общим информационным фондом (БД), и рассчитана на работу одного пользователя или группы пользователей, разделяющих по времени одно рабочее место.

И база данных, и приложения, работающее с БД находятся на одном компьютере.

АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Подобные приложения создаются с помощью так называемых "настольных СУБД" (например, MS Access) или с помощью файловой системы и диалоговой оболочки для ввода, редактирования и обработки данных.



АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

При выборе настольных СУБД необходимо обратить внимание на следующие аспекты:

- наличие базового инструментария для построения таблиц, работы с данными и структурами в диалоговом режиме, сортировки и импорта записей;
- наличие форм как инструментального средства для формирования структур элементов данных и экранов просмотра;
- наличие средств конструирования простых и сложных запросов и отчетов;

АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

При выборе настольных СУБД необходимо обратить внимание на следующие аспекты:

- наличие средств быстрой разработки программ и внешних интерфейсов;
- наличие средств связности характеризует связь с данными, хранящимися в файлах различных форматов;
- наличие средства для публикации статических и динамических документов в Internet, а также для ввода и извлечения данных из сети.

АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Групповые информационные системы ориентированы на коллективное использование информации членами рабочей группы (одного подразделения), чаще всего строятся как локальная вычислительная сеть ПК (распределенный вариант построения) или реже как *многотерминальная централизованная* вычислительная система.

И база данных, и приложения, работающее с БД находятся на одном компьютере — центральном, главном.

АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Однотипные или специализированные рабочие места обеспечивают запуск на центральном компьютере и вызов одного или нескольких приложений.

При разработке таких приложений используются многопользовательские "настольные СУБД", серверы БД для рабочих групп (MS SQL Server, Workgroup Oracle и др.).



АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Однотипные или специализированные рабочие приложения, разработанные для таких систем, часто имеют монолитную архитектуру, когда интерфейс пользователя, бизнес логика и функциональность доступа к данным располагаются в одном большом приложении.



АРХИТЕКТУРЫ ИС НА ОСНОВЕ БД

Распределенные информационные системы принято строить различными способами, как системы:

- с архитектурой клиент-сервер;
- с распределенными вычислениями;
- на основе Internet/Intranet-технологий.



ФАЙЛ-СЕРВЕРНАЯ АРХИТЕКТУРА



ФАЙЛ-СЕРВЕРНАЯ АРХИТЕКТУРА

Пример 1. Предположим, что в базе данных на сервере хранится список сотрудников крупного предприятия. На предприятии 1500 сотрудников и 10 подразделений. Пользователю нужно получить число сотрудников, работающих в каждом подразделении.

Для решения этой задачи пользователь должен запросить данные всех 1500 сотрудников с сервера по сети, после чего на пользовательском компьютере выполнится процедура, которая осуществит подсчет сотрудников в каждом подразделении. Результатом процедуры будет 10 строк. Таким образом, чтобы получить 10 строк придется передать по сети 1500 строк.

КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА



КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА

Пример 2. Рассмотрим задачу из примера 1 в условиях клиент-серверной архитектуры.

Пользователь отправит на сервер запрос, который запустит процедуру. Процедура выполнится непосредственно на сервере. Она подсчитает количество сотрудников в каждом подразделении и отправит полученные 10 строк по сети на клиентский компьютер. Таким образом, произойдет существенная экономия трафика: вместо 1500 строк будет передано по сети всего 10.

ТРЕХУРОВНЕВАЯ АРХИТЕКТУРА



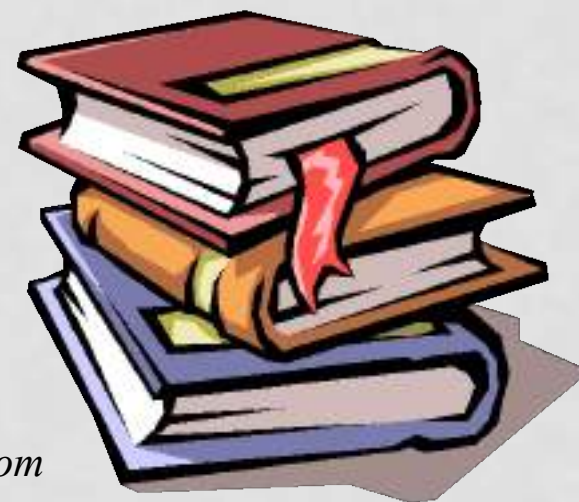
ЗАДАНИЯ ДЛЯ СРС

1. Перечислите этапы физического проектирования БД.
2. Что понимается под архитектурой ИС?
3. Охарактеризуйте локальные (централизованные) и распределенные решения. Выделите их достоинства и недостатки.
4. Охарактеризуйте архитектуру «файл-сервер».
5. Охарактеризуйте двухуровневую архитектуру «клиент-сервер».
6. Охарактеризуйте трехуровневую архитектуру «клиент-сервер».

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.



***ЛЕКЦИЯ ОКОНЧЕНА.
БЛАГОДАРЮ ЗА ВНИМАНИЕ!***