

# БД и СУБД (Проектирование БД)

Аблов Игорь Васильевич, преподаватель  
кафедры информационных систем

# Введение

- **В рамках курса определяются основные вопросы выбора СУБД при проектировании информационных систем, обоснование выбранного решения по набору оцениваемых показателей. А также уточняется момент принятия решения и обоснования**

# Что будет изучено,

- Терминология СУБД.
- Основные этапы проектирования информационной системы и ее базы данных: текстовое описание задачи, построение концептуальной и физической модели, генерация программного кода для создания базы данных.
- Роль и место СУБД в прикладных системах.
- Основные функции СУБД.
- Взаимодействие СУБД с другими компонентами программного обеспечения.
- Критерии выбора СУБД

# Словарь (проектирование)

- *Проектирование* – это разработка принципов построения и эффективного функционирования систем, процессов и др. Под проектированием любого объекта понимается процесс построения его образа, используемого затем для определённой (заданной) цели.

# Словарь (СУБД)

- **База данных (БД, database)** - поименованная совокупность структурированных данных, относящихся к определенной предметной области.
- **Предметная область** - некоторая часть реально существующей системы, функционирующая как самостоятельная единица. Полная предметная область может представлять собой экономику страны или группы союзных государств, однако на практике для информационных систем наибольшее значение имеет предметная область масштаба отдельного предприятия или корпорации.
- **Система управления базами данных (СУБД)** - комплекс программных и языковых средств, необходимых для создания и модификации базы данных, добавления, модификации, удаления, поиска и отбора информации, представления информации на экране и в печатном виде, разграничения прав доступа к информации, выполнения других операций с базой.

# Словарь (объекты БД)

- **Таблица** базы данных (table) - регулярная структура, которая состоит из однотипных строк (записей, records), разбитых на столбцы (поля, fields).
- **Ключевой элемент таблицы** (ключ, regular key) – поле, набор полей таблицы, строковое выражение, образованное из значений нескольких полей, по которому можно определить значения других полей для одной или нескольких записей таблицы. (Первичный, альтернативный)
- **Связь** (relation) - функциональная зависимость между объектами. (Как правило, порождает ограничения и внешние ключи)
- **Хранимые процедуры** (stored procedures) - программные модули, сохраняемые в базе данных для выполнения определенных операций с информацией базы.
- **Триггеры** (triggers) - хранимые процедуры, обеспечивающие соблюдение условий ссылочной целостности данных в операциях изменения первичных и внешних ключей.

