

**Российская Академия Наук**  
**Институт машиноведения им. А.А. Благонравова**  
**Центр дистанционного обучения**

**А.Ю. Топорец**

**ПРОЕКТИРОВАНИЕ КОРПОРАТИВНЫХ  
ИНФОРМАЦИОННЫХ СИСТЕМ КЛАССА ERP ДЛЯ  
УПРАВЛЕНИЯ СЕТЬЮ ТЕРРИТОРИАЛЬНО  
РАСПРЕДЕЛЕННЫХ ФИЛИАЛОВ**

**Москва 2002**

УДК 519.68

Проектирование корпоративных систем класса ERP для управления сетью территориально распределенных филиалов

А.Ю. Топорец

Московский Государственный Институт Электроники и Математики

Рассмотрена проблема проектирования корпоративных систем класса ERP для управления сетью территориально распределенных филиалов. Описаны процессы управления и планирования ERP системах, а также методы организации и хранения информации в распределенных корпоративных системах. Сформулированы основные критерии выбора базовых программных продуктов. Предложена структура распределенной базы данных КИС управления сетью филиалов и поставлена задача расчета параметров функционирования, как задача минимизации суммарной стоимости эксплуатации системы. Приведен пример проектирования информационной системы для управления сетью предприятий розничной торговли.

Библиография двенадцать названий

# ОГЛАВЛЕНИЕ

<b>КОНЦЕПЦИЯ ERP.....</b>	<b>5</b>
СТАНДАРТЫ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ .....	5
УПРАВЛЕНИЕ И ПЛАНИРОВАНИЕ В ERP СИСТЕМАХ .....	8
<b>ВЫБОР БАЗОВЫХ ПРОГРАММНЫХ ПРОДУКТОВ.....</b>	<b>16</b>
ВЫБОР СУБД .....	16
ВЫБОР ОПЕРАЦИОННОЙ СИСТЕМЫ И ПРОТОКОЛА СЕТИ .....	37
<b>ОРГАНИЗАЦИЯ И ХРАНЕНИЕ ИНФОРМАЦИИ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ .....</b>	<b>40</b>
МОДЕЛИ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ.....	41
РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ .....	46
ВЫБОР СТРУКТУРЫ РАБД ДЛЯ КИС УПРАВЛЕНИЯ СЕТЬЮ ФИЛИАЛОВ .....	47
РАСЧЕТ ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ФУНКЦИОНИРОВАНИЯ РАБД .....	52
<b>ПРИМЕР ПРОЕКТИРОВАНИЯ КИС КЛАССА ERP ДЛЯ УПРАВЛЕНИЯ СЕТЬЮ ПРЕДПРИЯТИЙ РОЗНИЧНОЙ ТОРГОВЛИ.....</b>	<b>55</b>
АНАЛИЗ ПРОБЛЕМЫ.....	56
ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ИС ТОРГОВОГО ПРЕДПРИЯТИЯ .....	58
РАСЧЕТ ПАРАМЕТРОВ ФУНКЦИОНИРОВАНИЯ РАБД .....	63

## ВВЕДЕНИЕ

В современных условиях наблюдается рост рынка информационных систем автоматизации предприятия. При переходе к рыночной экономике для многих предприятий стал актуальным вопрос комплексной автоматизации. Теперь споры и дискуссии ведутся не о необходимости автоматизации, а на тему способов ее осуществления.

Любая из систем – лишь механизм для повышения эффективности управления, принятия правильных стратегических и тактических решений на основе своевременной и достоверной информации, выдаваемой компьютером.

Прежде всего, предприятие должно определить какие функциональные области и какие типы производства должна охватывать информационная система. Далее производится выбор базовых программных продуктов. Основой корпоративной информационной системы является система управление базой данных. Важно выбрать СУБД, которая не только в полной мере удовлетворяет как текущим, так и будущим потребностям предприятия. Кроме этого, большое значение имеет выбор технологии организации и хранения информации.

Каждый проект в области автоматизации должен рассматриваться предприятием как стратегическая инвестиция средств, которая должна окупиться за счет улучшения управленческих процессов, повышения эффективности производства, сокращения издержек. В выборе правильного решения должно быть, в первую очередь, заинтересовано руководство предприятия. Данный проект должен ставиться на один уровень с приобретением, например, новой производственной линии или строительством цеха.

## **КОНЦЕПЦИЯ ERP**

Сейчас практически все современные западные производственные и основные системы управления производством базируются на стандартах, которые вырабатываются американской общественной организацией APICS, объединяющей производителей, консультантов в области управления производством, разработчиков программного обеспечения.

### **СТАНДАРТЫ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ**

Для определения качества корпоративных информационных систем существуют общепринятые стандарты MRP, MRP II, ERP и CSRP, определенные Американским обществом управления производствами и запасами (APICS).

MRP (Material Requirements Planning) – планирование потребностей предприятия. Поддержания в достатке материальных ресурсов.

MRP II (Manufacturing Resource Planning) – планирование производственных ресурсов. Управление производством с элементами финансового планирования, а также обратной связи и моделирования.

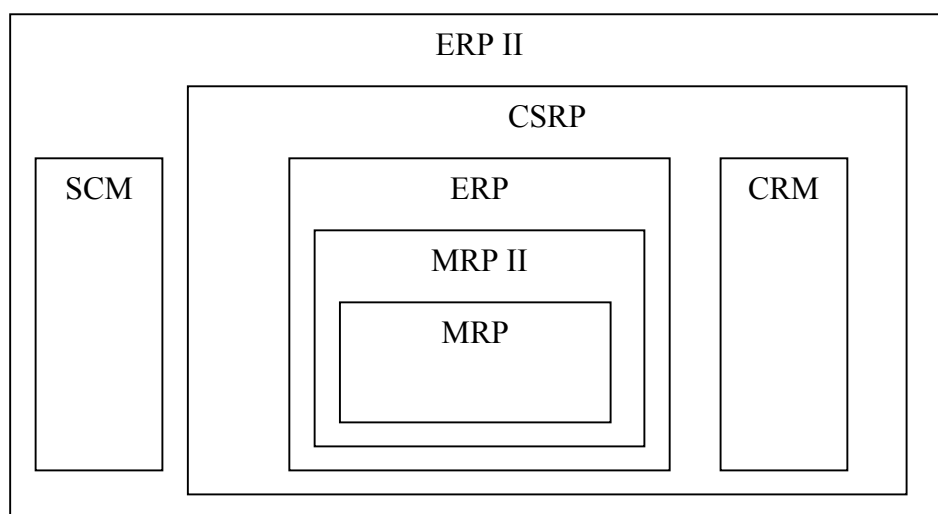
ERP (Enterprise Resource Planning) – планирование ресурсов предприятия. Управление всеми ресурсами предприятия, включая финансовые, маркетинговые и человеческие. Моделирование возможностей, поддержка принятия решений, аналитические инструменты. ERP также позволяет выходить за границы предприятия, например, управлять цепочками поставщиков.

CSRP (Customer Synchronized Resource Planning) – планирование ресурсов, синхронизированное с покупателем. Расширение функциональности ERP-системы на сферу взаимодействия предприятия с его заказчиками. Корпоративные ресурсы, охватываемые CSRP-системой, обслуживают такие этапы производственной деятельности, как проектирование будущего изделия с

учетом специфических требований заказчика, гарантийное и сервисное обслуживание.

Концепция ERP II (Enterprise Resource and Relationship Processing) – управление внутренними ресурсами и внешними связями предприятия.

B2C (Business to Customer) и B2B (Business to Business) – обозначения широких классов программных продуктов, обслуживающих взаимоотношения предприятий с покупателями (B2C) и между собой (B2B).



**Рисунок 1 Структура ERP**

CRM (Customer Relations Management) – управление отношениями с заказчиками.

SCM (Supply Chain Management) – управление отношениями с поставщиками.

По стандарту APICS, MRP II включает следующие функции:

- Sales and Operation Planning – Планирование продаж и производства
- Demand Management – Управление спросом
- Master Production Scheduling – Составление плана производства
- Material Requirement Planning – Планирование потребностей в сырье и материалах

- Bill of Materials – Спецификации продукции
- Inventory Transaction Subsystem – Складская подсистема
- Scheduled Receipts Subsystem – Отгрузка готовой продукции
- Shop Flow Control – Управление производством цеховом уровне
- Capacity Requirement Planning – Планирование производственных мощностей
- Input/output control – Контроль входа/выхода
- Purchasing – Материально-техническое снабжение
- Distribution Resource Planning – Планирование запасов сбытовой сети
- Tooling Planning and Control – Планирование и управление инструментальными средствами
- Financial Planning – Финансовое планирование
- Simulation – Моделирование
- Performance Measurement – Оценка результатов деятельности

Класс ERP, в отличие от MRP и MRP II, для которых имеются строгие определения и формализованные перечни требований, описан только на уровне концепции. Поэтому утверждения о том, что такая-то система относится к классу ERP, строго говоря, является рекламным утверждением, или, в лучшем случае, экспертным заключением.

Различие между концепциями MRP II и ERP заключается в том, что первая ориентирована на производство, а вторая – на бизнес. Например, такие вещи, как условия кредитования заказчика по отгрузке готовой продукции, попадают в поле зрения ERP, но не MRP II.

Инструментарий OLAP, средства поддержки принятия решений – обязательные принадлежности ERP, но не MRP/MRP II систем. То же самое можно сказать про управление процессом планирования, к которому относятся моделирование и сравнение различных вариантов производственных планов по

критериям ресурсной и финансовой реализуемости, а также и процесс выработки решений при отклонениях, когда система выдаёт перечень возможных вариантов изменения плана: снять заказ, изменить его, перераспределить ресурсы - и, таким образом, формирует пространство для выбора управленческого решения.

## УПРАВЛЕНИЕ И ПЛАНИРОВАНИЕ В ERP СИСТЕМАХ

Корпоративная информационная система призвана повысить потенциал предприятия. Как уже было сказано ранее, оправдано применение КИС класса ERP, которая была предложена компанией Gartner Group в начале 90-х годов и на сегодняшний день уже доказала свою эффективность.



**Рисунок 2 Повышение эффективности работы предприятия**

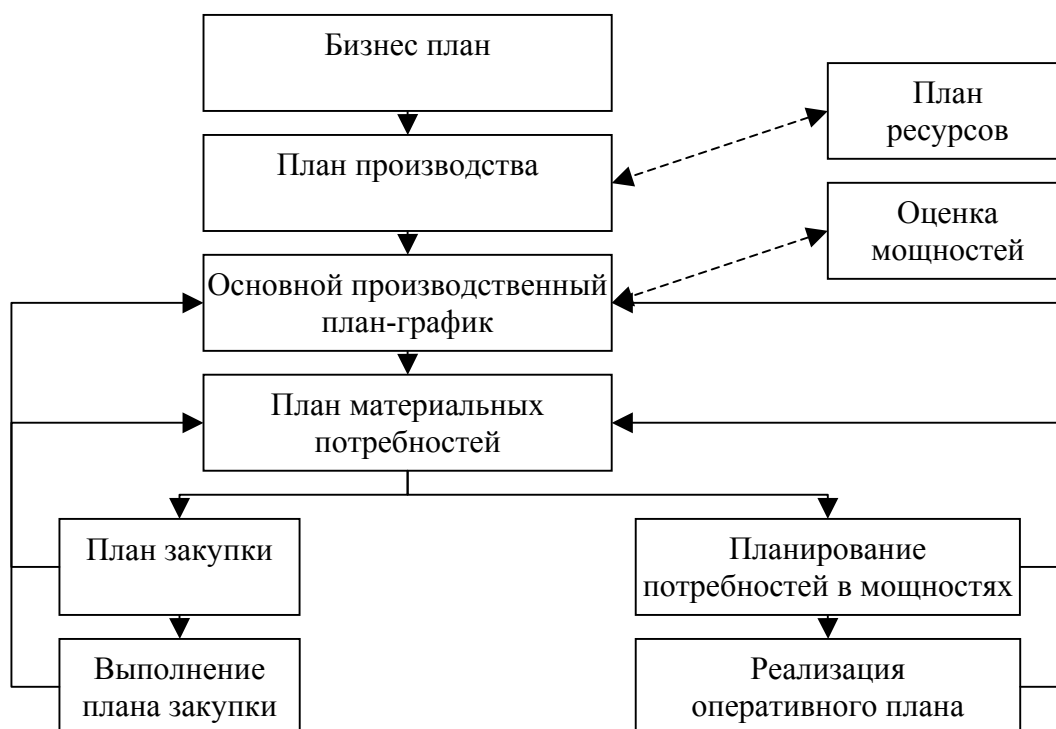
Система ERP представляет собой набор интегрированных приложений, которые позволяют создать единую среду для автоматизации планирования, учета, контроля и анализа всех основных бизнес операций в масштабе предприятия.

