Логическое проектирование БД

Структура курса

Курс **Базы данных** для профилей **"Проектирование и разработка информационных систем",**

"Разработка алгоритмов и программного обеспечения" и

"Математическое обеспечение и администрирование информационных систем" читает **Сосинская Софья Соломоновна** в объеме 2 час /неделю (34 часа)

Лабораторные работы в таком же объеме ведут:

- Для профилей "Проектирование и разработка информационных систем", "Разработка алгоритмов и программного обеспечения" **Харахинов** Владимир Александрович
- Для профиля "Математическое обеспечение и администрирование информационных систем" **Сосинская Софья Соломоновна**

Лабораторные работы в количестве 4 штуки выполняются по определенным темам, имеют определенный бюджет времени, оформляются в виде отчета и защищаются у преподавателя

Номер лабораторной работы	Тема лабораторной работы	Бюджет времени (число недель)
1	Проектирование логической структуры БД	3
2	Разработка серверной части клиент-серверного приложения	4
3	Разработка клиентского приложения для локальной сети	6
4	Разработка клиентского приложения для сети Интернет	4

Для выполнения всех лабораторных работ каждому студенту выдается свой вариант словесного описания предметной области.

Курс заканчивается **экзаменом**, для допуска к которому необходимо выполнить и защитить все лабораторные работы и выполнить **контрольную работу «Использование языка SQL» .**

Имеется возможность получить экзамен автоматом при условии выполнения всех лабораторных в срок

База данных. Система управления базами данных

Слово «**данные»** определяется как составная часть информации.

Изначально при создании баз данных применялись следующие типы данных:

1) числовые (например, 17; 0,27; 2E-7);

2) символьные или алфавитно-цифровые (в частности, «потолок», «стол»);

3) даты, которые задаются с помощью специального типа «Дата» или как обычные символьные данные (например, 12.02.2005, 12/02/2005).

Позднее были определены другие типы данных, в том числе:

1) временные и дата-временные, которые применяются для хранения информации о времени и/или дате (например, 5.02.2005, 7:27:04, 23.02.2005 16:00);

2) символьные данные переменной длины, предназначенные для хранения текстовой информации большой длины;

3) двоичные, которые используются для хранения графических объектов, аудио– и видеоинформации, пространственной, хронологической и другой специальной информации;

4) гиперссылки, позволяющие хранить ссылки на различных ресурсах, располагающихся вне базы данных.

База данных – это совокупность определенным образом взаимосвязанных данных, хранящихся в памяти ЭВМ для отображения структуры объектов и их связей в изучаемой предметной области. Она является основной формой организации хранения данных в информационных системах.

Система управления базами данных представляет собой комплекс символьных и программных средств, предназначенных для создания, ведения и организации совместного доступа к базам данных множества пользователей.

Первые СУБД были разработаны фирмами IBM – IMS (1968 г.) и Software AG– ADABA• (1969 г.). В настоящий момент существует большое число различных систем управления базами данных (более нескольких тысяч), и их количество постоянно растет.

Среди основных функций СУБД (функций высшего уровня) можно выделить

- хранение,
- изменение и обработку информации, а также
- разработку и получение различных выходных документов.

К функциям СУБД более низкого уровня относятся:

1) управление данными во внешней памяти;

2) управление буферами ОП;

3) управление транзакциями;

4) ведение журнала изменений в базе данных;

5) обеспечение целостности и безопасности баз данных.

Иерархическая, сетевая и реляционная модели представления данных

Информация в базе данных некоторым образом структурирована, т. е. ее можно описать моделью представления данных (моделью данных), которая поддерживается СУБД.

Эти модели подразделяют на иерархические, сетевые и реляционные.

При использовании **иерархической модели** представления данных связи между данными можно охарактеризовать с помощью упорядоченного графа (или дерева).