# Взаимосвязь терминов

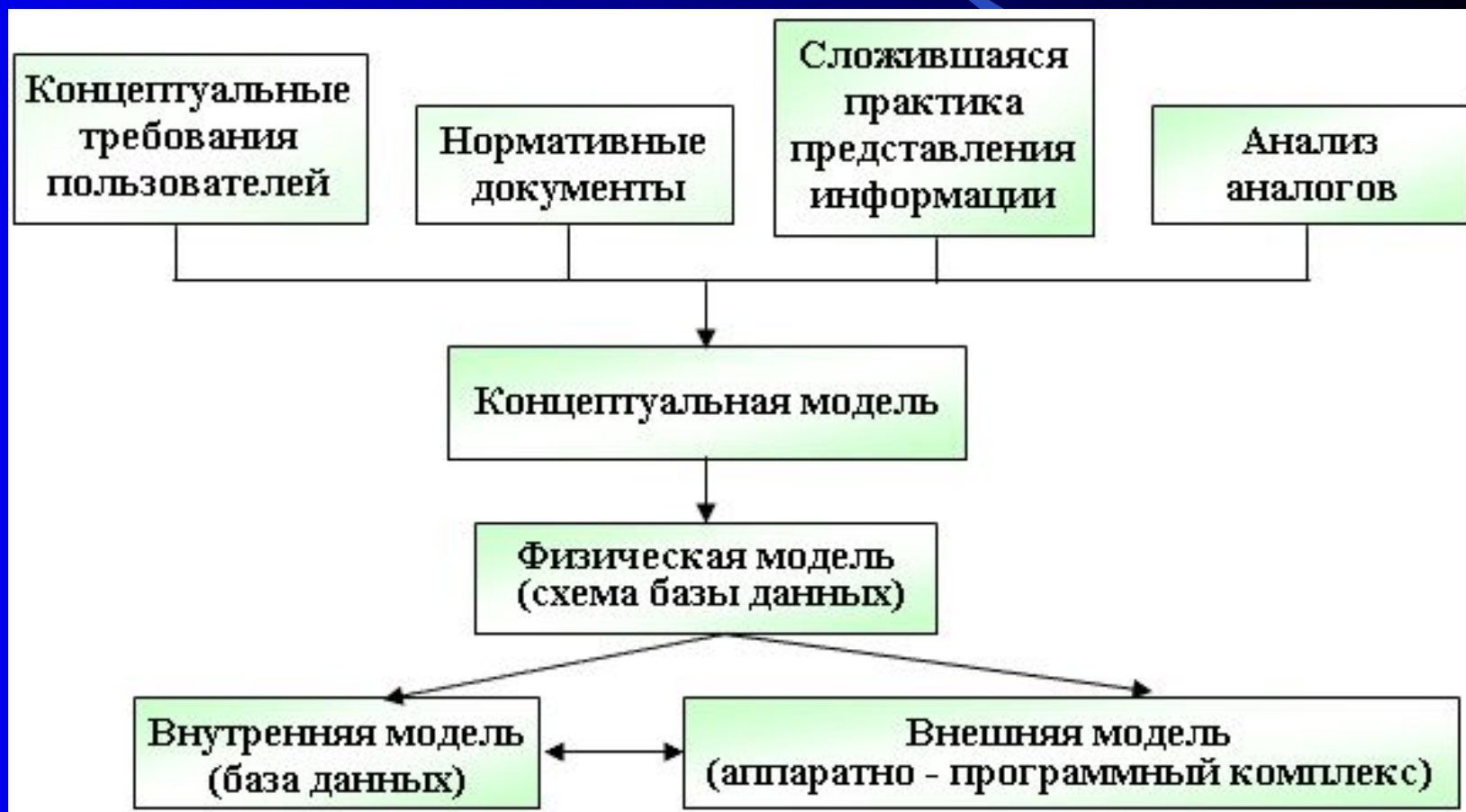


## Словарь (модели)

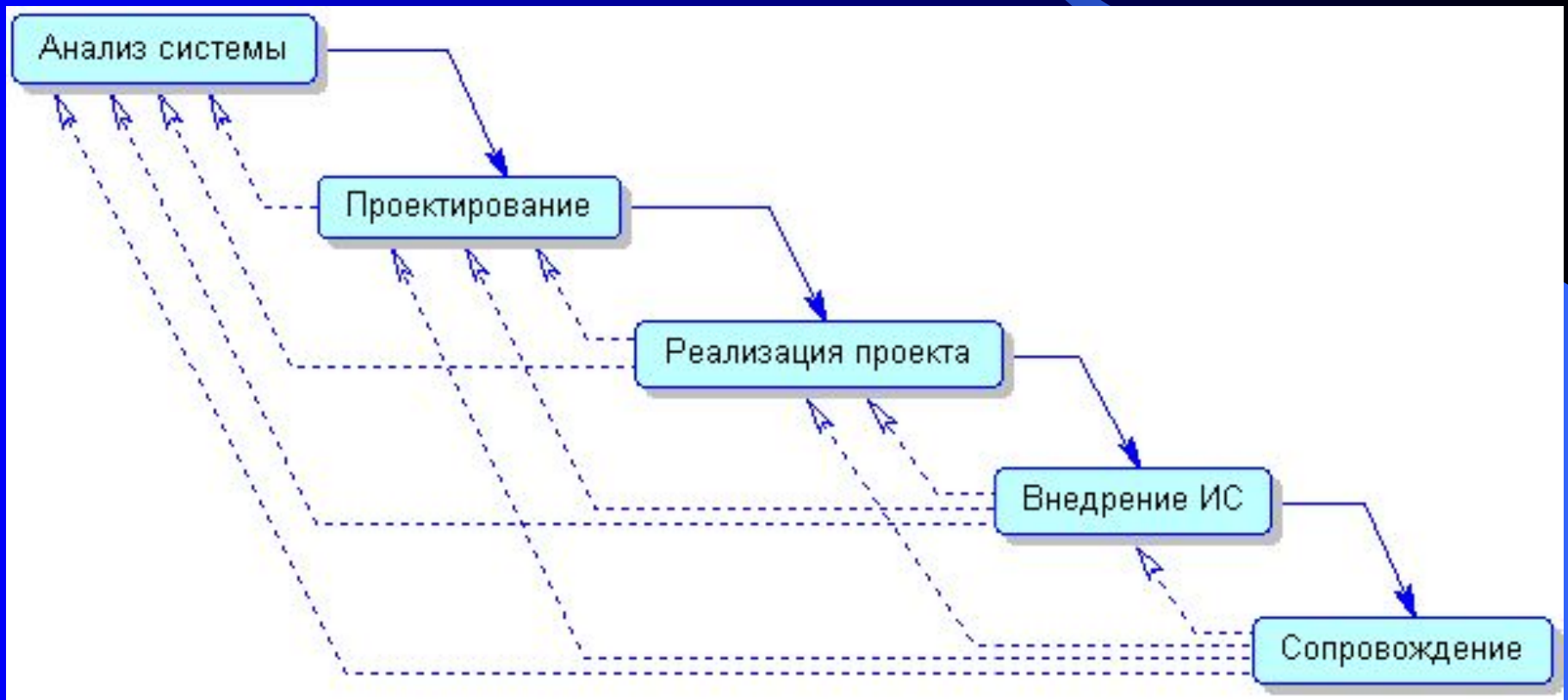
- **Концептуальная модель** - отображает информационные объекты, их свойства и связи между ними без указания способов физического хранения информации (модель предметной области, иногда ее также называют информационно-логической или **инфологической моделью**). Информационными объектами обычно являются **сущности** - обособленные объекты или события, информацию о которых необходимо сохранять, имеющие определенные наборы свойств - **атрибутов**.
- **Физическая модель** - отражает все свойства (атрибуты) информационных объектов базы и связи между ними с учетом способа их хранения - используемой СУБД.
- **Внутренняя модель** - база данных, соответствующая определенной физической модели.
- **Внешняя модель** - комплекс программных и аппаратных средств для работы с базой данных, обеспечивающий процессы создания, хранения, редактирования, удаления и поиска информации, а также решающий задачи выполнения необходимых расчетов и создания выходных печатных форм.



# Схема формирования информационной модели



# Жизненный цикл разработки



# Этапы проектирования БД

- Этап 1. Определение сущностей.
- Этап 2. Определение взаимосвязей между сущностями.
- Этап 3. Задание первичных и альтернативных ключей, определение атрибутов сущностей.
- Этап 4. Приведение модели к требуемому уровню нормальной формы.