Основа управления систем ERP – планирование. Соответственно, все модули решают задачи трех основных групп:



- Учет и контроль
- Анализ и планирование
- Анализ и коррекция плана

Существуют несколько видов планов, а именно бизнес-план, маркетинговый план, производственный план, основной производственный план-график, план закупки материалов, оперативный план производства.



**Рисунок 3 Общая модель системы производства**

Бизнес-план представляет собой долгосрочный, стратегический план деятельности всего предприятия и его разработка является прерогативой высшего руководства предприятия. Пункты плана могут включать развитие новых рынков сбыта, развитие новых семейств производимой продукции, комбинаций бизнеса, реструктуризации бизнеса и т. д. Форма плана, как правило, соответствует основным финансовым формам: баланс предприятия, отчет о прибылях и убытках, отчет о движении денежных средств и выражается в финансовых показателях.

Маркетинговый план (не отображен на рисунке), является результатом бизнес-плана и должен соответствовать показателям бизнес-плана и плана производства. Маркетинговый план может включать в себя методы и средства продвижения продукции на рынке, организации продаж и распределения.

Производственный план утверждается на основе бизнес-плана и может охватывать период до нескольких лет. Как правило, производственный план составляется в объемных показателях (тонны, штуки) производства семейств продукции в зависимости от особенностей бизнеса предприятия.

Производственный план отражает общие производственные показатели и объединяет плановые данные, касающиеся оборудования, основных инструментов, человеческих ресурсов, материалов и комплектующих. Иногда перечисленные данные составляют основу отдельного так называемого плана ресурсов предприятия, на который ссылается производственный план. План производства и план ресурсов в свою очередь инициируют разработку финансового плана или бюджетов, основанных на основных нормативных показателях.

Основной производственный план-график (Master Production Plan – MPS) составляется на основе производственного плана, как правило, на три, шесть месяцев или на год в зависимости от природы производственных процессов. MPS выражается в объемных показателях реально выпускаемой продукции и разрабатывается в соответствии с маркетинговым планом, предсказывающим, что должно быть произведено и продано либо на основе прогноза, либо на основе подтвержденных клиентских заказов, либо их комбинации. Как правило, в то же время осуществляется предварительная оценка производственных мощностей (RCCP – Rough Cut Capacity Planning) для определения возможности выполнения заданных показателей производства с использованием доступных в настоящее время оборудования и трудовых ресурсов.

Планирование потребностей в материалах (Material Requirement Planning – MRP). Процедура MRP преобразует график производства конечной продукции (MRS) в график заказов на закупку комплектующих и материалов и заказов на производство компонент с учетом требований по срокам и технологии производства конечной продукции. Процедура данного преобразования строится на основе состава производимого изделия, ведомости материалов, существующих и заказанных запасов, особенностей планирования и т. д.

Планирование потребностей в производственных мощностях (Capacity Requirement Planning – CRP). Совместно с другими выходными данными процедура планирования материальных ресурсов формирует предварительный график выпуска производственных заказов. Процедура планирования потребностей в производственных мощностях осуществляет анализ упомянутых производственных заказов с учетом существующего незавершенного производства для оценки возможности использования производственных ресурсов в нужные периоды для реализации показателей основного производственного плана-графика. В случае обнаружения недостатка или несбалансированной загрузки оборудования необходимо принимать решение либо по ресурсам, либо по планируемыми производственным заказам, либо по изменению MPS.

План закупки. Результатом процесса MRP планирования является график приобретения материалов и комплектующих, который используется службами закупки для организации процесса закупки. Данные MRP планирования являются основой для организации выбора поставщика, согласования договоров и, в конечном счете, определяют возможность удовлетворения запросов производства. Если возникают проблемы с приобретением всей необходимой для производства номенклатуры материалов и комплектующих,

принимается решение о ревизии MRP запланированных производственных заказов или изменении основного производственного плана-графика.

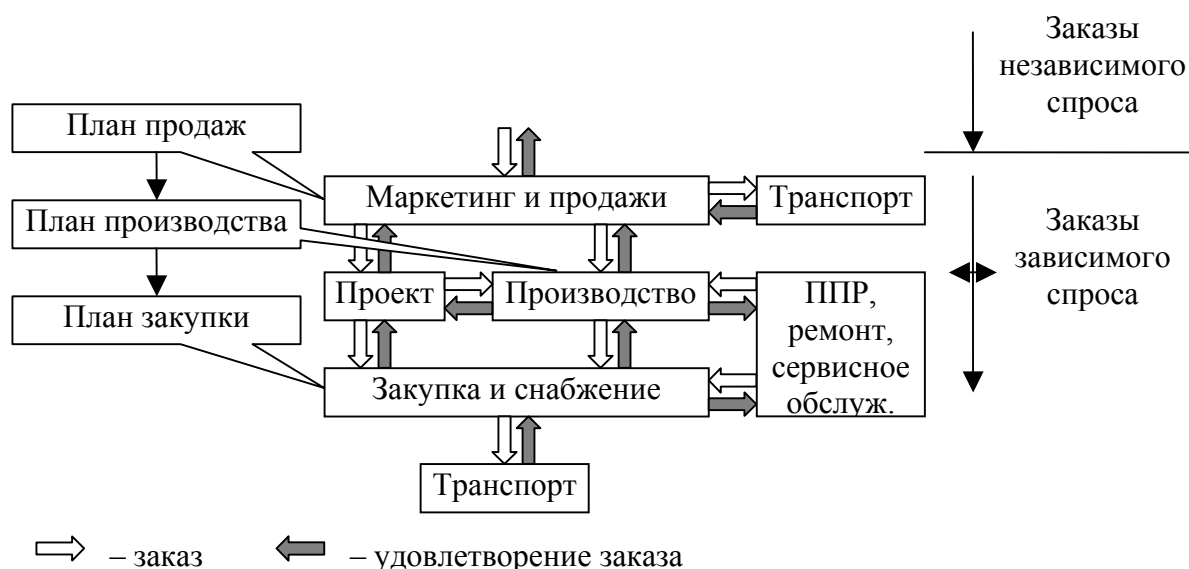
Выполнение плана закупки. Процедура реализации MRP планирования в части закупки материалов и комплектующих требует выполнения соответствующих затрат после их утверждения соответствующими полномочными лицами.

Оперативный план производства. Формируется на основе MRP для производства необходимого для выпуска готовой продукции запаса узлов, компонент на собственных мощностях.

Выполнение оперативного плана производства. Процедура реализации MRP плана в части производства узлов и компонент основывается на MRP графике выпуска производственных заказов с определением, каким образом производственные заказы выпускаются, отслеживаются и контролируются. Данная операция часто называется "Диспетчеризация производства" (PAC – Production Activity Control). Процесс диспетчеризации производства по обратной связи может оказывать влияние на перепланирование деятельности на более высоких уровнях.

Таким образом, можно заметить, что общая модель системы производства описывается циклическими процессами на нескольких уровнях, в особенности на уровне реализации планов закупки и собственного производства, оказывающих влияние на более высокие уровни в связи с реальными возможностями и практикой по процессам приобретения и производства. Такого рода обратная связь часто трактуется как замкнутая петля в системной модели. В литературе по управлению производством и запасами часто используется термин MRP системы замкнутого цикла.

Базовым объектом управления является заказ. Он обеспечивает организацию иерархической структуры систем планирования и учета, реализованных в ИС ERP.



**Рисунок 4 Стандартная схема потока заказов**

Заказы, как правило, рассматриваются с двух точек зрения: заказы независимого спроса, заказы зависимого спроса.

Заказы независимого (рыночного) спроса поступают извне и, в свою очередь, имеют следующие основные категории: прогнозные (предполагаемые на основе анализа потребностей рынка), подтвержденные (существующий портфель заказов).

Заказы независимого спроса определяют показатели программы сбыта готовой продукции или ее компонент.

Программа продаж является основой для формирования плана производства и, как следствие, план закупок, реализация которых обеспечивается заказами зависимого спроса: заказ на производство, заказ на закупку, заказ на внутреннее перемещение (склад – склад, склад – производство/проект/профилактика или ремонт), заказ на транспортировку и другие.

В пределах независимого спроса на производимую продукцию может быть несколько разновидностей заказов, порождающих особенности планирования производства. Основные разновидности заказов определяются

так называемой точкой привязки заказа клиента (ТПЗК) или Customer Order Decoupling Point (CODP).

Точка привязки заказа клиента – наиболее дальняя точка в производственном процессе, в которой планирование материалов, компонент, деталей и узлов конечного продукта определяется заказом клиента. Другими словами, ТПЗК есть глубина проникновения заказа клиента в систему планирования.

Производство проектного типа (опытное производство) практически полностью определяется заказом. Таким образом, стадии планирования производственных заказов и заказов на закупку материалов и комплектующих предшествует стадия заказов на разработку конструкторской и технологической документации и ее непосредственная подготовка.

В ряде случаев предприятие может выпускать повторяющуюся продукцию, но по заказам независимого спроса возникает потребность модификации некоторого стандартного изделия. ТПЗК в данной ситуации предполагает усложнение процедуры планирования выпуска конечного изделия под заказ с учетом использования большинства стандартных компонент.

И, наконец, для продукции, потребительские свойства которой удовлетворяют заказы независимого спроса, ТПЗК не оказывает влияние на особенности организации процесса производства, а определяет объемные характеристики планирования производства.

В случае изготовления на заказ (МТО – Make to Order), производимые изделия выполняются на основе заказа клиента, планирование потребности в материалах и комплектующих, а также конструктивные и технологические особенности производства определяются заказной спецификацией и заказной технологией. Часто используется термин ЕТО (Engineering To Order – Разработка на заказ). План продаж формируется на основании портфеля заказов.

В случае сборки на заказ, производимые изделия выполняются на заказ в основном из стандартных компонент, отличия изделий одного типа друг от друга формирует в результате некоторый «модельный ряд». План продаж в основном формируется на основании портфеля заказов и может дополняться прогнозными данными.

В случае производства на склад, имеется ограниченная номенклатура продукции. Готовая продукция производится на склад для последующей доставки. План продаж в основном формируется на базе прогноза спроса плюс портфель заказов.

Несмотря на то, что концепция ERP разрабатывалась, в основном, для материального производства, возможно расширение сферы примени ее методов.

В качестве примера можно рассмотреть через призму ERP предприятие розничной торговли. Предприятие розничной торговли всегда работает «на заказ», где в качестве заказчика выступает весь рынок. В сущности, торговля это производство и реализация так называемых торговых услуг. Основным сырьем для производства торговой услуги являются товары. При этом товары аналоги являются различными видами одного и того же сырья. Сопутствующие товары можно считать второстепенным сырьем. Товары также выступают в качестве орудий производства. Длительность производственного цикла сводится к среднему времени пребывания покупателя в магазине. В процессе производства происходит формирование торговой услуги за счет использования ряда технологических операций, часть состава которых определяет заказчик, – подбор, упаковка, доставка и т. д. Некоторые операции, например, самообслуживание, могут выполняться автоматически, но они также предполагают оплату, так как, в частности, открытый доступ к товару предполагает дополнительные системы контроля и слежения, хотя бы выделение специального сотрудника. К таким операциям относится также

предложение фонового ассортимента, выкладка, оформление прилавков и витрин и т. д. Поскольку основным ресурсом является время, основное внимание уделяется ускорению без потерь качества всех технологических операций, которые связаны со временем. В первую очередь это касается процедур, имеющих отношение к подбору нужных товаров и к их доставке.