Основными достоинствами иерархической модели данных являются:

1) эффективное использование памяти ЭВМ;

2) высокая скорость выполнения основных операций над данными;

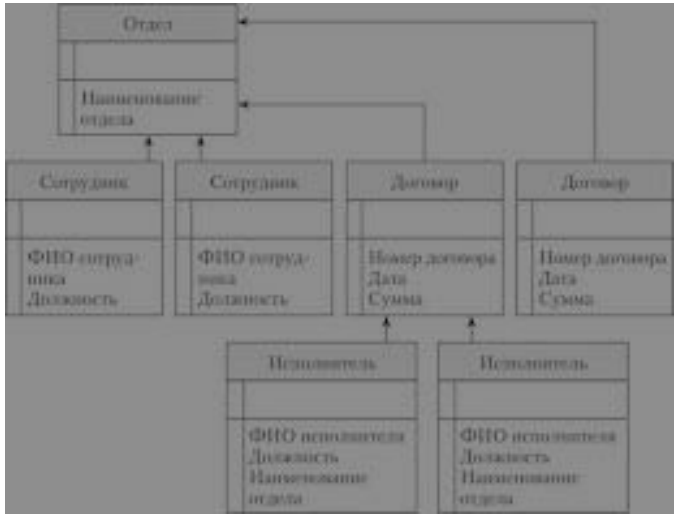
3) удобство работы с иерархически упорядоченной информацией.

К недостаткам иерархической модели представления данных относятся:

1) громоздкость такой модели для обработки информации с достаточно сложными логическими связями;

2) трудность в понимании ее функционирования обычным пользователем.

На сегодняшний день существует лишь незначительное число


Отдел
Наименование отдела
Сотрудник
ФИО сотрудника
Должность
Сотрудник
ФИО сотрудника
Должность
Договор
Номер договора
Дата
Сумма
Договор
Номер договора
Дата
Сумма
Исполнитель
ФИО исполнителя
Должность
Наименование отдела
Исполнитель
ФИО исполнителя
Должность
Наименование отдела

Сетевая модель может быть представлена как развитие и обобщение иерархической модели данных, позволяющее отображать разнообразные взаимосвязи данных в виде произвольного графа.

Достоинствами сетевой модели представления данных являются:

1) эффективность в использовании памяти компьютера;

2) высокая скорость выполнения основных операций над данными;

3) большие, чем у иерархической модели возможности образования произвольных связей.

К недостаткам сетевой модели представления данных относятся:

1) высокая сложность и жесткость схемы базы данных, которая построена на ее основе;

2) трудность для понимания и выполнения обработки информации в базе данных непрофессиональным пользователем.

Студент(№зачетной книжки, фамилия, группа)

Работа(шифр, руководитель, область)

Реляционная модель представления данных была разработана сотрудником фирмы 1ВМЭ. Коддом. Его модель основывается на понятии «отношения» (relation). Простейшим примером отношения служит двумерная таблица.

Достоинствами реляционной модели представления данных (по сравнению с иерархической и сетевой моделями) являются ее понятность, простота и удобство практической реализации реляционных баз данных на ЭВМ.

К недостаткам реляционной модели представления данных относятся:

1) отсутствие стандартных средств идентификации отдельных записей;

2) сложность описания иерархических и сетевых связей.

Большинство СУБД, применяемых как профессиональными, так и непрофессиональными пользователями, построены на основе реляционной модели данных (Access, SQL Server фирмы Microsoft, Oracle фирмы Oracle и др.).

Основные понятия реляционной модели данных

Наиболее распространенная трактовка реляционной модели данных, по-видимому, принадлежит Дейту, который утверждает, что реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода: структурной части, манипуляционной части и целостной части.

В **структурной части** модели фиксируется, что единственной структурой данных, используемой в реляционных БД, является нормализованное n-арное отношение.

В **манипуляционной части** модели утверждаются два фундаментальных механизма манипулирования реляционными БД - реляционная алгебра и реляционное исчисление.