- Этап 5. Физическое описание модели.

# Варианты терминологии

- *Инфологическая модель данных - обобщенное неформальное описание создаваемой базы данных, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств, понятных всем людям, работающим над проектированием базы данных.*
- *Даталогическая модель — описание на языке конкретной СУБД*

# Вариант терминологии

На основе даталогической модели строится физическая модель. Физическая организация данных оказывает основное влияние на эксплуатационные характеристики БД. Разработчики СУБД пытаются создать наиболее производительные физические модели данных, предлагая пользователям тот или иной инструментарий для поднастройки модели под конкретную БД.

**Пример.** Для реляционной БД она уже учитывает:

1. физические аспекты хранения таблиц в определенных файлах,
2. создания индексов, оптимизирующих скорость выполнения операций над данными с помощью приложения,
3. выполнения различных действий над данными при определенных событиях, определяемых пользователем с помощью триггеров, хранимых процедур.

# Классификация СУБД, типы СУБД, виды СУБД

- По модели данных
- По архитектуре СУБД и организации хранения данных
- По способу доступа СУБД к базе данных

# По архитектуре СУБД и организации хранения данных

- локальные СУБД (все части локальной СУБД размещаются на одном компьютере);
- распределенные СУБД (части СУБД могут размещаться на двух и более компьютерах).

