

Базы данных

**Реляционная модель
данных**

Реляционная модель данных (РМД)

- В 1970 г. американский математик Э.Ф.Кодд опубликовал статью, с которой отсчитывается начало существования РМД.
- РМД основана на теории множеств.
- Домен, D – множество значений, которые может принимать элемент данных.
- Декартово произведение доменов – множество всех возможных комбинаций значений доменов:
 $D_1 \times D_2 \times \dots \times D_n = \{(d_{1i}, d_{2i}, \dots, d_{ni})\}$, где $d_{ki} \in D_k$
- Пример: $D_1 = (1, 2)$, $D_2 = (a, b, c)$.
 $D_1 \times D_2 = \{(1,a), (1,b), (1,c), (2,a), (2,b), (2,c)\}$
- Отношение – подмножество декартова произведения доменов.

Пример декартова произведения

<i>Должность</i>
директор
инженер
экономист

<i>ФИО</i>
Белов С.Ю.
Рогов А.И.
Панина А.А.
Волкова Н.М.

<i>Оклад</i>
40000
75000

<i>ФИО</i>	<i>Должность</i>	<i>Оклад</i>
Белов С.Ю.	директор	40000
Белов С.Ю.	директор	75000
Белов С.Ю.	инженер	40000
Белов С.Ю.	инженер	75000
Белов С.Ю.	экономист	40000
Белов С.Ю.	экономист	75000
Рогов А.И.	директор	40000
Рогов А.И.	директор	75000
Рогов А.И.	инженер	40000
Рогов А.И.	инженер	75000
Рогов А.И.	экономист	40000
Рогов А.И.	экономист	75000
Волкова Н.М.	экономист	40000

Полужирным шрифтом выделены записи, имеющие соответствие в предметной области.

Пример таблицы реляционной БД

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<i>Год рождения</i>	<u><i>Отдел</i></u>
023	Волкова Елена Павловна	секретарь	26000	1985	2
113	Белов Сергей Юрьевич	инженер	39800	1980	1
101	Рогов Сергей Михайлович	директор	62000	1972	2
056	Панина Анна Алексеевна	инженер-программист	41800	1978	1
...
098	Фролов Юрий Вадимович	начальник отдела	49200	1971	9

Мощность отношения. Арность отношения.

Термины. Свойства отношения

первичный ключ

Отношение, таблица

столбец

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<i>Год рождения</i>
023	Волкова Елена Павловна	секретарь	26000	1985
113	Белов Сергей Юрьевич	инженер	39800	1980

описание (схема отношения)

строка, запись, кортеж

Отношение обладает двумя основными свойствами:

1. В отношении не должно быть одинаковых кортежей, т.к. это множество.
2. Порядок кортежей в отношении несущественен.

Ключи отношения

- **Ключ** – атрибут (группа атрибутов), которые позволяют классифицировать кортеж (запись таблицы).
- **Потенциальный ключ (уникальный ключ)** – атрибут (группа атрибутов), которые позволяют идентифицировать кортеж (запись таблицы).
- **Первичный ключ** – обязательный уникальный ключ. Для каждой таблицы может быть определен только один первичный ключ.
- **Вторичный ключ** – любой другой ключ, кроме первичного. Может быть необязательным и неуникальным.
- **Внешний ключ** – служит для организации связей между таблицами.

Организация связей между таблицами

Связь один-ко-многим: Отделы – Сотрудники

Таблица

«Сотрудники»

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Отдел</i>
023	Волкова Елена Павловна	2
113	Белов Сергей Юрьевич	1
101	Рогов Сергей Михайлович	2
056	Панина Анна Алексеевна	1
...
098	Фролов Юрий Вадимович	9

Таблица «Отделы»

<i>Номер отдела</i>	<i>Название отдела</i>
1	Информационный отдел
2	Администрация
3	Отдел кадров
...	...
9	Проектный отдел

«Номер отдела» - первичный ключ в таблице «Отделы»

«Отдел» – внешний ключ в таблице «Сотрудники»

Организация связей между таблицами

Связь многие-ко-многим: Проекты – Сотрудники

Таблица
«Сотрудники»

<i>ФИО</i>	<i>Номер</i>
Волкова Е.П.	023
Белов С.Ю.	113
Рогов С.М.	101
Панина А.А.	056
Фролов Ю.В.	098
...	...