Первый механизм базируется в основном на классической теории множеств (с некоторыми уточнениями), а второй - на классическом логическом аппарате исчисления предикатов первого порядка. Основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка реляционных БД: язык называется реляционным, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

Целостная часть модели определяет требования целостности сущностей и целостности ссылок. Первое требование состоит в том, что любой кортеж любого отношения отличим от любого другого кортежа этого отношения, то есть. другими словами, любое отношение должно обладать **первичным ключом**.

Требование целостности по ссылкам, или требование **внешнего ключа** состоит в том, что для каждого значения внешнего ключа, появляющегося в ссылающемся отношении, в отношении, на которое ведет ссылка, должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным (то есть ни на что не указывать).

Основными понятиями реляционных баз данных являются

тип данных,

домен,

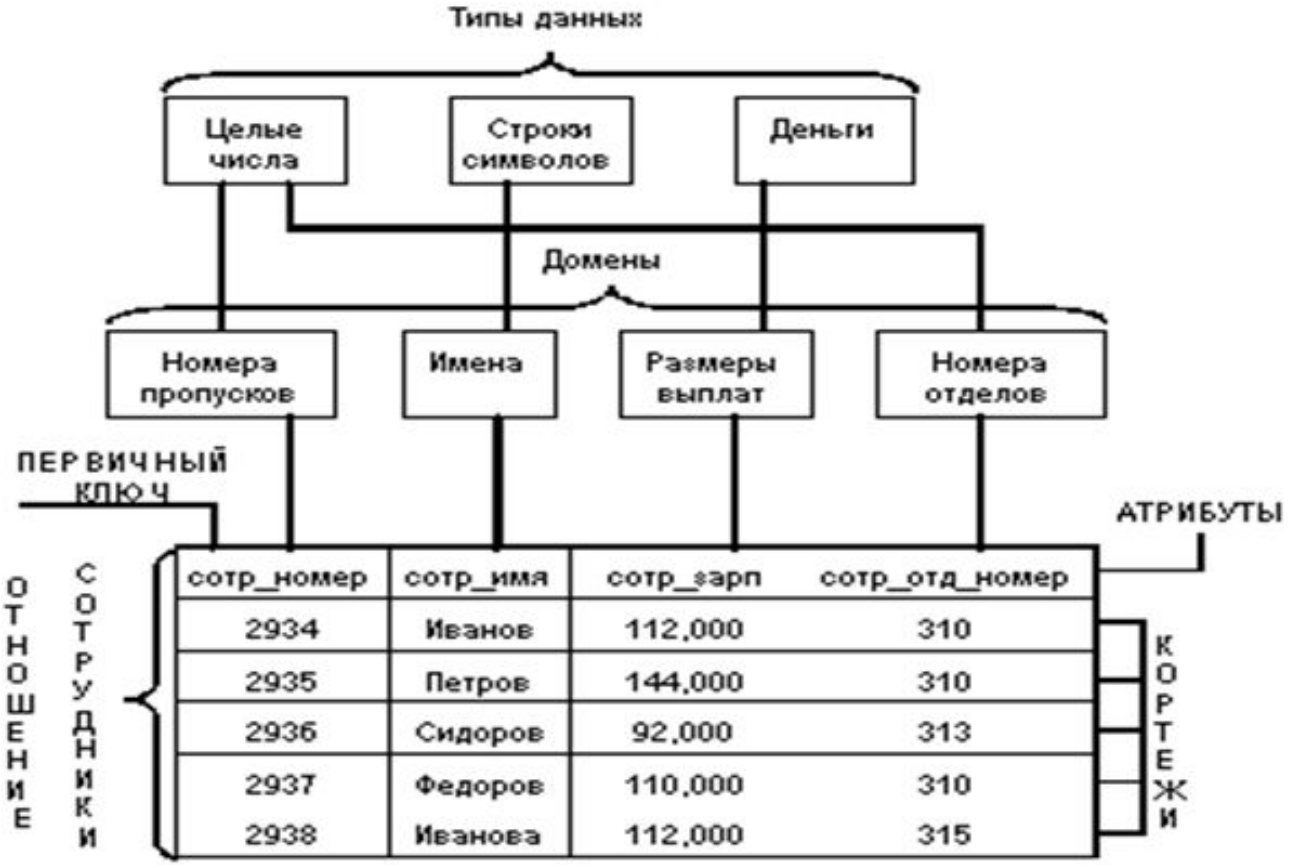
атрибут,

кортеж,

первичный ключ и

отношение.

