



Кафедра информационных систем

ПРЕЗЕНТАЦИЯ

# ОБЪЕКТНЫЕ БАЗЫ ДАННЫХ

# ОБЪЕКТНЫЕ БАЗЫ ДАННЫХ (ОБД)

## ИСТОКИ ВОЗНИКНОВЕНИЯ ОБД.

ОБД стали разрабатываться с середины 80-х годов в основном для поддержки приложений САПР. Сложные структуры данных систем автоматизированного проектирования оказалось очень удобно оформлять в виде объектов, а технические чертежи проще хранить в БД, чем в файлах. Это позволяет обойтись без декомпозиции графических структур на элементы и записи их в файлы после завершения работы с чертежом, выполнения обратной операции при внесении любого изменения. ОБД хорошо соответствовали подобным задачам, и эволюция многих ОБД началась именно с рынка САПР. Этим объясняются другие их особенности:

- 1) проектирование ОБД для поддержки небольших рабочих групп (порядка 10 человек);
- 2) обработка с помощью ОБД немногочисленных, но продолжительных транзакций;
- 3) ОБД- одни из первых продуктов входящих в распределенные архитектуры клиент/сервер при достаточной обеспеченности клиентов ресурсами.

Таким образом, ОБД появились на свет с возникновением насущной потребности решать задачи, связанные с обработкой и хранением сложных многосвязных данных, а также слабоструктурированной и неструктурированной информации ( текст, изображение, музыка) требующей специфической обработки. С развитием объектной технологии постепенно развивались объектно- ориентированные базы данных (ООБД).

Что же такое объектные базы данных?

Термин объект – « нечто, имеющее четко определенные границы».

Основная идея – объект как единица информации

**Объектные базы данных (ОБД)** – это базы данных, позволяющие хранить полноценные объекты, не сводимые к некоторому заданному внутреннему формату.

**Объектная база данных (ОБД)** – это база данных, построенная на так называемом объектно– ориентированном подходе к структуре БД, который предполагает, что при разработке программы должны быть определены классы используемых в программе объектов и построены их описания, затем созданы экземпляры необходимых объектов и определено взаимодействие между ними.

**Объектно- ориентированная база данных (ООБД)** – это база данных, в которой данные оформлены в виде объектов, включающих прикладные программы, которые управляются внешними событиями.

Большинство ООБД представляют собой библиотеку, процедуры управления данными, которые включаются в прикладную программу.

## ОСОБЕННОСТИ ПОСТРОЕНИЯ ОБД

### Основные свойства объектной модели:

- Базовыми примитивами являются объекты и литералы. Каждый объект имеет уникальный идентификатор. Литерал не имеет идентификатора.
- Объекты и литералы различаются по их типу (рис. 1). Все элементы одного типа имеют одинаковый диапазон изменения состояния (множество свойств) и одинаковое поведение (множество определенных операций). Объект, на который можно установить ссылку, называется экземпляром (хранит какие-либо данные).
- Состояние объекта определяется набором значений, реализуемых множеством свойств. Этими свойствами могут быть атрибуты объекта или связи между объектом и одним или несколькими другими объектами.
- Поведение объекта определяется набором операций, которые могут быть выполнены над объектом или самим объектом. Операции могут иметь список входных и выходных параметров, причем каждый из них строго определенного типа. Каждая операция может также возвращать типизированный результат.
- База данных хранит объекты, позволяя их совместно использовать различным пользователям и приложениям.
- **Методы ОБД** – это программный код, привязанный к конкретному классу и применимый к объектами этого класса. В ОБД O2 определение метода происходит в два этапа. Сначала объявляется его имя, типы и классы аргументов и тип и класс результата. Методы могут быть публичными (доступными из объектов других классов) или приватными (доступными внутри данного класса). На втором этапе определяется реализация класса на одном из языков программирования O2.