# По способу доступа СУБД к базе данных

- **Файл-серверные СУБД.**
- **Клиент-серверные СУБД.**
- **Встраиваемые СУБД.**



# Файл-серверные

- В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере СУБД. Ядро СУБД располагается на каждом клиентском компьютере. Доступ к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на ЦП сервера, а недостатком — высокая загрузка локальной сети.

# Клиент-серверные СУБД

- Такие СУБД состоят из клиентской части (которая входит в состав прикладной программы) и сервера СУБД (см. Клиент-сервер). Клиент-серверные СУБД, в отличие от файл-серверных, обеспечивают разграничение доступа между пользователями и мало загружают сеть и клиентские машины. Сервер является внешней по отношению к клиенту программой, и по надобности его можно заменить другим. Недостаток клиент-серверных СУБД в самом факте существования сервера СУБД (что плохо для локальных программ — в них удобнее встраиваемые СУБД) и больших вычислительных ресурсов, потребляемых сервером.

# Встраиваемые СУБД

- Встраиваемая СУБД — библиотека, которая позволяет унифицированным образом хранить большие объёмы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объёмами данных (например, геоинформационные системы).

# По числу пользователей

- Однопользовательская
- Многопользовательская

# Список СУБД

- 4th Dimension
- Adabas D
- Alpha Five
- Apache Derby
- Asql
- CA-Datcom
- CSQL
- Daffodil database
- DataEase
- Dataphor
- Derby, известный как Java DB
- EnterpriseDB
- eXtremeDB
- FileMaker Pro
- Firebird
- Gladius DB
- H2
- Helix database
- HSQLDB
- IBM DB2
- IBM DB2 Express-C
- Informix
- Ingres
- InterBase
- Intersystem Cache Database
- Kognitio
- Linter
- MaxDB
- Mckoi SQL Database
- Microsoft Access
- Microsoft Jet Database Engine
- Microsoft SQL Server
- Microsoft SQL Server Express
- Microsoft Visual FoxPro
- Mimer SQL
- MonetDB
- MongoDB
- mSQL
- MySQL
- Netezza
- NonStop SQL
- Openbase
- OpenLink Virtuoso (Open Source Edition)
- OpenLink Virtuoso Universal Server
- Oracle
- Oracle Rdb for OpenVMS
- Pervasive
- PostgreSQL
- Progress 4GL
- RDM Embedded
- RDM Server
- The SAS system
- Sav Zigzag
- ScimoreDB
- SmallSQL
- solidDB
- SQLBase
- SQLite
- Sybase Adaptive Server Enterprise
- Sybase Adaptive Server IQ
- Sybase SQL Anywhere
- tdbengine
- Teradata
- TimesTen
- txtSQL
- Valentina
- Vertica
- VistaDB
- VMDS