Покажем смысл этих понятий на примере отношения СОТРУДНИКИ, содержащего информацию о сотрудниках некоторой организации


Типы данных
Целые числа
Строки символов
Деньги
Домены
Номера пропусков
Имена
Размеры выплат
Номера отделов
ПЕРВИЧНЫЙ КЛЮЧ
АТРИБУТЫ
ОТНОШЕНИЕ СОТРУДНИКИ
КОРТЕЖИ
сотр_номер
сотр_имя
сотр_зарп
сотр_отд_номер
2934
Иванов
112,000
310
2935
Петров
144,000
310
2936
Сидоров
92,000
313
2937
Федоров
110,000
310
2938
Иванова
112,000
315

Понятие **Тип данны**х в реляционной модели данных идентично понятию Тип данных в языках программирования. Обычно в современных реляционных БД допускается хранение символьных, числовых данных, а также данных типа дата и время.

Наиболее правильной трактовкой понятия **домена** является понимание его как допустимого потенциального множества значений данного типа. Например, домен «Имена» в нашем примере определен на базовом типе строк символов, но в число его значений могут входить только те строки, которые могут изображать имя (в частности, такие строки не могут начинаться с мягкого знака).

Следует отметить также семантическую нагрузку понятия домена: данные считаются сравнимыми только в том случае, когда они относятся к одному домену. В нашем примере значения доменов «Номера пропусков» и «Номера отделов» относятся к типу целых чисел, но не являются сравнимыми.

Атрибут - именованный столбец, принимающий значения из некоторого домена.

Схема отношения – это именованное множество пар {имя атрибута, имя домена}.

Степень, или «арность» схемы отношения – мощность этого множества. Степень отношения СОТРУДНИКИ равна четырем, то есть оно является 4-арным.

Схема БД (в структурном смысле) – это набор именованных схем отношений.

Кортеж – это набор именованных значений заданного типа.

Первичный ключ отношения – один или более атрибутов, однозначно идентифицирующих каждый кортеж.

Отношение – это множество кортежей, соответствующих одной схеме отношения. Схему отношения называют заголовком отношения, а отношение как набор кортежей – телом отношения.

Обычным представлением отношения является таблица, заголовком которой является схема отношения, а строками – кортежи отношения-экземпляра; в этом случае имена атрибутов именуют столбцы этой таблицы. Поэтому иногда говорят «столбец таблицы», имея в виду «атрибут отношения».

Реляционная база данных – это набор отношений, имена которых совпадают с именами схем отношений в схеме БД.

Метод «Объект-Связь» для логического проектирования БД

Этот метод был предложен Питером Ченом в 1976 году.

Рассматриваются объекты, имеющие экземпляры и обладающие набором свойств - атрибутов (например, КУРС имеет атрибуты название_курса, номер_семестра, шифр_группы, число_часов и т. д.).

Объекты допускают однозначную идентификацию (например, объекты КУРС и ПРЕПОДАВАТЕЛЬ, причем первый объект идентифицируется названием курса, а второй - ФИО преподавателя).

Атрибут (или набор атрибутов), однозначно идентифицирующий объект, называется **ключом объекта**.

Между объектами существуют **связи** (например, ПРЕПОДАВАТЕЛЬ ЧИТАЕТ КУРС). В дальнейшем будут рассматриваться только бинарные связи (связи между двумя объектами).