Рис 1. Основные элементы объектной модели



## ТИП И СТРУКТУРА ДАННЫХ ОБЪЕКТНОЙ МОДЕЛИ ДАННЫХ

Объектная модели данных база данных – это набор объектов произвольного типа.

В объектной модели данных вводятся две разновидности типов: *литеральные* и *объектные* типы.

Литеральные типы данных – это обычные типы данных, принятые в традиционных языках программирования. Они подразделяются на базовые скалярные числовые типы, символьные и булевские типы (*атомарные литералы*), конструируемые типы записей (*структур*) и *коллекций*.

Литеральный тип записи – это традиционный определяемый пользователем структурный тип, подобный структурному типу языка C или типу записи языка Pascal. Отличие состоит лишь в том, что в объектной модели атрибут типа записи может определяться не только на литеральном, но и на объектном типе, т.е. значение литерального типа записи может в качестве компонентов включать объекты.

Имеются четыре вида типов коллекций: типы множеств, мультимножеств (неупорядоченные наборы элементов, возможно, содержащие дубликаты), списков (упорядоченные наборы элементов, возможно, содержащие дубликаты) и словарей (множества пар <ключ, значение>, причем все ключи в этих парах должны быть различными). Типом элемента любой коллекции может являться любой скалярный или объектный тип за исключением того же типа коллекции.

Объектные типы в объектной модели данных по смыслу ближе всего к понятию *класса* в объектно-ориентированных языках программирования.

Имеются два вида объектных типов. Первый из них называется атомарным объектным типом. Нестрого говоря, при определении атомарного объектного типа указывается его внутренняя структура (набор свойств – атрибутов и связей) и набор операций, которые можно применять к объектам этого типа. Для определения атомарного объектного типа можно использовать механизм наследования, расширяя набор свойств и/или переопределяя существующие и добавляя новые операции.

*Атрибутами* называются свойства объекта, значение которых можно получить по OID объекта. Значениями атрибутов могут быть и литералы, и объекты (т.е. OID), но только тогда, когда не требуется обратная ссылка. *Связи* – это инверсные свойства. В этом случае значением свойства может быть только объект. Связи определяются между атомарными объектными типами. В объектной модели ODMG поддерживаются только бинарные связи, т.е. связи между двумя типами.

Второй вид – это объектные типы коллекций. Как и в случае использования литеральных типов коллекций, можно определять объектные типы множеств, мультимножеств, списков и словарей. Типом элемента объектного типа коллекции может быть любой литеральный или объектный тип за исключением того же типа коллекции. У объектных типов коллекций имеются предопределенные наборы операций. В отличие от литеральных типов коллекций, которые, как и все литеральные типы являются множествами значений, объектные типы коллекций обладают операцией создания объекта, имеющего, как и все объекты, собственный OID.

## ОБЪЕКТНАЯ БАЗА ДАННЫХ НА ПРИМЕРЕ СУБД ODB-Jupiter

**СУБД ODB-Jupiter** – это оригинальная разработка научно-производственного центра «Интелтек Плюс» (г. Москва). Основные цели данной разработки: обработка неструктурированной информации, хранение документов сложной Структуры, встроенная поддержка полнотекстового поиска, возможность расширения наборов типов СУБД. Клиентом СУБД может быть как любая программа, использующая клиентскую библиотеку СУБД, так и браузер Интернет.

СУБД **ODB-Jupiter** построена по многоуровневой схеме, в которой можно четко выделить четыре «этажа» – Управление Файлами базы данных, Deskрипторный уровень СУБД, Объектный уровень и Приложение с логикой связи объектов и Логикой связи данных.

**Управление файлами базы данных** - база данных представляет собой хранилище структурированных записей переменной Длины и библиотеку, реализующую набор операций над записями: добавление/чтение/замещение/удаление.

На этом уровне решается задача поддержания целостности данных. Каждой записи при добавлении присваивается Уникальный номер – идентификатор, который однозначно определяет положение записи в базе данных.