## **ВЫБОР БАЗОВЫХ ПРОГРАММНЫХ ПРОДУКТОВ**

Выбор базовых программных продуктов представляет собой сложную многопараметрическую задачу и является одним из важных этапов при разработке корпоративных информационных систем. Этот выбор должен удовлетворять как текущим, так и будущим потребностям предприятия, при этом следует учитывать финансовые затраты не только на приобретение, но и на разработку системы, сопровождение, а также обучение персонала.

### **ВЫБОР СУБД**

Основой корпоративной информационной системы является система управления базой данных. Важно выбрать такую СУБД, которая не только в полной мере удовлетворяет текущим потребностям, но и имеет необходимый «запас прочности» для дальнейшего расширения и интеграции.

#### **Критерии выбора СУБД**

Наиболее простой подход при выборе СУБД основан на оценке того, в какой мере существующие системы удовлетворяют основным требованиям проекта информационной системы. Более сложным и дорогостоящим вариантом является создание испытательного проекта на основе нескольких СУБД и последующий выбор наиболее подходящей. Но и в этом случае следует ограничивать круг возможных систем, опираясь на некие критерии отбора. Перечень требований к СУБД может меняться в зависимости от поставленных целей. Тем не менее, можно выделить несколько групп критериев:



производительность, надежность, особенности разработки приложений, контроль работы системы, требования к рабочей среде, особенности архитектуры и функциональные возможности, моделирование данных, смешанные критерии.

### ***Моделирование данных***

Моделирование данных включает в себя следующие критерии:

*Используемая модель данных.* Существуют несколько моделей данных: иерархическая, сетевая, реляционная, объектно-реляционная, объектная. Выбор модели осуществляется исходя из основных потребностей информационной системы на начальном этапе проектирования.

*Поддерживаемые типы данных.* Некоторые современные системы помимо общепринятых базовых типов данных (integer, character) поддерживают дополнительные типы данных, а также возможности расширения типов данных.

*Реализация языка запросов.* Практически все реляционные СУБД совместимы с той или иной спецификацией стандартного языка доступа к данным SQL (SQL-89, SQL-92). Кроме того, во многих системах реализованы некоторые расширения данного стандарта (например, в свое время таким расширением были триггеры).

*Средства поиска.* Некоторые современные системы имеют встроенные дополнительные средства контекстного поиска.

*Триггеры, хранимые процедуры, ограничения и утверждения как конструкции активных сред.* Традиционные базы данных являются пассивными, т.е. изменения и выборка данных осуществляются только под влиянием внешнего источника. Однако в последнее время важную роль приобретает концепция активных баз данных, когда в базу данных закладываются возможности управления данными. В создании активных сред помогают несколько основных конструкций базы данных: ограничения, утверждения, хранимые процедуры и триггеры. Триггер это программа базы

данных автоматически выполняемая при некоторых событиях (создание записи, изменение записи, удаление записи и др.). Хранимая процедура это программа, которая относится (хранится) к базе данных, а не к внешней программной системе. Ограничения и утверждения это относительно простые конструкции, имеющие вид от спецификации связей первичного и внешнего ключей, используемых в ограничениях целостности по ссылке, до SQL-подобных ограничений, используемых для проверки принадлежности значений заданному диапазону или списку значений.

### ***Особенности архитектуры и функциональные возможности***

*Сетевые возможности.* Практически все современные СУБД позволяют использовать широкий диапазон сетевых протоколов и служб для работы и администрирования.

*Масштабируемость.* Как правило, система должна предусматривать рост информационной системы. Различают увеличение объема хранимых данных, числа пользователей и объема обрабатываемой информации.

*Распределенность.* Обычно информационная система создается с целью объединения информации предприятия, но наиболее простая реализация в виде централизации данных на одном сервере не всегда применима из соображений повышенных требований к отказоустойчивости или сильного территориального распределения подразделений. Механизмы создания распределенных баз данных в различных системах реализованы по-разному.

*Максимальный размер базы данных.* При выборе системы следует учитывать, сможет ли база данных вместить планируемые объемы информации.

### ***Контроль работы системы***

*Средства мониторинга.* Современная система должна обладать средствами мониторинга основных параметров базы данных для администрирования и принятия решений по оптимизации комплекса.

*Автоматическая настройка.* Некоторые СУБД включают возможность автоматически подстраиваться для достижения максимальной эффективности.

### ***Требования к рабочей среде***

*Минимальные требования к оборудованию.* Поскольку большинство современных информационных систем построены на основе архитектуры клиент-сервер, различают требования к оборудованию клиентской и серверной части.

*Поддерживаемые аппаратные платформы и операционные системы.* Современные СУБД могут работать на различных аппаратных платформах под управлением нескольких операционных систем. Это дает возможность выбирать операционные системы и аппаратную конфигурацию для достижения максимальной надежности и производительности в условиях ограниченного финансирования.

### ***Производительность***

*Рейтинг TPC.* Существуют несколько тестовых рейтингов. Одним из самых популярных и объективных является TPC. Он представляет собой комплексный анализ СУБД и аппаратуры. Существует два вида рейтинга TPC оценивающие производительность и отношение производительность к цене комплекса. Кроме этого рейтинг разбит на категории с точки зрения архитектуры серверной части (кластерные системы, не кластерные системы), производителя серверной части (Compaq, IBM, HP, Dell и т.п.), а также области применения (OLTP, электронная коммерция, поддержка принятия решений, бизнес отчетность и т.п.)

*Возможности встроенной оптимизации запросов.* Современные СУБД должна содержать оптимизатор запросов, который в результате семантический преобразований с учетом статистических данных вырабатывает оптимальный план выполнения запроса.

*Возможности параллельной архитектуры.* Для повышения производительности во многих современных СУБД применяется параллельная обработка. Например, возможна поддержка технологий параллельной обработки последовательности запросов на нескольких процессорах или использование нескольких компьютеров объединенных в параллельный сервер базы данных.

*Возможности перераспределения обработки между сервером и клиентом.* Некоторые СУБД позволяют частично или полностью переносить обработку данных с клиента на сервер. В результате становится возможным по выбору строить конфигурации с «толстым» или с «тонким» клиентом. В некоторых СУБД возможно построение приложений, которые в зависимости от ситуации сами решают, какая часть обработки должна быть перенесена на сервер или другого клиента.

### ***Надежность***

*Модель транзакций.* Существуют многочисленные модели транзакций – от простейших, например плоских, до более изощренных, таких как вложенные или многозвенные. Большое значение имеет правильный выбор одной из сложных моделей транзакций с учетом специфики деятельности конкретной организации. Например, вложенные транзакции предоставляют исключительно гибкие возможности управления субтранзакциями, но они сложны в реализации и не во всех случаях эти возможности необходимы. Для построения распределенных архитектур СУБД должна поддерживать распределенные транзакции.

*Защита информации от несанкционированного доступа.* Любая современная информационная система должна иметь возможность разграничения доступа к информации. Существуют различные уровни защиты информации. В большинстве случаев не достаточно простой идентификации пользователей при входе в систему, поэтому СУБД должна поддерживать

средства, с помощью которых для каждого пользователя определяются полномочия доступа к данным и модификации данных.

*Резервное копирование.* В некоторых критических ситуациях восстановление потерянных данных, возможно только с использованием резервной копии. СУБД должна позволять варьировать планы резервного копирования в зависимости от ограничений на время восстановления, стоимость хранения и возможную потерю части информации (например, за последний день). Некоторые СУБД позволяют выполнять резервное копирование без остановки системы. Кроме этого существуют технологии снижающие стоимость хранения. В качестве примера можно привести инкрементальное резервное копирование, при котором сохраняются только те блоки базы данных, которые изменились с момента последнего резервного копирования.

*Восстановление после сбоев.* Программные или аппаратные сбои могут привести к нарушению целостности информации. В этом случае дальнейшее функционирование системы зависит от реализации механизмов восстановления. Например, ведение журнала незавершенных транзакций позволяет автоматически восстановить целостность базы данных после сбоя возникшего при выключении электричества.

### ***Особенности разработки приложений***

*Спектр поддерживаемых языков программирования.* СУБД может поддерживать несколько языков программирования. Наличие широкого набора языков программирования повышает доступности системы для разработчиков и позволяет создавать оптимальные по производительности и функциональности приложения.

*Поколение поддерживаемых языков программирования.* Функциональность приложения, а также его производительность и время разработки, существенно зависят от поколения используемого языка

программирования. Применение проблемно-ориентированного языка четвертого поколения (4GL), которые оперирует конкретными понятиями предметной области, существенно повышает функциональность приложения. Один оператор 4GL может соответствовать десяткам операторов 3GL. 4GL содержит средства визуальной разработки, что позволяет вести разработку в стиле RAD (Rapid Application Development – быстрая разработка приложений). Если же прикладная область ориентирована на Internet, то требуется применение языков 3GL, основанных на языке Java.

*Средства разработки Web-приложений.* При проектировании Internet-ориентированного приложения, желательно, чтобы СУБД содержала специальные средства разработки Web-приложений.

*Собственные средства разработки приложений.* Многие поставщики создали специальные средства разработки приложений для своих систем, которые позволяют наилучшим образом использовать возможности СУБД. Это могут быть средства создания отчетов, средства разработки Web-приложений, язык 4GL.

*Многоязыковая поддержка.* Поддержка большого количества национальных языков расширяет область применения СУБД и построенных на ее основе приложений.

### ***Смешанные критерии***

*Поддержка национального языка.* Поддержка национального языка может существенно сказываться на стоимости разработки и сопровождения приложений.

*Качество и полнота документации.* К сожалению, не все системы имеют полную и подробную документацию.

*Стоимость системы.* При оценке стоимости системы должны учитываться и цена базовых продуктов, и стоимость их сопровождения. Кроме этого, следует учитывать стоимость разработки приложения для выбираемой

СУБД. На стоимость системы сильно влияет модель формирования стоимости базовых продуктов. Например, как правило, цена конкретного продукта зависит от числа пользователей. Поэтому, в некоторых случаях следует выбирать более дорогую СУБД, которая позволит снизить финансовые вложения при будущем расширении информационной системы.

### **Обзор наиболее известных СУБД**

На сегодняшний день известно более двух десятков серверных СУБД, однако наиболее популярными следует признать Oracle, Microsoft SQL Server, Informix, Sybase, DB2, Progress.

#### ***Oracle***

Oracle была первой коммерческой реляционной СУБД, которая поддерживала язык SQL. Ее первая версия появилась в 1979 году. Фактически все это время Oracle является бессменным лидером на рынке производителей коммерческих СУБД и второй (после Microsoft) по величине компании, производящей программное обеспечение.

Oracle была первой компанией, создавшей СУБД, которая использовала предоставляемые некоторыми серверными платформами средства параллельных вычислений. Oracle Parallel Server дает возможность нескольким процессорам обращаться к одной базе данных, что позволяет обеспечить высокую скорость обработки транзакций. Последние версии дают возможность осуществить декомпозицию операций с большими объемами данных с целью параллельного выполнения их на нескольких процессорах.

Существуют версии этой СУБД для различных серверных платформ: UNIX, VMS, Windows и др. СУБД Oracle поддерживает объектные расширения и соответствующие типы данных, такие как вложенные таблицы, массивы, объекты и др. Имеется возможность создания таблиц, содержащих агрегированные данные и возможность частичного их обновления при

изменении данных, на основании которых они вычислены. Поддерживаются Java, XML, C/C++, PL/SQL.

СУБД дает возможность использовать дополнительные средства обеспечения безопасности, например шифрование данных, поддержку SSL, роли уровня базы данных и уровня предприятия.

Oracle 8i существует в трех редакциях: Oracle 8i, Oracle 8i Enterprise Edition, Oracle 8i Personal Edition. Для создания многомерных хранилищ данных существует и отдельный продукт – Oracle Express OLAP.

Помимо различных версий сервера баз данных среди продуктов Oracle имеется также Designer/2000 – ориентированное на эту СУБД CASE-средство для анализа бизнес-процессов и проектирования данных, а также средства разработки клиентских приложений. Одно из них – Developer/2000. Приложения, созданные с помощью Developer/2000, могут выполняться на различных платформах. Язык PL/SQL, используемый в этом средстве разработки, является интерпретируемым и представляет собой тот же самый язык, что используется в Oracle для написания серверного кода. Это позволяет отлаживать серверный код с помощью Developer/2000.