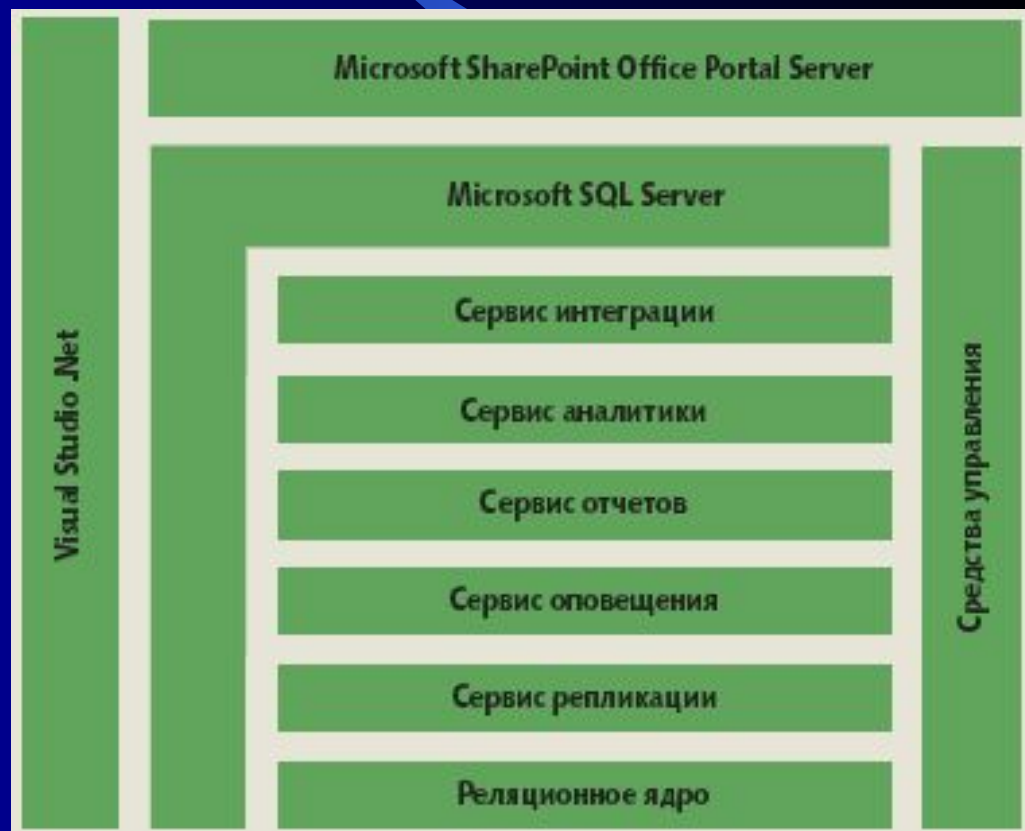
# Архитектура СУБД POSTGRES95

- **postmaster** - управляющий процесс-демон.
- **postgres** - внутренний серверный процесс базы данных, исполняющий запросы клиента.
- **внешняя прикладная программа**, которая может находиться на другом компьютере



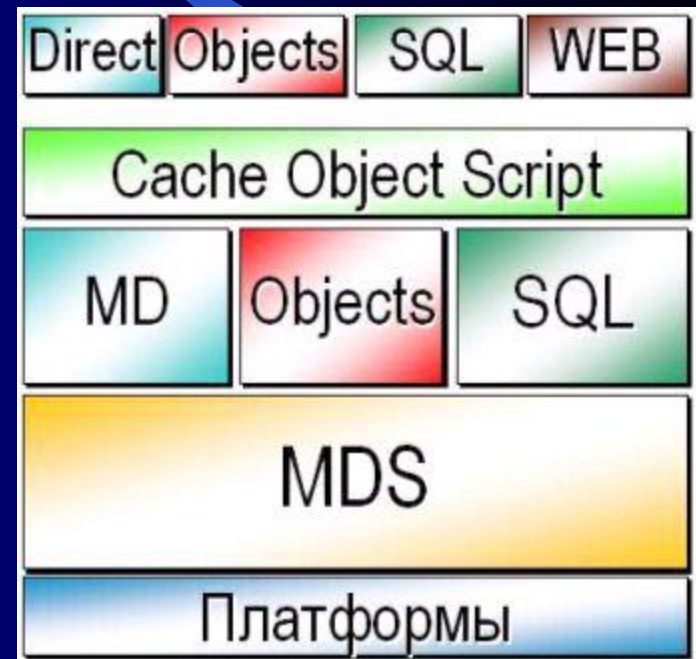
# MS SQL 2005 и выше

- Сервер БД
- Сервер приложений
- Web-сервер



# Архитектура Cache

Принципиальное отличие от других СУБД – наличие набора интерфейсов доступа к данным многомерного сервера, функционирующего на самых разных платформах





# Состав сервера СУБД

Обычно современные СУБД или сервер СУБД, содержат следующие компоненты:

- **ядро** , которое отвечает за управление данными во внешней и оперативной памяти и журнализацию;
- **процессор языка базы данных** , обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода;
- **подсистему поддержки времени исполнения** , которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД;
- а также **сервисные программы** (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

# Функции ядра СУБД

- **Непосредственное управление данными во внешней памяти**
- **Управление буферами оперативной памяти**
- **Управление транзакциями**
- **Журнализация**
- **Поддержка языков БД**

# Непосредственное управление данными во внешней памяти

Эта функция включает обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для ускорения доступа к данным в некоторых случаях (обычно для этого используются индексы). В некоторых реализациях СУБД активно используются возможности существующих файловых систем, в других работа производится вплоть до уровня устройств внешней памяти. Но подчеркнем, что в развитых СУБД пользователи в любом случае не обязаны знать, использует ли СУБД файловую систему, и если использует, то, как организованы файлы. В частности, СУБД поддерживает собственную систему именования объектов БД.