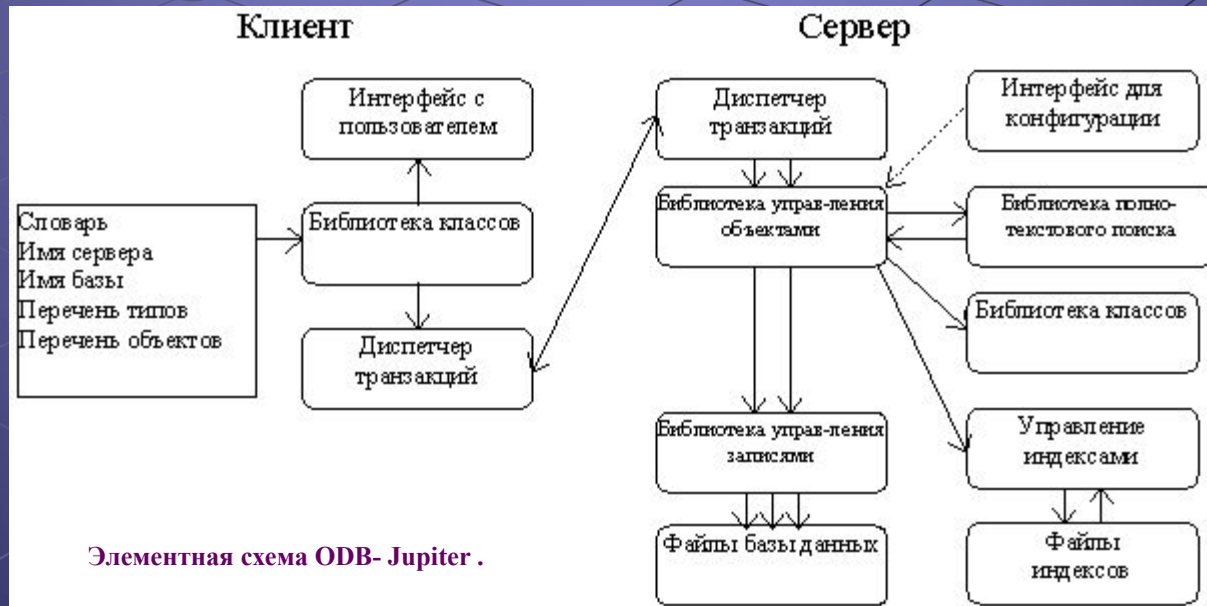
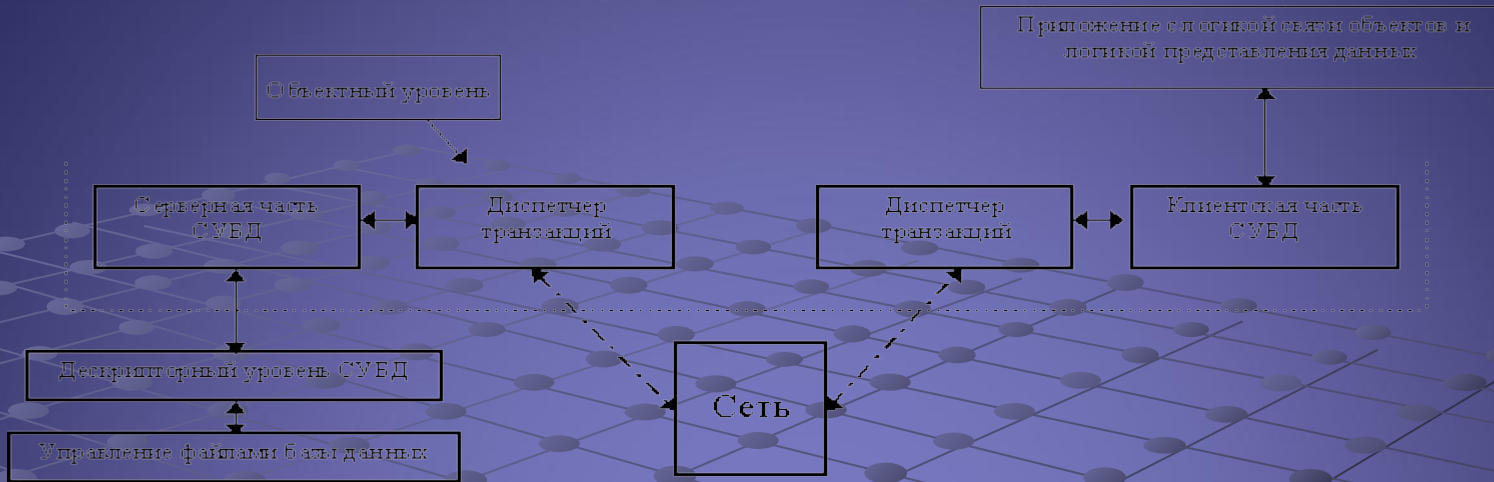
**Deskрипторный уровень СУБД** – на этом уровне известны имена типов записей, хранящихся в БД и присутствуют методы Для добавления компонентам структурированной записи. Таким образом, возможно оперативно извлекать информацию без Загрузки объекта в оперативную память целиком.