Производя собственные средства разработки, Oracle предоставляет своим пользователям возможность создавать клиентские приложения с помощью других средств. Помимо стандартного клиентского API (Oracle Call Interface), клиентская часть Oracle содержит объектную модель (Oracle Objects for OLE), позволяющую использовать клиентскую часть Oracle в качестве набора COM-объектов для доступа к данным. Кроме этого клиентская часть Oracle может содержать ODBC-драйвер для доступа к данным этой СУБД.

Отметим, что и многие другие компании производят ODBC-драйверы и OLE DB-провайдеры для доступа к Oracle (в частности, Microsoft). Компании, производящие средства разработки, использующие собственные библиотеки доступа к данным (такие как Inprise или Gupta/Centura), также включают



библиотеки доступа к Oracle в состав наиболее дорогих версий своих продуктов.

Из готовых информационных систем на базе Oracle следует особо отметить одну из самых крупных систем управления предприятием SAP/R3. На Западе также нередко используются готовые решения от самой Oracle Corporation, объединенные под общим названием Oracle Applications, такие как Oracle Financials, Oracle Human Resources, Oracle Market Management, Oracle Project Systems и др.

### ***Microsoft SQL Server***

Первая версия Microsoft SQL Server, совместно разработанная в 1988 году компаниями Microsoft и Sybase, предназначалась для платформы OS/2. Последующие версии этого сервера баз данных предназначались для платформы Windows NT и со временем были тесно интегрированы с этой операционной системой. Позднее был осуществлен переход от Windows NT к Windows 2000. Для других платформ версии этого сервера не выпускались и не выпускаются.

СУБД отличается удобством пользовательского интерфейса утилит администрирования в сочетании с достаточно высокой производительностью и относительно невысокой стоимостью эксплуатации. Наибольший рост популярности этой СУБД пришелся на конец 90-х годов, когда были выпущены Microsoft SQL Server 6.0 (1995 год), обладавший централизованными функциями администрирования и встроенными возможностями репликации данных, Microsoft SQL Server 6.5 (1996 год) и Microsoft SQL Server 6.5 Enterprise Edition, поддерживающий параллельные вычисления в многопроцессорных системах.

СУБД обладает простыми утилитами администрирования. Сервисы преобразования данных (Data Transformation Services) облегчают перенос данных в SQL Server из других типов СУБД. Поддерживаются распределенные

запросы и транзакции, а также расширенные функции для создания Web-приложений.

Вместо Microsoft SQL Server в качестве встроенной СУБД для настольных приложений и приложений для небольших рабочих групп можно использовать Microsoft Data Engine (MSDE). Это настольный сервер баз данных совместим с Microsoft SQL Server и предназначен для использования в настольных системах или в сетевых приложениях с небольшим (до 2 Гбайт) объемом данных и малым количеством пользователей.

Клиентские приложения для Microsoft SQL Server и MSDE можно создавать как с помощью средств разработки Microsoft (Visual Basic, Visual C++, Access, Visual FoxPro), так и с помощью средств разработки других производителей. Для этой цели имеются ODBC-драйвер и OLE DB-провайдер, а также содержащий их набор библиотек Microsoft Data Access Components (MDAC), позволяющий использовать в средствах разработки объекты ActiveX Data Objects (ADO). MDAC является составной частью Windows 2000, а для пользователей других Windows-платформ доступен отдельно на Web-сайте Microsoft.

В отличие от Oracle, Microsoft не производит средств разработки, использующих тот же самый язык программирования, что и язык для создания кода триггеров и хранимых процедур, однако производит средства отладки серверного кода (например, SQL Server Debugger входит в состав Visual Basic и Visual C++).

### ***Sybase***

СУБД компании Sybase, называемая Adaptive Server Enterprise существует для Windows NT и некоторых версий UNIX (включая Linux). СУБД предназначен для обслуживания крупных предприятий. Сервер поддерживает упреждающее асинхронное чтение, что повышает скорость выполнения сложных запросов. Возможна распределенная обработка запросов как для

многопроцессорных, так и для кластерных систем. Расширенные хранимые процедуры позволяют осуществить легкий доступ к не-SQL функциям (Java, 3GL-системы, функции операционной системы и т.д.). СУБД обеспечивает интеграцию с популярными системами безопасности, такими как Kerberos.

Еще одна линия серверных продуктов Sybase ведет свое начало от сервера баз данных Watcom SQL Anywhere, отличавшегося компактностью и простотой администрирования. Этот сервер предназначен для обслуживания небольших рабочих групп, для применения в портативных компьютерах в качестве персонального сервера с периодической репликацией, а также в мобильных устройствах. Существуют версии этого сервера для Windows CE и версия UltraLite для разнообразных мобильных устройств.

Для управления распределенными транзакциями Sybase выпускает монитор транзакций Jaguar CTS.

Для создания многомерных хранилищ данных у Sybase существует еще один серверный продукт – Adaptive Server IQ, позволяющий создавать хранилища на основе данных не только из СУБД производства Sybase, но и из СУБД других производителей. Отметим также, что существует ряд продуктов под общим названием Sybase Industry Warehouse Studio, ориентированных на обслуживание конкретных предметных областей: торговли (Retail Warehouse Studio), здравоохранения (Healthcare Warehouse Studio), страхования (Life Insurance Warehouse Studio) и др.

Помимо серверных продуктов Sybase производит средства проектирования данных и генерации кода приложений. Для создания клиентских приложений созданы средства разработки PowerBuilder и PowerJ, а также продукт PowerSite, предназначенный для создания Web-приложений. CASE-средство DataArchitect поддерживает широкий спектр СУБД различных производителей, а генератор приложений AppModeler способен генерировать

код не только для PowerBuilder и Optima++, но и для Delphi, Visual Basic, Web-приложений с использованием ASP.

### *Informix*

Ведущим продуктом фирмы Informix является Informix Dynamic Server. Данный продукт поддерживает платформы UNIX и Microsoft Windows NT. Он обеспечивает эффективную работу как на одно-, так и на многопроцессорных системах, а также в кластерах. Сервер построен по архитектуре Dynamic Scalable Architecture (DSA), обеспечивающей мощные средства для параллельной обработки данных.

СУБД Informix управляет дисковым пространством как с помощью средств операционной системы (UNIX или Microsoft Windows NT), так и собственных функций, позволяющих обойти ограничения операционной системы и добиться более высокой производительности. Поддерживается фрагментация таблиц и индексов на нескольких дисках и управление разделением памяти для одновременного доступа к данным несколькими приложениями. Используются технологии распараллеливания запросов и зеркалирования данных.

Сервер поддерживает двухфазное завершение транзакций, гетерогенные транзакции (в этом случае в транзакциях может принимать участие и не-Informix сервер, доступный через Informix Enterprise Gateway).

Расширения функциональности сервера реализуются на базе DataBlade. Это коллекции объектов баз данных и подпрограмм на языке C, подключаемых к базе данных. Для разработки DataBlades необходимо использовать DataBlade Developer's Kit. Фирма Informix и целый ряд независимых производителей выпускают модули DataBlade, такие, например, как Excalibur Text DataBlade Module, Informix Geodetic DataBlade Module, Informix TimeSeries DataBlade Module, Excalibur Image DataBlade Module, Informix Web DataBlade Module и ряд других.

Входящие в состав Informix Dynamic Server клиентские утилиты предназначены для подключения к серверу и обработки информации (DB-Access) и для выполнения функций администрирования (DB/Cockpit).

Клиентские приложения могут создаваться с использованием языков Informix ESQL (средство для разработки на языке C, позволяющее включать в приложения запросы к данным на языке SQL), а также C, C++, Java, Visual Basic и Delphi. Помимо этого существуют собственные средства разработки – Informix-4GL и Informix Client Software Developer's Kit.

Фирма Informix выпускает Informix ODBC Driver, OLE DB Provider для Informix Dynamic Server и Informix JDBC Driver.

В состав продукта входят собственно сервер, а также Informix Connect, DataBlade Developer's Kit и Informix Server Administrator. Для генерации отчетов предлагается визуальное средство Informix-ViewPoint.

Говоря о сервере фирмы Informix, следует упомянуть и поддержку OLAP. Продукт под названием Informix MetaCube поставляется как часть Informix Decision Frontier – комплексного решения для создания хранилищ данных.

Среди других продуктов фирмы Informix следует отметить i.Reach, i.Sell и Informix Internet Foundation.2000. Informix Internet Foundation.2000 представляет собой специально разработанный для Internet вариант Informix Dynamic Server. i.Reach это корпоративный репозиторий для хранения данных различного типа, интеллектуального управления информацией и извлечения данных. Основное назначение данного продукта – поддержка включения в содержимое корпоративных сайтов электронных документов и их последующее обслуживание. Продукт i.Sell это комплексное решение для электронной коммерции на базе Informix Dynamic Server.

## ***DB2***

Семейство серверных СУБД фирмы IBM, известное под названием DB2 Universal Database, представляет собой стратегию IBM по объединению

продуктов DB2 для различных платформ в единую линию. Впервые появившееся в 1996 году семейство DB2 Universal Database объединяло в себе функциональные возможности таких продуктов фирмы, как DB2 Common Server, DB2 Parallel Edition (DB2 PE), Net.Data, Data Propagator и технологии DataHub. Семейство предназначалось для платформ UNIX, OS/2 и Microsoft Windows NT.

Отметим, что при переносе DB2 на не-IBM-платформы фирма старается максимально использовать уникальные функциональные возможности конкретной платформы. Например, в DB2 for Windows 2000 для обеспечения безопасности используется Windows NT LAN Manager, полностью поддерживаются Windows Performance Monitor, Systems Management Server, интеграция с Active Directory для каталогизации баз данных, а также такие интерфейсы доступа к данным, как ODBC, ADO и OLE DB. Помимо этого, DB2 for Windows 2000 поддерживает Microsoft Transaction Services (MTS) в качестве координатора при создании приложений, использующих распределенные транзакции.

Для разработчиков, использующих Microsoft Visual Studio, становятся доступными дополнительные модули, например Stored ProcedureBuilder, включаемый непосредственно в среду Visual Studio. IBM также предлагает собственные средства разработки, например IBM VisualAge for Java. СУБД поддерживает создание хранимых процедур на языке Java (Java Stored Procedure Builder).

IBM предлагает бесплатное средство для миграции данных из Microsoft Access в DB2, а также средства для миграции данных из Oracle, Microsoft SQL, Sybase и Informix.

К основным характеристикам СУБД можно отнести поддержку реляционных и комплексных данных через объектные расширения, возможность работы на мультипроцессорных платформах, поддержку

кластеров, 64-битную архитектуру памяти, распараллеливание запросов, возможность создания Web-приложений (поддерживаются такие технологии, как Java, JDBC, SQLJ, XML) и наличие средств для гетерогенного администрирования и обработки данных.

Семейство DB2 функционирует на системах AS/400 и RISC System/6000, мэйнфреймах IBM, машинах от Hewlett-Packard и Sun Microsystems и под управлением таких операционных системах, как Windows NT, Windows 95/98, OS/2, AIX, HP-UX, SCO UnixWare, Linux, NUMA-Q и Sun Solaris. Поддерживаются портативные устройства под управлением Windows CE и Palm OS.

Кроме этого поставляется средство для онлайн-аналитической обработки данных и реализации хранилищ данных – DB2 OLAP Server, интегрирующее ядро Hyperion Essbase с семейством DB2 Universal Database. Продукт работает с Hyperion Integration Server (Hyperion), Hyperion Wired for OLAP (Hyperion), Brio.Insight (Brio Technology), BUSINESSOBJECTS (Business Objects), PowerPlay (Cognos), Lotus 1-2-3 (Lotus), Excel, Internet Explorer, Visual Basic (Microsoft) и Crystal Info (Seagate). Продукт DB2 Connect является средством для управления соединениями различных клиентов с DB2 на AS/400.

Для создания и тестирования приложений в архитектуре клиент-сервер имеется средство DB2 Universal Developer's Edition. Продукт DB2 DataJoiner позволяет получать доступ к данным, расположенным на различных платформах как к единому образу данных.