Степень связи

Каждая связь характеризуется степенью связи, которая определяет, сколько экземпляров одного объекта связано с экземплярами другого объекта. В соответствии с этим определением существуют бинарные связи степеней:

- 1:1 (каждый экземпляр одного объекта связан не более чем с одним экземпляром другого)
- 1:N (каждый экземпляр одного объекта связан с несколькими экземплярами другого, но каждый экземпляр второго объекта связан не более чем с одним экземпляром первого),
- M:N (каждый экземпляр первого объекта связан с несколькими экземплярами второго и наоборот),

.

Класс принадлежности

При любой степени связи **класс принадлежности** каждого объекта может быть обязательным или необязательным.

Класс принадлежности объекта является **обязательным**, если все экземпляры объекта участвуют в связи, и

необязательным - в противном случае.

Будем записывать возможные варианты классов принадлежности в виде

O-O, O-H, H-O, H-H.

Указание * вместо O или H означает, что класс принадлежности несущественен.

Сочетание степени связи и класса принадлежности может быть любым и зависит от условий функционирования предметной области.

Например, связь ПРЕПОДАВАТЕЛЬ ЧИТАЕТ КУРС может иметь **степень 1:1** (это означает, что каждый преподаватель читает не более одного курса, и каждый курс читается не более чем одним преподавателем),

степень 1:N (это означает, что каждый преподаватель может читать несколько курсов, но каждый курс читается не более чем одним преподавателем) или

степень M:N (это означает, что каждый преподаватель может читать несколько курсов, и каждый курс может читаться несколькими преподавателями).

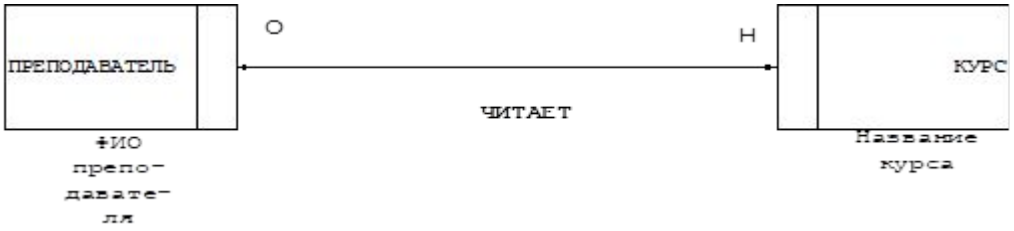
При каждом варианте степени связи могут быть различные варианты класса принадлежности для каждого из двух объектов, участвующих в связи.

Например, если степень связи 1:1 и класс принадлежности для первого объекта обязателен, а для второго необязателен, то это означает, что все экземпляры объекта ПРЕПОДАВАТЕЛЬ читают какие-либо курсы, но есть курсы, которые не читаются ни одним преподавателем.

Графически такой вариант для экземпляров объектов отображается следующим образом:

4

Чаще дается графическое отображение на уровне объектов



Вид стрелки обозначает степень связи; О и Н - класс принадлежности обязателен или необязателен; линия - связь.

Такое отображение называется ER - диаграммой.

Аналогичные отображения могут быть даны для всех комбинаций вариантов степеней связи и классов принадлежности.

Правила для построения отношения

Проектирование логической модели состоит в построении реляционных отношений и их первичных ключей на основе анализа ER - диаграмм и применения следующих правил, учитывающих степень связи и класс принадлежности объектов:

ПРАВИЛО 1. Если степень связи 1:1 и классы принадлежности О-О, то требуется только одно отношение, первичным ключом которого может быть ключ любого объекта.

ПРАВИЛО 2. Если степень связи 1:1 и классы принадлежности О-Н, то необходимо построение двух отношений. Каждому объекту соответствует одно отношение; при этом ключ объекта является первичным ключом соответствующего отношения. Кроме того, ключ второго объекта добавляется в качестве атрибута в первое отношение.

ПРАВИЛО 3. Если степень связи 1:1 и классы принадлежности Н-Н, то необходимо построение трех отношений: по одному на каждый объект (первичные ключи отношений совпадают с ключами объектов) и связующего отношения, первичным ключом которого будет комбинация ключей объектов.