Таблица «Участие»

<u><i>Участник</i></u>	<i>Роль</i>	<u><i>Проект</i></u>
113	исполнитель	23/Н
101	руководитель	18-К
056	исполнитель	18-К
101	консультант	09/Р
098	руководитель	23/Н
...

Таблица «Проекты»

<i>Шифр</i>	<i>Название проекта</i>
23/Н	АИС "Налог"-2
18-К	ИПС "Жители"
09/Р	ГИС "Город"
...	...

В таблице «Участие»:
«Участник» – внешний ключ к таблице «Сотрудники»
«Проект» – внешний ключ к таблице «Проекты»

Пример связи внутри таблицы



<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<u><i>Начальник</i></u>
023	Волкова Елена Павловна	секретарь	26000	101
113	Белов Сергей Юрьевич	инженер	39800	205
101	Рогов Сергей Михайлович	директор	62000	NULL
205	Махова Ольга Алексеевна	начальник отдела	51300	101
...

Операции над данными в РМД

Операции применяются к кортежам отношений.

В РМД используются следующие операции:

- *запомнить*;
- *извлечь*;
- *обновить*;
- *удалить*.

Сравнение структуризации данных в РМД и по версии CODASYL

<i>Термины версии CODASYL</i>	<i>Термины (и синонимы) РМД</i>
Элемент данных	Атрибут (поле)
Агрегат	—
Запись (группа)	Кортеж (запись, строка)
Совокупность записей одного типа	Отношение (таблица)
Набор (групповое отношение)	Таблицы, связанные внешним ключом
База данных	База данных

Достоинства и недостатки РМД

- Достоинства:
 - ✓ наличие теоретического базиса;
 - ✓ максимальная степень независимости данных от программ;
 - ✓ наличие декларативного языка запросов.
- Недостатки:
 - ✓ низкая эффективность выполнения запросов;
 - ✓ отсутствие однозначного соответствия между сущностями предметной области и таблицами реляционной базы данных.

Объектно-ориентированные базы данных

История вопроса

Начало пути:

- Конец 80-х гг. – образование промышленных компаний и рынка систем управления объектными базами данных (СУОБД).
- Области применения: САПР, промышленность программного обеспечения, географические информационные системы, финансовая сфера, медицинские применения, телекоммуникации, мультимедиа, управляющие информационные системы.
- Лето 1991 г. – образование в США Object Database Management Group (ODMG) – Группы Управления Объектными Базами Данных – как консорциума фирм-производителей СУОБД и других заинтересованных участников для разработки стандарта СУОБД.
- Конец 1993 г. – группа завершила работу опубликованием Стандарта объектных баз данных ODMG-93.
- В состав ODMG во время разработки стандарта ODMG-93 входили: Object Design, Objectivity, Ontos, O2 Technology, SunSoft, Versant. Фирмы-члены ODMG представляли 90% рынка СУОБД.

Постулат того времени: новая парадигма в корне изменит способы проектирования и разработки приложений баз данных!

К разговору об определениях

- Что такое ООСУБД?
- Что такое объектно-ориентированная база данных?

Объектный мир определен в целом настолько расплывчато и нечетко, что невозможно однозначно говорить об основных объектных возможностях.

Сегодня под ООСУБД следует понимать системы, которые следуют духу *Манифеста систем баз данных третьего поколения* и букве стандарта ODMG:1993.

В качестве примеров реализаций ООСУБД можно привести:

GemStone (компания GemStone Systems Inc. (<http://www.gemstone.com/>)).