Кроме этого компанией IBM поставляется DB2 продукт Query Patroller. Это набор средств для создания запросов и управления ресурсами для систем принятия решений. DB2 Query Patroller получает ODBC-запросы от клиента, анализирует их и динамически распределяет запросы по различным узлам DB2 UDB Enterprise – Extended Edition. Приложение DB2 Net.Data позволяет Web-

разработчикам создавать динамические Internet-приложения, используя Web Macros.

Предлагаются средства для внедрения масштабируемых мобильных решений и для управления удаленными пользователями. СУБД поддерживает функции репликации, централизованное администрирование и средства управления через Internet.

### ***Progress***

Корпорация Progress Software Corp. основана в 1981 году. В настоящий момент Штаб-квартира корпорации размещается в г. Bedford, USA. Корпорация имеет свыше 1300 сотрудников, работающих в 49 представительствах по всему миру. В 60 других странах Progress представлена компаниями-дистрибьюторами.

Progress Software Corp. является производителем средств разработки приложений (язык 4GL) и SQL сервера базы данных СУБД Progress. Компания производит средства для построения информационных систем масштаба промышленного предприятия, ориентированных на интенсивную обработку транзакций в распределенных вычислительных средах в реальном масштабе времени.

Progress Software Corp. имеет свыше 2300 партнеров во всем мире, имеющих собственные приложения, разработанные на Progress. Суммарный годовой объем продаж программного обеспечения, созданного на базе средств Progress превышает пять миллиардов долларов США. По данным подразделения Dataquest компании Gartner Group Progress Software Corp. занимает 21 процентов рынка встроенных в приложения СУБД и является лидером на этом рынке.

Предлагаемые Progress инструменты позволяют решать все задачи по разработке высокопроизводительных приложений от проектирования ядра базы данных до создания графических инструментов и бизнес-логики



информационной системы, обеспечивая высокую межоперабельность и открытость.

Progress поддерживает как обычный процедурный, событийно-ориентированный, так и объектно-ориентированный стили программирования. Progress позволяет работать с данными, хранящимися в базах данных других форматов: Oracle, Sybase, Rdb, C-ISAM, OS/400 и др.

Приложения, написанные на Progress, полностью переносимы более чем на 160 аппаратных платформ, на которых функционирует Progress, включая MS-Windows, OS/2, Windows NT, Unix ( UnixWare, SCO UNIX, Interactive, Solaris, AIX, HP-UX, IRIX), OpenVMS, IBM AS/400.

Гибкая архитектура сервера позволяет конфигурировать сервер базы данных для оптимальной поддержки требуемого размера базы данных, от маленькой, однопользовательской базы данных до системы, управляющей тысячами пользователей 4GL и SQL. Кроме этого, независимые продавцы ПО или конечные пользователи имеют возможность внедрять только те составляющие сервера базы данных, которые необходимы. Это минимизирует ресурсы и персонал, необходимый для работы с приложением, основанным на реляционной СУБД Progress.

Динамическое ядро базы данных позволяет настраивать и оптимизировать выбранные параметры сервера базы данных во время работы сервера и под нагрузкой реальными пользователями. Это обеспечивает самую высокую степень точности, поскольку можно видеть изменения непосредственно под нагрузкой. В системах, работающих круглосуточно и без перерывов, параметры могут изменяться для того, чтобы использовать преимущества неполной загрузки – без выключения базы данных.

Для соответствия промышленным стандартам, реляционная СУБД Progress может использоваться с широким набором клиентов (Progress 4GL, Java, C, C++, Visual Basic, PowerBuilder и т.д.), включая хост/терминальные

продукты третьих фирм, отвечающие требованиям ODBC и JDBC, такие как средства работы с отчетами, анализ данных, CASE, управление системой и множество других.

Для построения Internet-приложений разработаны продукты: Progress Apptivity 3, WebSpeed 3, ProtoSpeed. WebSpeed – это собственная разработка Progress Software предназначенная как для существующих web-приложений, так и для построения транзакционных Internet-приложений с использованием HTML. WebSpeed WorkShop – это среда, основанная на технологии мастеров, которая включает в себя язык скриптов серверной части, под названием SpeedScript, основанный на progress 4GL (также поддерживаются JavaScript, VBScript, CGI и апплеты Java). Apptivity, выпущенный Progress летом 1997 г, предназначен для построения серверной части приложений, которые могут поддерживать сложные бизнес-процессы посредством визуальной среды разработки, генерирующей Java-код. Apptivity обеспечивает поддержку межбраузерных связей, а также включает в себя интеграцию с CORBA. В дополнение, данный инструмент может быть использован для построения основанных на Java клиент-серверных приложений, и включает встроенный отладчик и Java-компилятор. ProtoSpeed – это отдельное средство отладки на Java, созданное для трассировки распределенных Internet-приложений в гетерогенных средах. Инструмент поддерживает Internet-протоколы: HTTP, FTP, SMTP, POP3 и IMAP4.

### **Оценка характеристик СУБД для построения КИС**

Ранее были сформулированы общие критерии выбора СУБД для построения приложений. Если же необходимо решить задачу выбора СУБД для корпоративной информационной системы, то ряд критериев становится определяющим.

Прежде всего, следует произвести предварительный отбор, отбросив СУБД не соответствующие классу проектируемой системы. Очевидно, информационная система средней и большой по величине корпорации не может строиться на основе «настольных» СУБД. Если же предположить расширения компании, то целесообразно сразу исключить их из рассмотрения.

Исходя анализа общих требований предъявляемых к надежности и производительности, лидируют продукты Oracle, Microsoft SQL Server, Informix, Sybase, DB2 и Progress. Только они могут обеспечить надлежащую масштабируемость будущей системы. Именно эти СУБД позволяют на высоком уровне создавать сложные распределенные архитектуры. С точки зрения требований к рабочей среде не в лучшую сторону выделяется лишь Microsoft SQL Server. Этот продукт имеет ограниченный набор поддерживаемых операционных систем (OS/2, Windows NT, Windows 2000) и, как следствие, может работать на ограниченном наборе аппаратных платформ. В тоже время следует отметить, что, не смотря на все недостатки, благодаря легкости и низкой стоимости администрирования операционная система Windows 2000 вполне может конкурировать с UNIX-подобными.

Гораздо большее внимание следует обращать на моделирование данных. Например, в том случае, когда необходимо работать с большими объемами мультимедийной информации, следует выбирать Oracle. Для большинства приложений вполне достаточно модели данных DB2 или Microsoft SQL Server. Многие из расширенных возможностей (например, объектные расширения) реализованных признанным лидером в разработке СУБД – корпорацией Oracle – скорее всего не будут использованы. Oracle, Microsoft SQL Server, Informix, Sybase, DB2 и Progress поддерживают огромные размеры базы данных, но, как правило, реальный объем данных никогда не достигнет и сотой доли разрешенного.

Большое значение в большинстве случаев играют особенности разработки приложений. СУБД Oracle, Microsoft SQL Server, Informix, Sybase, DB2 и Progress все поддерживают широкий набор языков программирования, но только в Informix и Progress активно используются языки 4GL. Применение языка 4GL серьезно снижает затраты на разработку и позволяет создавать более гибкие приложения.

Далеко не последнюю роль при выборе СУБД играют экономические показатели. Вполне естественно, когда компания приобретает пусть не самый «продвинутый», но более дешевый продукт, который может удовлетворить все потребности предприятия, как в настоящий момент, так и в обозримом будущем. Например, Progress вовсе не уступает Oracle в производительности для большинства задач, возникающих при построении КИС. При этом Progress дешевле, с точки зрения общей стоимости владения базой данных. Нельзя утверждать, что чем дороже СУБД, тем она лучше. Например, все тот же Progress обойдется компании дешевле, чем Microsoft SQL Server, несмотря на то, что Progress поддерживает более широкий набор операционных систем и аппаратных платформ, а также поддерживает язык 4GL. По результатам последних лет Progress является лидером на рынке встроенных баз данных. Ниже приведена таблица для сравнения стоимости владения СУБД Microsoft SQL Server, Progress и Oracle.

**Таблица 1 Явная стоимость пятилетнего владения для Microsoft SQL Server 2000 и для Progress Workgroup RDBMS V9.1 (январь 2001 года)**

	Progress V9.1			Microsoft SQL Server 2000		
	10 клиентов	25 клиентов	Интернет	10 клиентов	25 клиентов	Интернет
Серверные и клиентские лицензии	\$3 250	\$7 250	\$6 250	\$2 249	\$11 099	\$11 099
Средства разработки (Одна копия)	\$3 980	\$3 980	\$4 400	\$1 619	\$1 619	\$1 619
Внедрение	\$9 120	\$12 540	\$11 400	\$8 000	\$11 000	\$10 000
Затраты на администраторов баз данных	\$14 250	\$20 520	\$14 250	\$25 000	\$25 000	\$25 000
Обучение	\$6 384	\$6 384	\$7 980	\$5 330	\$5 330	\$5 330
Три обновления за пять лет	\$0	\$0	\$0	\$1 498	\$5 549	\$4 999
Поддержка/Обслуживание лицензии	\$2 600	\$5 800	\$6 000	\$10 170	\$10 170	\$10 170
Явная стоимость владения базой данных	\$39 584	\$56 474	\$50 280	\$53 866	\$69 767	\$68 217

**Таблица 2 Явная стоимость пятилетнего владения для Oracle8i Enterprise Edition и для Progress Enterprise RDBMS V9.1 (январь 2001)**

	Progress V9.1 Enterprise RDBMS			Oracle8i Enterprise Edition		
	50 клиентов	100 клиентов	Интернет	50 клиентов	100 клиентов	Интернет
Серверные и клиентские лицензии	\$36 750	\$72 000	\$8 900	\$63 333	\$126 666	\$7 410
Средства разработки (Одна копия)	\$3 980	\$3 980	\$4 400	\$4 995	\$4 995	\$4 995
Внедрение	\$62 843	\$123 120	\$15 129	\$94 999	\$189 999	\$13 328
Затраты на администраторов баз данных	\$14 250	\$20 520	\$14 250	\$155 000	\$160 000	\$150 000
Обучение	\$6 384	\$6 384	\$11 400	\$10 581	\$10 581	\$11 115
Три обновления за пять лет	\$0	\$0	\$0	\$0	\$0	\$0
Поддержка/Обслуживание лицензии	\$29 400	\$57 600	\$8 120	\$33 696	\$61 776	\$80 496
Явная стоимость владения базой данных	\$153 607	\$283 604	\$62 199	\$362 604	\$554 017	\$267 344

## ВЫБОР ОПЕРАЦИОННОЙ СИСТЕМЫ И ПРОТОКОЛА СЕТИ

Нет нужды говорить, что правильный выбор операционной системы требует тщательного взвешивания всех "за" и "против". Ведь ошибка в этом

вопросе приводит к неоправданным затратам на разработку прикладных программ, эксплуатацию самой вычислительной системы и ее стыковки с вычислительными сетями.

### **Выбор операционной системы для сервера и клиента**

В первую очередь следует отметить, что задача выбора операционной системы распадается на две подзадачи: выбор операционной системы сервера и выбор операционной системы для клиента.

#### ***Выбор операционной системы для клиентской части***

На выбор операционной системы самое прямое влияние оказывают аппаратные средства. Оставшихся после загрузки операционной системы ресурсов должно хватать для стабильной работы пользовательских приложений. Поэтому при выборе оцениваются потребности в оперативной памяти, дисковой памяти и вычислительных ресурсах процессора. Очень важным является набор поддерживаемых аппаратных платформ. Поскольку корпоративная информационная система предполагает наличие компьютерной сети, операционная система должна поддерживать ряд сетевых протоколов.

Существенное влияние на выбор операционной системы оказывает специфика пользовательских приложений. Именно эта специфика определяет, будет ли использован однозадачный алфавитно-цифровой или многозадачный продукт с мощным графическим интерфейсом. Кроме прочего операционная система должна обеспечивать определенный уровень защиты данных от несанкционированного доступа. При выборе следует учитывать также стоимость операционной системы, поскольку даже небольшая разница в стоимости одной лицензии умноженная на число пользователей может вылиться в круглую сумму.