**Объектный уровень** – присутствуют объекты БД, которые взаимодействуют по определенным правилам. Объектный Уровень предоставляет обширный программный интерфейс ООСУБД. Программный интерфейс СУБД собран в одной динамически загружаемой библиотеке. Разработка приложения никак не ограничивается количеством используемых Узлов сети и БД,

Внутренняя структура **СУБД ODB-Jupiter** представлена основными блоками: Файлы БД, Файлы индексов, Библиотека Управления записями, Библиотека управления индексами, Библиотека управления объектами, Библиотека классов (клиента и сервера), Диспетчер транзакций.

Внутренний мир ОБД представлен тремя разновидностями объектов: Контейнеры данных, Индексаторы и Пользователи.

# УРОВНЕВАЯ СИСТЕМА ODB- Jupiter



Элементная схема ODB- Jupiter .

## СРАВНЕНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СУБД И РЕЛЯЦИОННОЙ СУБД

Признак	ООСУБД	РСУБД
Модель данных	Объектная	Табличная
Способ чтения объектов	Данные объекта помещаются в хранилище как единое целое	Объект собирается из отдельных элементов и только затем используется
Характеристика отношений между объектами	Более совершенные средства для отображения реального мира. Естественное представление данных на разных уровнях	Все отношения принадлежат 1 уровню
Новые типы данных	Не требует модификации ядра при добавлении нового типа данных	Необходима модификация ядра
Оптимизация ядра СУБД	Ядро оптимизировано для операций с объектами.	Ядро остается реляционным
Язык СУБД и запросы	Разновидность реализации OQL	Различные варианты SQL

По мнению идеолога СУБД **Мэри Луинса** актуальность объектного подхода к базам данных состоит в том, что «модель данных более близка сущностям реального мира. Объекты можно сохранить и использовать непосредственно не раскладывая их по таблицам. Типы данных определяются разработчиком и неограниченны набором predetermined типов».

## СОВРЕМЕННЫЕ БАЗЫ ДАННЫХ. ОСНОВНЫЕ ОБЛАСТИ ИХ ПРИМЕНЕНИЯ.

POET (компания POET Software) – сравнительно недорогая и скромная по ресурсам ОБД.

Одна из ее особенностей – компактность. Ядро базы данных занимает около 1 Мб. Ее можно рекомендовать к Использованию в среде Windows, Windows NT. Необходимо отметить, что БД имеет как собственную среду разработки, Так и средства стыковки интерфейса базы с популярными компиляторами C++ компаний Borland, Microsoft. Программный интерфейс поддерживает Java, Visual Basic, Active/X элементы.

JASMINE (компания Computer Associate), наиболее известная ОСУБД. Будучи одной из новейших БД, вобрала в себя Опыт предыдущих работ, добавив много нужных для объектного программирования новшеств. БД снабжена визуальной Средой разработки JASMINE STUDIO, которая отличается удобным и продуманным и понятным интерфейсом.