ITASCA (<http://www.ibex.ch>).

ObjectStore (компания Progress Software (<http://www.objectstore.net>)).

Objectivity /DB (Компания Objectivity (<http://www.objectivity.com>)).

Versant (компания Versant (<http://www.versant.com/>)).

Объектно-ориентированные базы данных



Манифест объектно-ориентированных баз данных

- Архитектура СУОБД, согласно ODMG-93

ODL – *Object Definition Language* (язык определения объектов);

OQL – *Object Query Language* (язык объектных запросов);

OML – *Object Manipulation Language* (язык манипулирования объектами).

□ .



Архитектура СУОБД

- *Инструментальные средства и библиотеки.*
Инструментальные средства, поддерживающие, например, разработку пользовательских приложений и их графических интерфейсов, программируются на одном из OML и сохраняются как часть иерархии классов.
- *Язык определения данных (ODL).* Схемы баз данных описываются в терминах языка ODL, в котором конструкции модели данных конкретизируются в форме языка определения. Кроме того, ODL является *виртуальным* языком в том смысле, что в стандарте ODMG не требуется его реализация в программных продуктах ООСУБД.

Архитектура СУОБД

- *Язык объектных запросов (ODL)*. Язык имеет синтаксис, похожий на синтаксис языка SQL, но опирается на семантику объектной модели ODMG.
- *Языки манипулирования объектами (OML)*. Для программирования реализаций операций и приложений требуется наличия объектно-ориентированного языка программирования. OML представляется собой интегрирование языка программирования с моделью ODMG; это интегрирование производится за счет определенных в стандарте правил *языкового связывания (language binding)*.

Введение в объектную модель ODMG (ООМД)

Модель ODMG – объектная модель данных, включающая возможность описания как объектов, так и литеральных значений.

Модель ODMG подстраивается под специфику систем баз данных следующим образом:

- ✓ Для баз данных, схем и подсхем обеспечивается набор встроенных объектных типов.
- ✓ Модель включает ряд встроенных структурных типов, позволяющих применять традиционные методы моделирования баз данных.
- ✓ Модель одновременно включает понятия объектов и литералов.
- ✓ В модели связи между объектами отличаются от атрибутов объектов (аналогично тому, как это делается в ER -модели).

Введение в объектную модель ODMG

Объекты и литералы

Первое отличие объекта от значения: наличие у объекта уникального идентификатора – OID, Object Identifier (объекты обладают свойством идентифицируемости)

В модели ODMG допускается описание всех данных в терминах объектов и использование традиционного вида значений, которые в модели называются *литеральными значениями*.

Схема базы данных в модели ODMG главным образом состоит из набора объектных типов, но компонентами этих типов могут быть типы литеральных значений.

Введение в объектную модель ODMG

Объекты и литералы

Второе отличие объектов и литералов – понятие ***изменчивости*** (*mutability*). Предположим, например, что данные о человеке составляют структуру

<имя, возраст, адрес проживания> .

Тогда возможны два варианта хранения этих данных:

1. Если человек представляется в виде объекта, то компоненты описывающей его структуры данных могут изменяться.
2. Если же данные о человеке сохраняются в виде литеральной структуры, и один из компонентов этой структуры изменяется, то вся структура трактуется как новое значение.

Атрибутами называются свойства объекта, значение которых можно получить по OID объекта, но не наоборот. Значениями атрибутов могут быть и литералы, и объекты, но только тогда, когда не требуется обратная ссылка.

Введение в объектную модель ODMG

СВЯЗИ

- ✓ Связи являются неотъемлемой характеристикой предметных областей.
- ✓ В большинстве объектных систем связи неявно моделируются как свойства, значениями которых являются объекты.
- ✓ В модели ODMG, подобно ER-модели, различаются два вида свойств – атрибуты и связи.
- ✓ **Связи** – это инверсные свойства. В этом случае значением свойства может быть только объект, поскольку литеральные значения не обладают свойствами. Поэтому возраст служащего обычно моделируется как атрибут, а компания, в которой работает служащий, – как связь.
- ✓ При определении связи должна быть определена ее инверсия. После этого система баз данных должна поддерживать согласованность связанных данных, что позволяет сократить объем работы программистов приложений и повысить надежность их программ.