В качестве операционных систем клиентской части, как правило, выбирают Windows, OS/2, Linux и др. В настоящее время лидером является

Windows. Несмотря на многочисленные упреки, именно она удовлетворяет основные потребности пользователей. Поэтому практически все коммерческие продукты имеют версии для Windows.

### ***Выбор операционной системы для сервера***

К операционной системе для сервера предъявляются особые требования, поскольку от нее будет зависеть работа всей информационной системы. Поскольку сервер должен обладать высокой производительностью, применяются специализированные аппаратные средства. Обычно это многопроцессорные или кластерные архитектуры. Поэтому операционная система в первую очередь должна поддерживать используемую аппаратную платформу. Кроме этого, выбранные серверные приложения должны поддерживать данную версию операционной системы.

Для кластерных или многопроцессорных архитектур с числом процессоров более четырех обычно применяются Unix-подобные операционные системы. Серверы средних и малых предприятий, а также узлы КИС зачастую строятся на базе однопроцессорных архитектур или многопроцессорных с числом процессоров не более четырех. В этом случае спектр возможных операционных систем достаточно широк. На практике обычно выбирают Unix-подобную систему или Windows 2000 (Windows NT). Windows более проста в администрировании и не требует такой высокой квалификации как Unix. Операционную систему Unix применяют только тогда, когда необходимо обеспечить особую надежность и защищенность.

### **Выбор протокола сети**

Современная корпоративная информационная система должна обеспечивать как локальный, так и удаленный сетевой доступ пользователей через Internet. Поэтому наиболее простым решением является выбор широко

известного сетевого протокола TCP/IP. Этот протокол поддерживается практически всеми современными СУБД и операционными системами.

## **ОРГАНИЗАЦИЯ И ХРАНЕНИЕ ИНФОРМАЦИИ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ**

Корпоративная информационная система, как правило, должна обеспечивать работу нескольких территориально распределенных подразделений. В связи с этим становится невозможным применение централизованной архитектуры базы данных. Информационные ресурсы должны стать распределенными.

Распределенная база данных (РаБД) – это совокупность логически взаимосвязанных баз данных, распределенных в компьютерной сети. Это определение можно дополнить, если рассмотреть следующие характеристики РаБД:

1. *Локальная автономность.* Локальные данные должны находится под локальным владением и управлением, включая функции безопасности, целостности, представления данных. Исключением может быть ситуация, когда ограничения целостности охватывают данные нескольких узлов и когда управление распределенной транзакцией осуществляется некоторым внешним узлом.

2. *Никакой конкретный сервис не должен возлагаться на какой-либо специально выделенный центральный узел.* Соблюдение этого правила, т.е. принципа децентрализованности функций, позволяет избежать узких мест.

3. *Непрерывность функционирования.* Система не должна останавливаться в случае необходимости добавления нового узла или удаления в распределенной среде некоторых данных и осуществления перехода к новой версии СУБД на отдельном узле.



4. *Независимость от местоположения.* Пользователи и приложения не обязаны знать о том, где физически располагаются данные.

5. *Независимость от фрагментации.* Фрагменты базы данных обрабатываются таким образом, чтобы пользователи и приложения могли вообще ничего не знать об этом. Более того необходимо обходить при обработке запросов фрагменты, не имеющие к ним никакого отношения.

6. *Независимость от тиражирования.*

7. *Распределенная обработка запросов.*

8. *Управление распределенными транзакциями.* Необходимо использовать механизмы управления транзакциями и одновременным доступом позволяющие выявлять и разрешать тупиковые ситуации прерывания по истечении временных интервалов, фиксировать и откатывать распределенные транзакции.

9. *Независимость от оборудования.*

10. *Независимость от операционных систем.*

11. *Независимость от сети.*

12. *Независимость от локальных СУБД.*

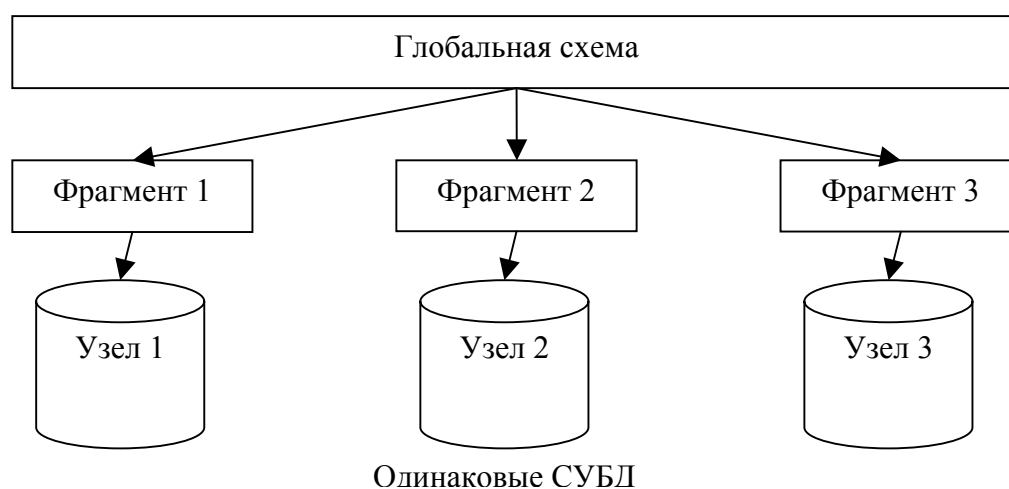
На практике достаточно сложно реализовать систему, удовлетворяющую всем двенадцати пунктам.

## **МОДЕЛИ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ**

Распределенные базы данных могут быть однородными и неоднородными, они могут строиться по принципам «сверху вниз» и «снизу вверх». Кроме этого различаются и формы распределения данных. В одних случаях данные фрагментируются, т.е. делятся на пропорции, распределяемые между множеством физических ресурсов. В других случаях данные тиражируются, т.е. дублируются на нескольких узлах.

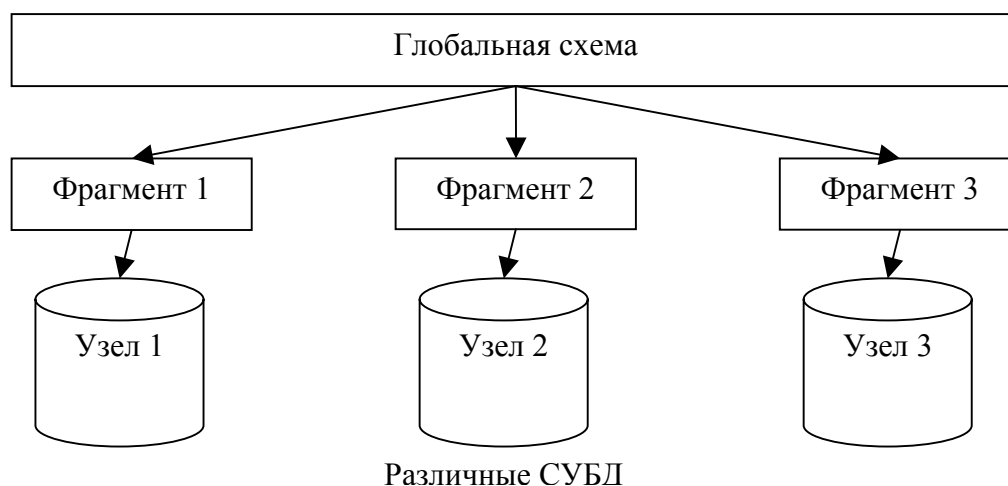
## Однородные и неоднородные системы

Однородные распределенные системы баз данных имеют в своей основе один продукт СУБД. СУБД с поддержкой однородного распределения являются сильно связанными системами, их встроенные средства оптимизированы и настроены для достижения максимальной производительности и пропускной способности. Однородные системы чаще всего проектируются методом «сверху вниз».



**Рисунок 5 Структура однородной РаБД**

Неоднородные распределенные системы включают два или более существенно различающихся продукта управления данными. Неоднородные системы чаще всего проектируются методом «снизу вверх».



**Рисунок 6 Структура неоднородно РаБД**

### **Методы построения РаБД «сверху вниз» и «снизу вверх»**

Проектирование РаБД методом «сверху вниз» осуществляется в целом аналогично проектированию централизованных баз данных. Однако при проектировании предполагается, что ее объекты не будут сосредоточены в одном месте, а распределяются по нескольким вычислительным системам. Распределение производится путем фрагментации и тиражирования.

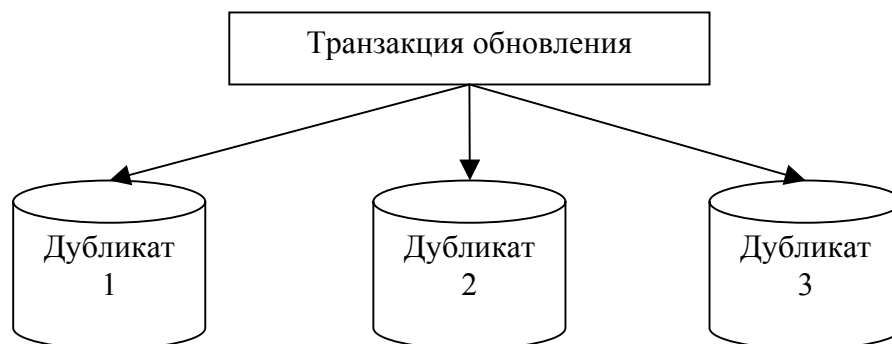
Фрагментация означает декомпозицию объектов базы данных, таких, как реляционные таблицы, на две или более частей, которые размещаются на разных компьютерных системах. Различают два вида фрагментации.

Горизонтальная фрагментация (фрагментация по строкам) означает распределение по узлам строк таблицы. Распределение записей может осуществляться в соответствии со значением параметров (например, на каждом узле магазина хранятся продажи только этого магазина) или не на основе значений (например, по принципу «карусели»).

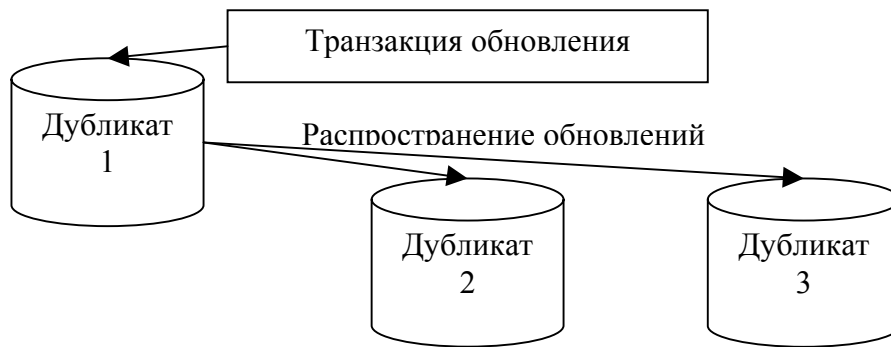
Вертикальная фрагментация (фрагментация по столбцам) означает распределение по узлам столбцов таблицы.

В независимости от вида фрагментации поддерживается глобальная схема, позволяющая воссоздать из имеющихся фрагментов логически централизованную таблицу. Но следует отметить, что идея глобальной схемы над распределенными фрагментами на практике не настолько привлекательна, как принято было считать. Динамические выборки в среде сильно распределенных данных могут поглотить большую долю производительности из-за необходимости пересылок больших объемов данных с одного узла на другой. Для решения этой проблемы могут использоваться алгоритмы интеллектуальной обработки запросов. Но полезности этого подхода к задачам реального управления информацией остается под вопросом.

Тиражирование (или репликация) означает создание дубликатов данных. Репликаты – множество различных физических копий некоторого объекта базы данных, для которого в соответствии с определенными правилами поддерживается синхронизация с некоторой «главной» копией. Возможны различные модели тиражирования.