VERSANT (разработка Versant Technologies). С использованием этой программы был реализован ряд крупных проектов ( в первую очередь в области телекоммуникаций).

O2 (компания Arden Software) - живая «классика» среди ОБД. Используется в области коммуникаций и систем специального применения.

### ОСНОВНЫЕ ОБЛАСТИ ПРИМЕНЕНИЯ ОБД










По мнению аналитиков, в настоящее время существует три основных типа приложений, в которых можно с успехом Использовать ОБД.

- 1) Системы, требующие высокопроизводительного доступа к данным в оперативной памяти ( телекоммуникации)
- 2) OLTP – системах со сложными данными ( приложения для сферы торговли, финансов, здравоохранения).
- 3) Приложения, интегрирующие разнородные данные( хранилищ информации или систем поддержки принятия решений DSS), для интеграции унаследованных источников данных с системами клиент/сервер), где они позволяют обойтись без сложных шлюзов и специализированных решений.




## ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ОБД

### Преимущества:

-  Высокая производительность ( манипулирование сложными данными);
-  Упрощенная модель БД, возможность повторного ее использования, простота организации отношений « многие – комногим».
- Масштабируемость ( объекты сохраняются в БД и могут использоваться неоднократно);
-  Целостность ( все помещаемые в БД объекты удовлетворяют ее схеме);
-  Надежность и безопасность ( БД возвращается в исправное состояние после программных и системных сбоев; защита от несанкционированного доступа, использование средств аутентификации пользователей);
-  Вычислительная полнота ( способность языка БД своими средствами решать все вычислительные задачи);
-  Независимость данных ( способность изменять схему БД без постоянного доступа к данным);
-  Гибкость, продуктивность, модульность.
-  Эффективное управление компонентами БД .
-  Высокий уровень абстракции .

### Недостатки:

-  Вынужденная простота БД;
-  Отсутствие единого стандарта БД;
-  Недостаточная безопасность ( пользователи должны постоянно устанавливать и снимать блокировки);
-  Недостаточная оптимизация запросов ( из-за сложных объектов БД имеется отсутствие мощных непроцедурных средств извлечения объектов из БД, все запросы приходится писать на процедурных языках);
-  Отсутствие развития средств манипулирования данными( необходимо расширять ОО- язык в сторону управления данными).

## ПЕРСПЕКТИВЫ ОБД

Существует несколько факторов, позволяющих сделать вывод о достаточно хороших перспективах объектных баз данных:

- 1) Объектные СУБД – открытые системы. Сравнительно несложно добавить новые типы данных. Сделать это несравненно проще, чем в Oracle, Informix. Это большой плюс, поскольку открытые системы более привлекательны для конкретных пользователей.
- 2) Большинство производителей ОБД предоставляют визуальные средства создания прикладных программ объектной СУБД. Таким образом, если раньше созданием прикладных программ для ОСУБД занимались в основном «зубры» -
- 3) Специалисты в C++, Smalltalk, то теперь пользоваться объектными свойствами базы стало намного проще.
- 4) Бум объектных средств моделирования. Совмещение объектного средства моделирования ОБ естественно в отличие от интеграции такого средства с реляционной БД.
- 5) Существует много областей, где применение реляционных баз данных нецелесообразно по разным причинам. Например, в прикладных программах САПР, телекоммуникациях встречаются многосвязные данные. Если хранить информацию в реляционной БД, то запросы будут выполняться недостаточно быстро. ОБД имеют отличные перспективы в прикладных задачах САПР, CASE, геодезии, финансах, телекоммуникациях, оборонной промышленности.

A 3D perspective grid of dots on a blue background. The dots are arranged in a regular pattern, receding into the distance. The background is a gradient of blue, darker at the bottom and lighter at the top.

СПАСИБО ЗА ВНИМАНИЕ !