Введение в объектную модель ODMG

Объектные и литеральные типы

- В модели ODMG база данных представляет собой коллекцию *различимых значений (denotable values)* двух видов – объекты и литералы.
- Модель данных содержит конструкции для спецификации объектных и литеральных типов.
- Объектные типы существуют в иерархии объектных типов; литеральные типы похожи на типы, характерные для обычных языков программирования (например, C или Pascal).
- В модели ODMG не используется термин *класс*. Единственная классификационная конструкция называется *типом*, и типы описывают как объекты, так и литералы.

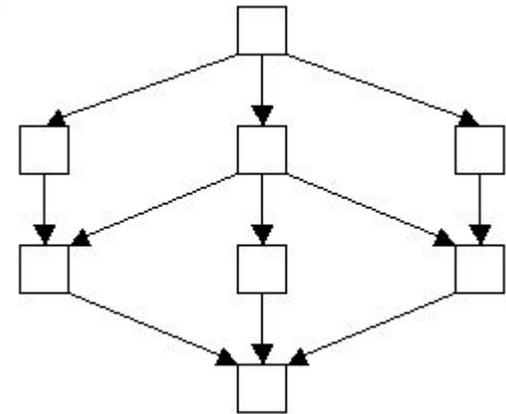
Литеральные типы:

базовые скалярные числовые типы, символьные и булевские типы (атомарные литералы), конструируемые типы литеральных записей (структур) и коллекций.

Введение в объектную модель ODMG

Объектный тип:

- Состоит из интерфейса и одной или нескольких реализаций.
- *Интерфейс* описывает внешний вид типа: какими свойствами он обладает, какие в нем доступны операции и каковы параметры у этих операций.
- В *реализации* определяются структуры данных, реализующие свойства, и программный код, реализующий операции.
- Интерфейс составляет общедоступную (*public*) часть типа, а в реализации при необходимости могут вводиться дополнительные частные (*private*) свойства
- Все объектные типы (системные или определяемые пользователем) образуют решетку (*lattice*) типов, в корне которой находится predeterminedный объектный тип Object.



Введение в объектную модель ODMG

Объектный тип. Определение

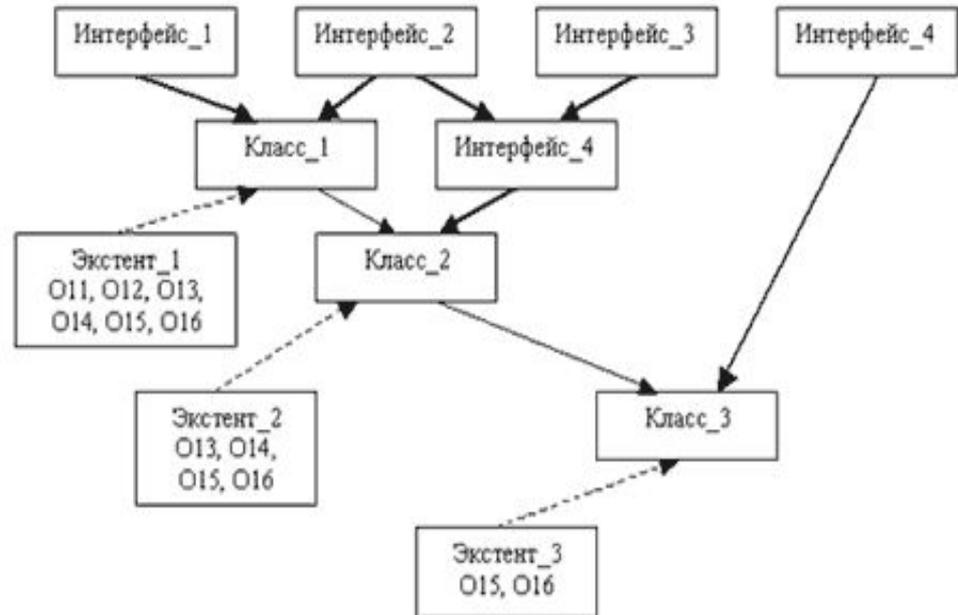
- Допускается создание объектов только тех объектных типов, которые определены как классы.
- Классы могут определяться на основе множественного наследования интерфейсов и одиночного наследования классов.
- Стабильность – это свойство объекта сохранять состояние между сеансами работы программы. Объектная база данных – это, по существу, хранилище стабильных объектов.