**Рисунок 7 Одновременное обновление (с управлением параллелизмом)**

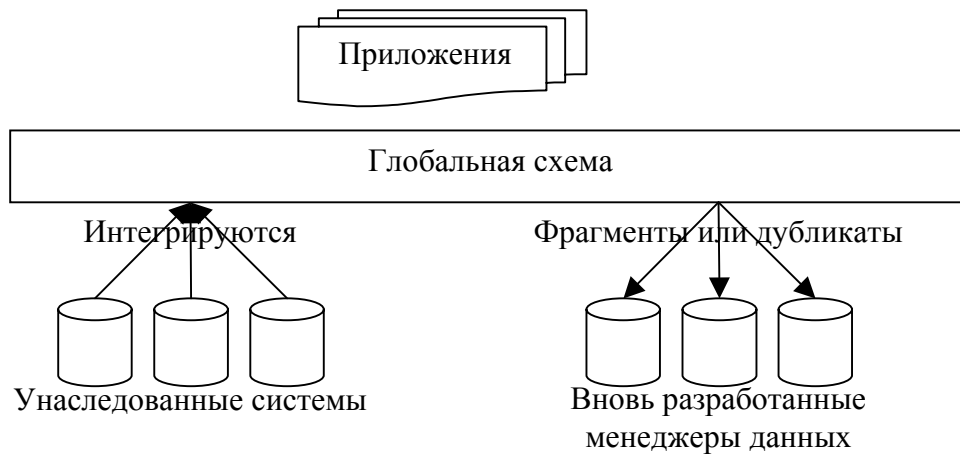


**Рисунок 8 Распространенные обновления**



**Рисунок 9 Запланированная синхронизация дубликатов только для чтения**

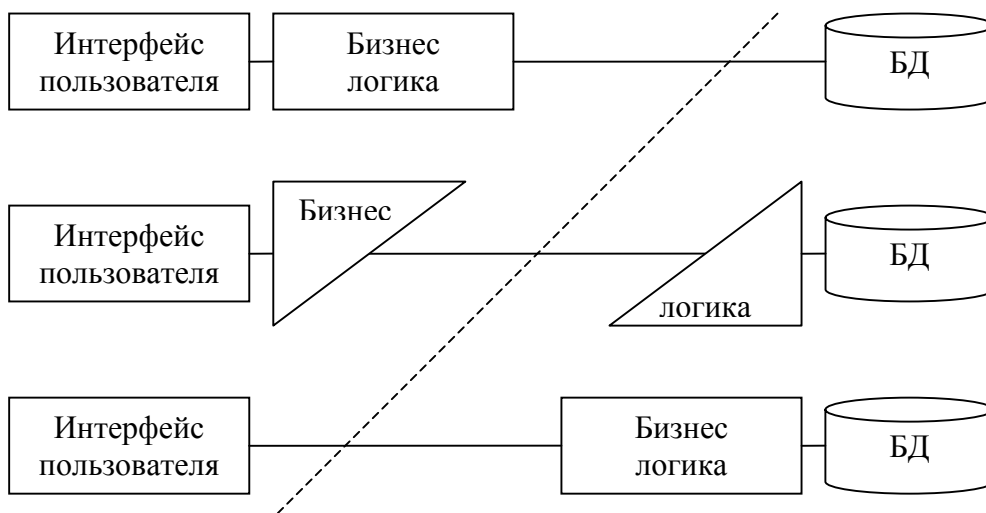
Подход проектирования «сверху вниз» обычно применяется к однородным системам. Этот оправдан при создании новых приложений. На практике часто приходится решать задачу интеграции существующих систем. В этом случае разработчик не может позволить себе «роскошь» проектирования «сверху вниз». Приходится прибегать к методу «снизу вверх», где основной проблемой становится объединение существующих схем баз данных, чтобы предоставить как новым, так и старым приложениям доступ к новым и старым ресурсам данных.



**Рисунок 10 Интеграция распределенных баз данных «снизу вверх»**

## **РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ**

Для повышения эффективности функционирования корпоративной информационной системы (минимизация сетевого трафика, распределение вычислительных мощностей и т.п.) целесообразно применять технологии распределения вычислений. Распределение вычислений заключается в частичном или полном переносе вычислений на сервер базы данных.



**Рисунок 11 Распределенные вычисления**

Наиболее эффективной технологией распределения вычислений является построение трехуровневой (или многоуровневой) архитектуры при помощи сервера приложений.



**Рисунок 12 Трехуровневая архитектура с использованием сервера приложений**

Система разбивается на три уровня: уровень представления (реализующий функции ввода и отображения данных); прикладной уровень (реализующий универсальные сервисы, а также функции, специфичные для определенной предметной области); уровень доступа к информационным ресурсам (реализующий фундаментальные функции хранения и управления информационно-вычислительными ресурсами).

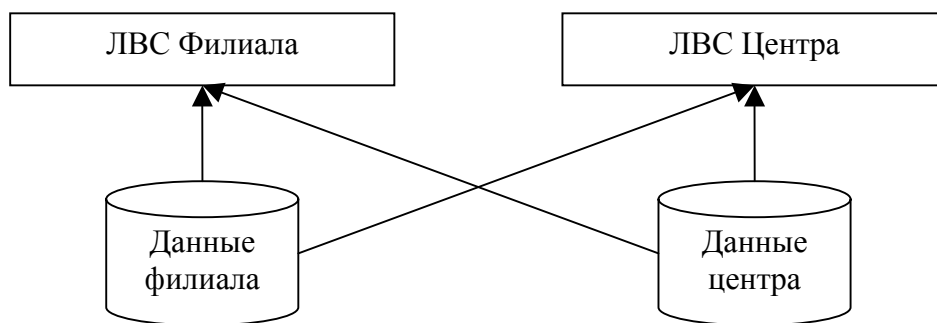
## **ВЫБОР СТРУКТУРЫ РАБД ДЛЯ КИС УПРАВЛЕНИЯ СЕТЬЮ ФИЛИАЛОВ**

Рассматривается предприятие, состоящее из сети  $N$  однотипных филиалов и одного центра управления. Распределенная база данных в этом случае представляет совокупность логически взаимосвязанных базы данных центра управления и  $N$  однотипных баз данных филиалов. Поскольку транзакции изменения данных, как правило, не выходят за пределы локальной

сети, методом их оптимизации является распределение вычислений для случая ЛБД. Основное внимание сосредоточим на оптимизации запросов к БД.

При условии отсутствия прямых связей между филиалами, все запросы к распределенной базе данных можно классифицировать следующим образом:

- Запросы к данным филиала в пределах локальной сети филиала
- Запросы к данным филиала из центра
- Запросы к данным центра из филиала
- Запросы к данным центра в пределах локальной сети центра

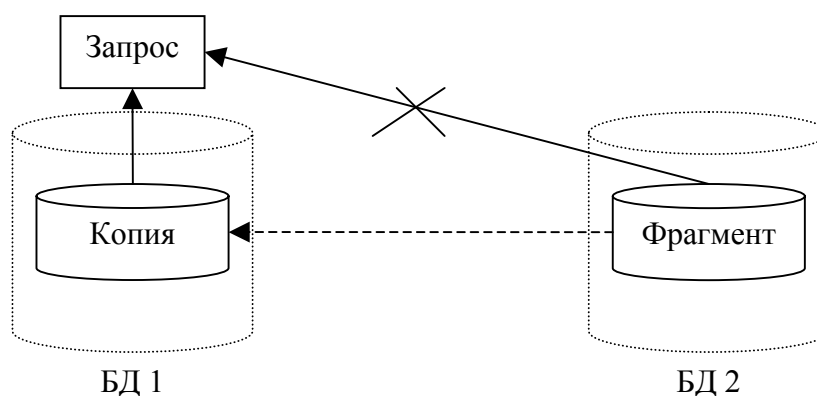


**Рисунок 13 Виды запросов**

Также все запросы можно дополнительно разбить на единовременные (данные передаются единовременно) и многократные (одни и те же данные передаются многократно). Следует отметить, что наиболее дорогостоящими являются удаленные запросы, т.е. запросы между филиалом и центром. Именно они должны быть существенно оптимизированы. Прежде всего, целесообразно применение распределенных вычислений, когда передаются только конечные данные, а вся предварительная обработка осуществляется в пределах сервера удаленной базы данных. Естественно при этом возрастает нагрузка на сервер БД, но она в большинстве случаев окупается за счет существенного ускорения выполнения запросов и уменьшения трафика. Для повышения скорости обработки многократных запросов применяются методы хранилищ данных, которые осуществляют агрегирование данных. Агрегирование требует



дополнительных вычислительных ресурсов и объема памяти на жестких дисках. Процесс агрегирования может осуществляться асинхронно, когда обработка осуществляется после изменения исходных данных. Это позволяет более равномерно распределять вычислительные ресурсы во времени, но при этом возникают задачи непротиворечивости и достоверности данных. Тиражирование чаще применяется для оптимизации многократных удаленных запросов, а также в случаях особых требований к скорости выполнения однократных запросов. При тиражировании вновь возникают задачи непротиворечивости и достоверности данных.



**Рисунок 14 Применение тиражирования для оптимизации многократных удаленных запросов**

Данные базы данных можно условно разделить на три группы:

1. Справочники (классификаторы, настройки и т.п.)
2. Статические данные (состояние объектов)
3. Динамические данные (история изменения состояний объектов)

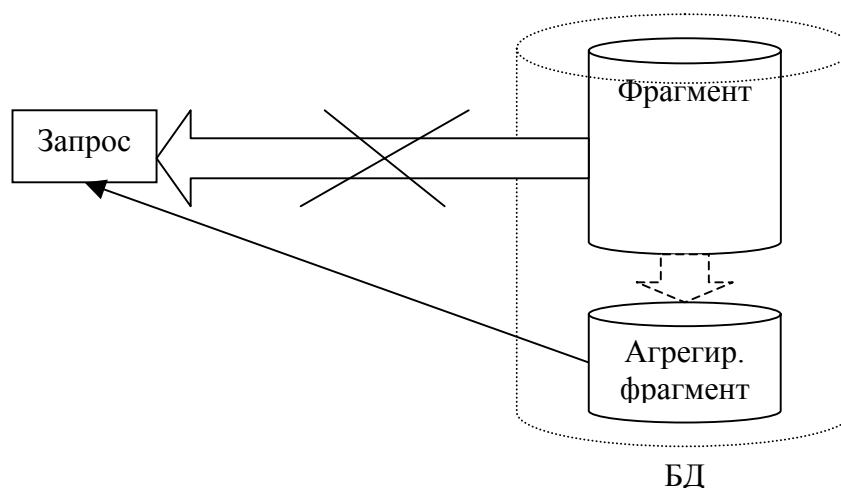
Свойства каждой группы определяют выбор стратегии оптимизации доступа.

К «справочникам» относятся данные, которые крайне редко добавляются, изменяются и удаляются. Группа имеет малый объем по сравнению со всей базой данных. При этом, данные участвуют в большинстве запросов, и высока

частота многократного обращения. В качестве примера можно привести классификаторы или настройки. В силу высокой частоты обращения (в том числе повторного) и крайне редкого изменения данных наиболее оптимальным методом оптимизации доступа является тиражирование. Поскольку объем информации мал, то в зависимости от требований к достоверности и непротиворечивости может быть использован вариант как одновременного (параллельного), так и отложенного тиражирования.

К «статическим данным» относится информация, которая также редко, как в случае справочников, добавляется и удаляется, но при этом имеет место очень частое изменение. Эти данные характеризуют текущее состояние объектов. В качестве примера можно привести балансовые счета. Группа обычно составляет малую часть базы данных. Информация группы имеет высокую частоту обращения. Следует отметить, что, как правило, именно на группу статических данных накладывается большая часть условий ограничения целостности и непротиворечивости. В связи с выше сказанным, для оптимизации доступа лучше всего подходит параллельное тиражирование.

К группе «динамических данных» относится информация об изменении состояний объектов. Эти данные не обновляются и не удаляются, но часто добавляются. В большинстве случаев, именно динамические данные занимают «львиную долю» базы. В качестве примера можно привести историю продаж. Именно к динамическим данным, в силу их свойств, и применяется агрегирование. В большинстве случаев интересно не точное время изменения состояния объекта, а принадлежность времени некому интервалу. При этом, кроме уменьшения использования вычислительных ресурсов достигается сокращение объема передаваемой по сети информации. Например, если не существенна информация о продажах товара в пределах дня, то вместо операций чтения нескольких записей и последующего их суммирования выполняется операция чтения только одной записи.