ПРАВИЛО 4. Если степень связи 1:N и классы принадлежности * - О, то достаточным является использование двух отношений, соответствующих объектам, причем первичными ключами будут ключи объектов, но дополнительно ключ первого объекта должен быть атрибутом второго отношения.

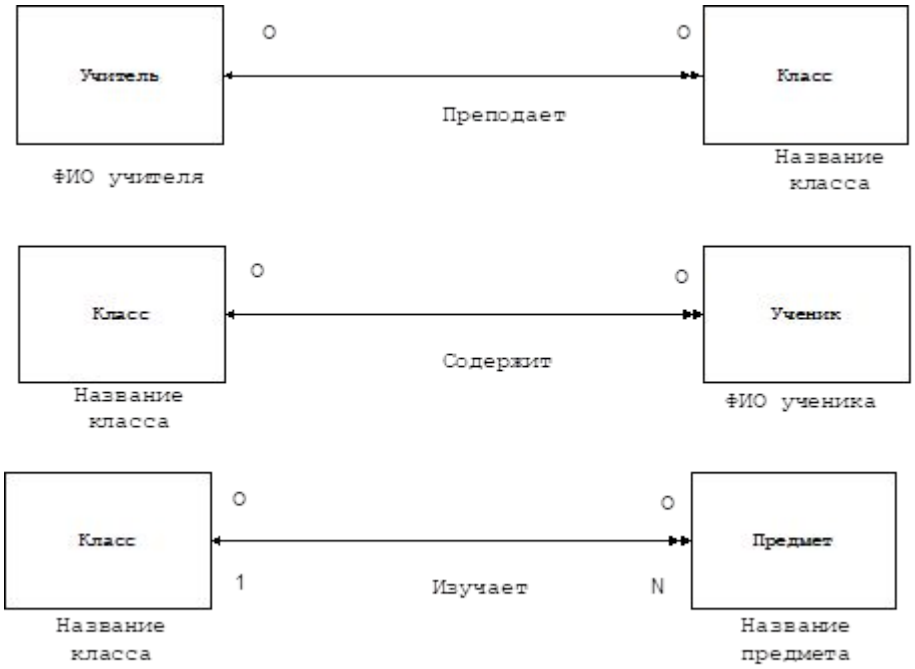
ПРАВИЛО 5. Если степень связи 1:N и классы принадлежности *-Н, то необходимо построение трех отношений: по одному на каждый объект (первичные ключи отношений совпадают с ключами объектов) и связующего отношения, первичным ключом которого будет комбинация ключей объектов.

ПРАВИЛО 6. Если степень связи M:N, то необходимо построение трех отношений: по одному на каждый объект (первичные ключи отношений совпадают с ключами объектов) и связующего отношения, первичным ключом которого будет комбинация ключей объектов.

После этого все неключевые атрибуты распределяются по отношениям.

Пример проектирования отношений

ER – диаграммы



Для каждой ER- диаграммы согласно правилу 4 получаем по 2 отношения:

Учитель (ФИО учителя) **Класс** (Название класса, ФИО учителя).

Класс (Название класса) **Ученик** (ФИО ученика, Название класса).

Класс (Название класса) **Предмет** (Название предмета, Название класса).

Исключая дублирующие отношения и добавляя неключевые атрибуты, получим отношения:

Учитель (ФИО учителя) **Класс** (Название класса, ФИО учителя).

Ученик(ФИО ученика, Название класса, Балл ученика, Дата получения балла).

Предмет (Название предмета, Название класса).

Рассмотрим другой вариант

Это означает, что в каждом классе изучается много предметов и каждый предмет изучается в нескольких классах
К этой диаграмме применяется правило 6, то есть создается 3 отношения:
Класс (Название класса, ФИО учителя).
Предмет (Название предмета)
Предмет_класса (Название класса, Название предмета, Число часов)