Связи явно определяются путем указания *путей обхода (traversal paths)*. Пути обхода указываются парами, по одному пути для каждого направления обхода связи.

Введение в объектную модель ODMG

Объектный тип. Наследование

Механизм наследования от интерфейсов называется в ODMG наследованием *IS - A* (жирные стрелки), а механизм наследования от классов – *extends* (обычные).



Объектно-ориентированные СУБД

GemStone

- ООСУБД GemStone была одной из первых коммерчески доступных ООСУБД.
- Система была разработана компанией Servio-Logic совместно с OGI.
- Разработчики GemStone опирались на язык Smalltalk, потом добавили C и C ++.
- И серверная, и клиентская части системы могут работать под управлением всех основных ветвей ОС UNIX и всех развитых вариантов Windows.
- В настоящее время продукт поддерживается, развивается и распространяется компанией GemStone Systems Inc.
(<http://www.gemstone.com/>).

Объектно-ориентированные СУБД

GemStone

- Для управления мультидоступом используется механизм транзакций.
- Механизм основан на так называемом *оптимистическом* подходе, при котором каждая сессия работает с собственной локальной копией хранилища объектов, и слияние произведенных в сессиях изменений хранилища происходит при завершении транзакции.
- Автоматическая блокировка объектов не производится.
- Для обеспечения безопасности данных поддерживается механизм авторизации доступа на уровне владельца объекта и его группы пользователей.
- К каждому объекту приписывается *авторизационный* объект, содержащий данные о том, какие пользователи и в каком режиме (чтения или изменения) имеют доступ к объекту.

Объектно-ориентированные СУБД

GemStone

- Для восстановления базы данных после сбоев аппаратуры используются механизмы репликации, резервного копирования и журнализации.
- Любой авторизованный пользователь может запросить выполнения полного или частичного копирования, а также поддержку реплицирования областей хранилища данных.
- В системе поддерживается целостность по ссылкам между всеми объектами.
- Объекты делаются стабильными (т.е. сохраняются в базе данных) путем использования своего рода стабильного *корня*, называемого *коннектором*.
- Все объекты, прямо или косвенно достижимые по объектным ссылкам от коннектора, являются стабильными.
- В GemStone для каждого класса, в котором существует хотя бы один стабильный объект, поддерживается эквивалентная серверная версия класса.

Объектно-ориентированные СУБД

GemStone

- В GemStone поддерживается динамическая *сборка мусора (garbage collection)*. Процесс-“мусорщик” автоматически освобождает память, занимаемую объектами, на которые отсутствуют ссылки.
- В среде GemStone можно использовать различные реализации Smaltalk, а также языки С и С++. Классы и объекты можно создавать с использованием любого из этих языков, и объекты, созданные на одном языке можно использовать в приложениях, написанных на любом другом языке.

ITASCA

- Распределенная ООСУБД ITASCA основана на результатах проекта Orion. Разработка серии из трех прототипов завершилась выпуском системы, основанной на архитектуре “много клиентов-много серверов”. Поддерживается компанией IBEX Corp.
- Каждое значение данных хранится в одном узле, но централизованное управление отсутствует; все серверы автономны.