# Управление буферами оперативной памяти

СУБД обычно работают с БД значительного размера; по крайней мере, этот размер обычно существенно больше доступного объема оперативной памяти. Понятно, что если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти. При этом, даже если операционная система производит общесистемную буферизацию (как в случае ОС UNIX), этого недостаточно для целей СУБД, которая располагает гораздо большей информацией о полезности буферизации той или иной части БД. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов.

# Управление транзакциями

- С управлением транзакциями в многопользовательской СУБД связаны важные понятия *сериализации транзакций* и *сериального плана выполнения смеси транзакций*. Под сериализацией параллельно выполняющихся транзакций понимается такой порядок планирования их работы, при котором суммарный эффект смеси транзакций эквивалентен эффекту их некоторого последовательного выполнения. Сериальный план выполнения смеси транзакций - это такой план, который приводит к сериализации транзакций. Понятно, что если удастся добиться действительно сериального выполнения смеси транзакций, то для каждого пользователя, по инициативе которого образована транзакция, присутствие других транзакций будет незаметно (если не считать некоторого замедления работы по сравнению с однопользовательским режимом).
- Существует несколько базовых алгоритмов сериализации транзакций. В централизованных СУБД наиболее распространены алгоритмы, основанные на синхронизационных захватах объектов БД. При использовании любого алгоритма сериализации возможны ситуации конфликтов между двумя или более транзакциями по доступу к объектам БД. В этом случае для поддержания сериализации необходимо выполнить откат (ликвидировать все изменения, произведенные в БД) одной или более транзакций. Это один из случаев, когда пользователь многопользовательской СУБД может реально (и достаточно неприятно) ощутить присутствие в системе транзакций других пользователей.

# Журнализация

- Журнал - это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. В разных СУБД изменения БД журналируются на разных уровнях: иногда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД), иногда - минимальной внутренней операции модификации страницы внешней памяти; в некоторых системах одновременно используются оба подхода.
- Для восстановления БД после жесткого сбоя используют журнал и архивную копию БД. Грубо говоря, архивная копия - это полная копия БД к моменту начала заполнения журнала



# Поддержка языков БД

- Для работы с базами данных используются специальные языки, в целом называемые *языками баз данных*. В ранних СУБД поддерживалось несколько специализированных по своим функциям языков. Чаще всего выделялись два языка - *язык определения схемы БД (SDL - Schema Definition Language)* и *язык манипулирования данными (DML - Data Manipulation Language)*. SDL служил главным образом для определения логической структуры БД, т.е. той структуры БД, какой она представляется пользователям. DML содержал набор операторов манипулирования данными, т.е. операторов, позволяющих заносить данные в БД, удалять, модифицировать или выбирать существующие данные.

# Поддержка языков БД 2

- В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language).
- DDL – Data Definition Language
- DML – Data Manipulation Language



# Классическая методология проектирования

- Сбор сведений о Предметной Области (ПрО)
- выбор языка представления т.н. "семантической" модели для фиксации сведений о ПрО, их последующего анализа и синтеза модели БД;
- анализ собранных сведений о ПрО: классификация, формализация и интеграция структурных элементов описания ПрО, формализация как структурных, так и процедурных ограничений целостности элементов в будущей модели ПрО, определение динамики экземпляров объектов ПрО
- синтез концептуальной модели БД: проектирование целостной концептуальной схемы БД на выбранном языке семантического моделирования

# Классическая методология проектирования

- выбор конкретной модели данных и СУБД для реализации БД
- проектирование логической схемы БД для выбранной СУБД (называющееся также "проектирование реализации")
- разработка физической структуры БД ("физической" или "внутренней" схемы, она же - "схема размещения"), включая размещение БД по узлам;
- разработка технологии и процедур начального создания и заполнения БД;
- разработка технологии и процедур сопровождения БД;

# Классическая методология проектирования

- разработка универсальных программ доступа к БД и соответствующих интерфейсов пользователей;
- информационное обеспечение разработки конкретных программ обработки данных: обеспечение метайнформацией, данными контрольных примеров и др.;
- получение обратной связи от разработчиков прикладных программ и пользователей Информационной Системы (ИС) о полноте и эффективности организации БД;
- тестирование БД, ее развитие и улучшение (настройка) ее структуры.

# Обеспечение эксплуатационных характеристик БД

На этапах проектирования для определения рациональной физической схемы БД от способов определения временных характеристик нужно следующее:

- возможности сравнения временных параметров вариантов реализации разных вариантов схемы БД, на некоторой СУБД;
- возможности сравнения параметров вариантов реализации одной схемы БД на разных СУБД;
- возможности сравнения параметров реализации одной схемы БД на разных аппаратных серверах БД;
- возможности предсказания временных параметров работы различных прикладных программ и служебных программ-утилит.

# Как сравнивать СУБД?

- <http://www.microsoft.com/sqlserver/2008/ru/ru/compare-oracle.aspx>
- <http://www.microsoft.com/sqlserver/2008/ru/ru/compare-mysql.aspx>
- <http://www.microsoft.com/sqlserver/2008/ru/ru/compare-ibm-db2.aspx>

# Экспертиза СУБД

- качеством конкретной СУБД как таковой,
- сравнением характеристик нескольких СУБД,
- пригодностью СУБД для конкретного проекта,
- особыми ситуациями, возникшими при использовании СУБД.

# Результат на практике

- Но в целом картина была такова:
- \* очень часто экспертиза проводилась для принятия решения о выборе СУБД: часто для одного проекта, но также – в качестве основной платформы для ИС организации,
- \*\* в этих и иных случаях цель экспертизы формально очень часто была связана с оценкой технических параметров СУБД, а среди них чаще всего – с эффективностью СУБД (временные характеристики и аппаратная ресурсоемкость),
- \*\*\* итоговые решения о выборе СУБД реально чаще принимались под влиянием политических и культурологических факторов, чем на основании значений технических характеристик СУБД.

# Тенденции выбора

- Зарубежные специалисты в первую очередь выбирают СУБД для конкретного проекта (предусматривая у себя в хозяйстве более одной «серьезной» СУБД) и опираются в первую очередь на рыночные и политические факторы,
- Наши специалисты чаще выбирают СУБД как «единственную и на всю жизнь» (попадая тем самым в большую зависимость от результата такого выбора), при этом они сообщают, что для них наиболее значимы технические параметры СУБД, а среди них – показатели эффективности.



# Характеристика проблемы выбора

Характеристика проблемы выбора СУБД для организации (или для расширяющегося комплекса проектов организации) состояла в том, что:

- выбор СУБД преимущественно на основе сравнения их детальных характеристик и детальных требований проекта часто становится тормозом, так как набор этих характеристик и требований столь велик, что их оценка будет производиться дольше, чем смена версий этой и конкурирующих СУБД, а также изменение требований проектов (и появление новых),
- СУБД – это не только «система управления базами данных», но еще и комплекс инструментов для разработчиков и пользователей, а также совокупность прикладных программ, поставляемых третьими фирмами или разработчиком СУБД.

# Уровни применения критериев оценки СУБД

Каждый из критериев, входящих в эти группы, связывается с требованиями какого-то уровня применения критерия (и экспертизы в целом), а именно, с требованиями:

- конкретного проекта ИС,
- организации-пользователя СУБД,
- формальных или фактических стандартов,
- рынка,
- перспективных значений параметров качества (экономических, технических, эргономических и иных), признанных таковыми в результате специальных исследований (технических, маркетинговых).

# Приоритеты отечественных разработчиков

Приоритеты отечественных разработчиков отличаются особенностями:

- критерий «технические характеристики» выделяется по важности,
- второе место занимают финансовые критерии (опрошенные зарубежные фирмы финансовый критерий сдвигали на 4-е место),
- в технических характеристиках на первое место ставилась «производительность», затем — «надежность» и «средства разработки» (опрошенные зарубежные фирмы критерий производительности сдвигали на места с 3-го по 6-е).

# Четыре вида временных характеристик СУБД

- Интегральные временные характеристики СУБД
- Элементарные временные характеристики СУБД
- Паспортные временные характеристики СУБД
- Проектные временные характеристики СУБД

Более подробно в статье

<http://www.masters.donntu.edu.ua/2002/fvti/pryadko/diss/library/4.htm>

# Сравнение СУБД для веб-проектов

- Поддерживаемые операционные системы
- Условия лицензирования
- Процесс установки и поддержки
- Наличие предустановленных драйверов в ОС семейства Windows
- Наличие драйверов ODBC, JDBC, ADO.NET
- Наличие View, доступных только для чтения
- Наличие программных продуктов с открытым исходным кодом, основанных на этой СУБД
- Использование в коммерческих проектах
- Обновляемые View
- Поддержка
- Materialized/Indexable Views
- Возможность добавлять столбцы, изменять названия, типы данных для view без их уничтожения
- Наличие графического ПО для конструирования и оптимизации запросов
- Наличие Computed Columns
- Поддержка функциональных индексов
- Поддержка частичных индексов
- Поддержка ACID-требований к транзакциям
- Каскадное обновление/удаление внешних ключей

# Сравнение СУБД для веб-проектов

- Внесение данных в несколько строк
- Поддержка UPSERT-логики (это когда происходит вставка, если поле пустое и обновление, если поле не пустое)
- Поддержка репликации
- Возможность писать хранимые функции на разных языках программирования
- Возможность создавать пользовательские агрегированные функции
- Поддержка триггеров
- Партиционирование таблиц
- Возможность создавать функции, возвращающие таблицу или набор таблиц, которые можно использовать в секции FROM запросов.
- Поддержка создания функций
- Поддержка хранимых процедур
- Поддержка динамического SQL в функциях
- Бесплатное ПО для графического управления БД
- Наличие встроенного планировщика (не CronTab)
- Возможность доступа к таблице из другой базы данных, находящейся на том же хосте
- Чувствительность к регистру
- Поддержка даты и времени
- Аутентификация
- Разграничение доступа к столбцам
- Поддержка DISTINCT ON

# Сравнение СУБД для веб-проектов

- Поддержка WITH ROLLUP
- Поддержка WITH CUBE
- Поддержка функций OVER..PARTITION BY
- Поддержка рекурсивных запросов
- Поддержка COUNT(DISTINCT), AGGREGATE(DISTINCT)
- Поддержка OGC
- Поддержка схем
- Поддержка CROSS APPLY
- Поддержка LIMIT .. OFFSET
- Наличие Advanced Database Tuning Wizard
- Наличие Maintenance Plan Wizard
- Наличие Pluggable Storage Engine
- Поддержка связанных подзапросов
- Производительность планировщика запросов для сложных запросов
- Наличие текстового процессора
- Поддержка последовательностей и автоматической нумерации
- Возможность откатить CREATE, ALTER

# Результат сравнения

Оценка производится по критерию — за полную поддержку 2 балла, за частичную — 1 балл, за никакую — 0 баллов.

Результаты оценки:

- MS SQL — 87 баллов
- Mysql — 64 балла
- PostgreSQL — 91 балл.



# Кто выбрал PostgreSQL?

PostgreSQL используют в качестве СУБД такие проекты и компании как:

- Yahoo!
- MySpace
- OpenStreetMap
- Sony Online Entertainment
- BASF
- hi5.com
- Skype
- Sun xVM
- Evergreen

# Выводы

- Изучены основные термины СУБД
- Изучена архитектура СУБД
- Изучены принципы организации СУБД
- Изучен подход к выбору СУБД

# Вопросы для самопроверки

- Обязательно будут после лабораторной работы!!!

# Источники дополнительных сведений

- Средства доступа к базам данных в Internet и свободно доступная СУБД POSTGRES95 Булах Е.В., Центр Информационных Технологий
- <http://www.intersystems.ru>
- <http://www.citforum.ru>
- <http://www.sql.ru>
- <http://www.osp.ru>
- <http://www.tspu.tula.ru>
- <http://blog.groovytel.ru/2009/11/20/сравнение-субд-для-веб-проектов/>
- <http://www.masters.donntu.edu.ua>