**Рисунок 15 Применение агрегирования для оптимизации запросов**

Повторное (многократное) обращение к одним и тем же динамическим данным происходит редко, но стоимость таких запросов высока, поскольку передаются большие объемы информации. Поэтому применяется тиражирование исходных или агрегированных данных. Следует отметить, что динамические данные могут с течением времени терять актуальность, тогда при ограничениях на размеры БД возникают задачи архивирования или чистки.

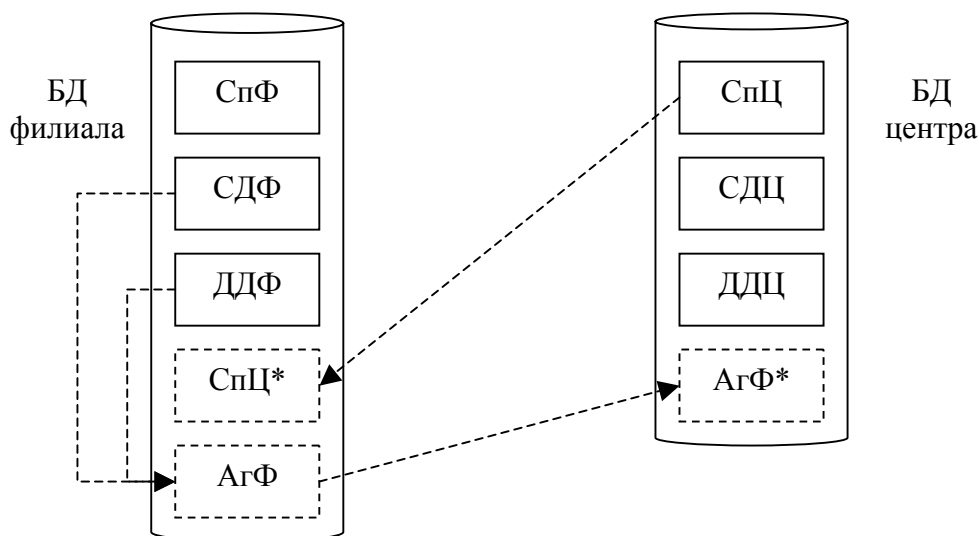
**Таблица 3 Характеристики группы данных**

	Справочники	Статические данные	Динамические данные
Обновление	редко	часто	не обновляются
Добавление	редко	редко	Часто
Удаление	редко	редко	не удаляются *
Объем информации	маленький	маленький	Большой
Частота обращения к одному и тому же блоку	большая	большая	Маленькая
Методы оптимизации	тиражирование	тиражирование	агрегирование, тиражирование

\* - Динамические данные не удаляются, тем не менее, существуют сроки полезности, при превышении которых данные помещаются в архив

Рассмотрим случай, когда центр осуществляет управление филиалами путем выработки правил и планов функционирования. Причем управленческие

решения принимаются на основании стандартизированной информации по филиалам. Тогда модель распределенной базы данных корпоративной системы управления сетью филиалов будет выглядеть следующим образом:



Сп – справочники, СД – статические данные, ДД – динамические данные,  
 Аг – агрегированные данные, \* – копия, Ф – филиал, Ц – центр

**Рисунок 16 РаБД КИС управления сетью филиалов**

## РАСЧЕТ ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ФУНКЦИОНИРОВАНИЯ РАБД

Имеется  $N$  однотипных ЛБД филиалов и одна ЛБД центра управления. Будем рассматривать случай, когда филиалы не взаимодействуют между собой, центр и филиалы связаны посредством ADSL. Тиражирование справочников центра на филиалы, по сути, есть передача глобального управления. Кроме глобального управления осуществляется локальное самоуправление филиалом.

Введем обозначения:

$T^\Phi$  – период агрегирования оперативной БД филиала

$T^\Pi$  – период тиражирования хранилища филиала

$k_{agr}^\Phi$  – коэффициент агрегирования БД филиала

$v^\Phi$  – мощность потока исходных данных БД филиала

$v^\Pi$  – мощность потока запросов данных пользователями филиала

$q^{\Pi}$  – число запросов пользователей

$N$  – число филиалов

$V^{\Pi}$  – объем справочников центра (глобального управления)

$T_y^{\Phi}$  – период локального управления

$T_y^{\Pi}$  – период тиражирования из центра (глобального управления)

$V_{ИУ}^{\Phi}$  – обрабатываемый объем данных для расчета локального управления

$V_y^{\Phi}$  – объем коррекции данных после расчета локального управления

$s_{\phi\phi}^{\Phi}$  – время обработки единицы данных СУБД филиала

$s^{\Pi}$  – время обработки единицы данных ЦП рабочей станции пользователя

$z$  – параметр распределения вычислений

$k_{cn}^{\Phi}$  – относительное сокращение объема данных при выполнении запроса

через сервер приложений

СУБД филиала выполняет задачи записи исходных данных, записи данных агрегирования в хранилище, записи данных тиражирования из центра, записи локального управления, чтения данных запрашиваемых пользователями, чтения данных для агрегирования, чтения данных для тиражирования в центр и чтения данных для локального управления. При этом одна операция записи соответствует  $k_{\phi\phi}^{\Phi}$  операций чтения.

$$k_{\phi\phi}^{\Phi} \cdot \left( v^{\Phi} + v^{\Phi} \cdot k_{aep}^{\Phi} + \frac{V^{\Pi}}{T_y^{\Pi} + T^{\Pi} + T^{\Phi}} + \frac{V_y^{\Phi}}{T_y^{\Phi} + T^{\Phi}} \right) + v^{\Pi} + v^{\Phi} + \frac{v^{\Phi} \cdot k_{aep}^{\Phi}}{T^{\Pi} + T^{\Phi}} + \frac{V_{ИУ}^{\Phi}}{T_y^{\Phi} + T^{\Phi}} \leq \frac{1}{s_{\phi\phi}^{\Phi}}$$

Локальная сеть филиала выполняет задачи передачи результатов прямых запросов и передачи результатов запросов через сервер приложений.

$$z \cdot v^{\Pi} + (1 - z) \cdot k_{cn}^{\Phi} \cdot v^{\Pi} \leq \frac{1}{s_{cem}^{\Phi}}$$

Глобальная сеть выполняет задачи передачи данных тиражирования от филиала центру и от центра филиалу. При этом входящая скорость в  $k^{ADSL}$  раз больше исходящей.

$$N \cdot k^{ADSL} \cdot \frac{V^U}{T_y^U + T^U + T^\Phi} + N \cdot \frac{v^\Phi \cdot k_{azp}^\Phi}{T^U + T^\Phi} \leq \frac{1}{s_{cem}^{ADSL}}$$

$$k^{ADSL} \cdot \frac{v^\Phi \cdot k_{azp}^\Phi}{T^U + T^\Phi} + \frac{V^U}{T_y^U + T^U + T^\Phi} \leq \frac{1}{s_{cem}^{ADSL}}$$

ЦП рабочей станции пользователя выполняет задачу обработки данных прямых запросов.

$$z \cdot v^\Pi \leq \frac{1}{s^\Pi}$$

ЦП сервера филиала помимо обслуживания работы СУБД выполняет задачи расчета агрегирования, расчета локального управления и обработки данных запросов через сервер приложений.

$$g_{\delta\delta}^\Phi \cdot \left( k_{\delta\delta}^\Phi \cdot \left( v^\Phi + v^\Phi \cdot k_{azp}^\Phi + \frac{V^U}{T_y^U + T^U + T^\Phi} + \frac{V_y^\Phi}{T_y^\Phi + T^\Phi} \right) + v^\Pi + v^\Phi + \frac{v^\Phi \cdot k_{azp}^\Phi}{T^U + T^\Phi} + \frac{V_{HY}^\Phi}{T_y^\Phi + T^\Phi} \right) +$$

$$+ g_{azp}^\Phi \cdot v^\Phi \cdot \frac{G_{azp}^\Phi}{T^\Phi} + \frac{G_y^\Phi}{T_y^\Phi + T^\Phi} + g_{cn}^\Phi \cdot (1-z) \cdot v^\Pi \leq \frac{1}{s_{обп}^\Phi}$$

Среднее время обработки запроса пользователей:

$$t_{cp}^\Pi = \frac{T}{q^\Pi} \cdot \left[ s_{\delta\delta}^\Phi \cdot \left( k_{\delta\delta}^\Phi \cdot \left( v^\Phi + v^\Phi \cdot k_{azp}^\Phi + \frac{V^U}{T_y^U + T^U + T^\Phi} + \frac{V_y^\Phi}{T_y^\Phi + T^\Phi} \right) + v^\Pi + v^\Phi + \frac{v^\Phi \cdot k_{azp}^\Phi}{T^U + T^\Phi} + \frac{V_{HY}^\Phi}{T_y^\Phi + T^\Phi} \right) + \right.$$

$$\left. + s^\Pi \cdot z \cdot v^\Pi + s_{azp}^\Phi \cdot v^\Phi + \frac{S_{azp}^\Phi}{T^\Phi} + \frac{S_y^\Phi}{T_y^\Phi + T^\Phi} + s_{cn}^\Phi \cdot (1-z) \cdot v^\Pi + \right.$$

$$\left. + s_{cem}^\Phi \cdot (z \cdot v^\Pi + (1-z) \cdot k_{cn}^\Phi \cdot v^\Pi) \right]$$

Штраф за превышение трафика глобальной сети:

$$C^{ADSL} = \begin{cases} 0, v^{ADSL} \cdot T < V_{абон}^{ADSL} \\ c^{ADSL} \cdot (v^{ADSL} \cdot T - V_{абон}^{ADSL}) \end{cases}, \text{ где } v^{ADSL} = N \cdot \left( \frac{V^U}{T_y^U + T^U + T^\Phi} + \frac{v^\Phi \cdot k_{azp}^\Phi}{T^U + T^\Phi} \right)$$

Штраф за задержки локального управления:

$$C_{yup}^{\Phi} = \begin{cases} 0, T_y^{\Phi} + T^{\Phi} < T_{yup}^{\Phi} \\ c_{yup}^{\Phi} \cdot T \cdot \left( 1 - \frac{T_{yup}^{\Phi}}{T_y^{\Phi} + T^{\Phi}} \right) \end{cases}$$

Штраф за задержки глобального управления:

$$C_{yup}^{\Pi} = \begin{cases} 0, T_y^{\Pi} + T^{\Pi} + T^{\Phi} < T_{yup}^{\Pi} \\ c_{yup}^{\Pi} \cdot T \cdot \left( 1 - \frac{T_{yup}^{\Pi}}{T_y^{\Pi} + T^{\Pi} + T^{\Phi}} \right) \end{cases}$$

Штраф за задержки обработки запросов пользователей:

$$C_{cp}^{\Pi} = \begin{cases} 0, t_{cp}^{\Pi} < T_{cp}^{\Pi} \\ c_{cp}^{\Pi} \cdot (t_{cp}^{\Pi} - T_{cp}^{\Pi}) \end{cases}$$

Параметры функционирования  $(T^{\Phi}, T^{\Pi}, T_y^{\Pi}, T_y^{\Phi}, z)$  должны выбираться таким образом, чтобы достигался минимум суммарно штрафа.

$$C^{ADSL} + C_{yup}^{\Phi} + C_{yup}^{\Pi} + C_{cp}^{\Pi} \xrightarrow{(T^{\Phi}, T^{\Pi}, T_y^{\Pi}, T_y^{\Phi}, z)} \min$$

$$1 \leq T^{\Phi} \leq T$$

$$1 \leq T_y^{\Phi} + T^{\Phi} \leq T$$

$$1 \leq T^{\Pi} + T^{\Phi} \leq T$$

$$1 \leq T_y^{\Pi} + T^{\Pi} + T^{\Phi} \leq T$$

$$0 \leq z \leq 1$$

Полученная задача представляет собой задачу нелинейного программирования с линейными ограничениями. Численное решение может быть найдено при помощи современных математических программных пакетов.

## **ПРИМЕР ПРОЕКТИРОВАНИЯ КИС КЛАССА ERP ДЛЯ УПРАВЛЕНИЯ СЕТЬЮ ПРЕДПРИЯТИЙ РОЗНИЧНОЙ ТОРГОВЛИ**

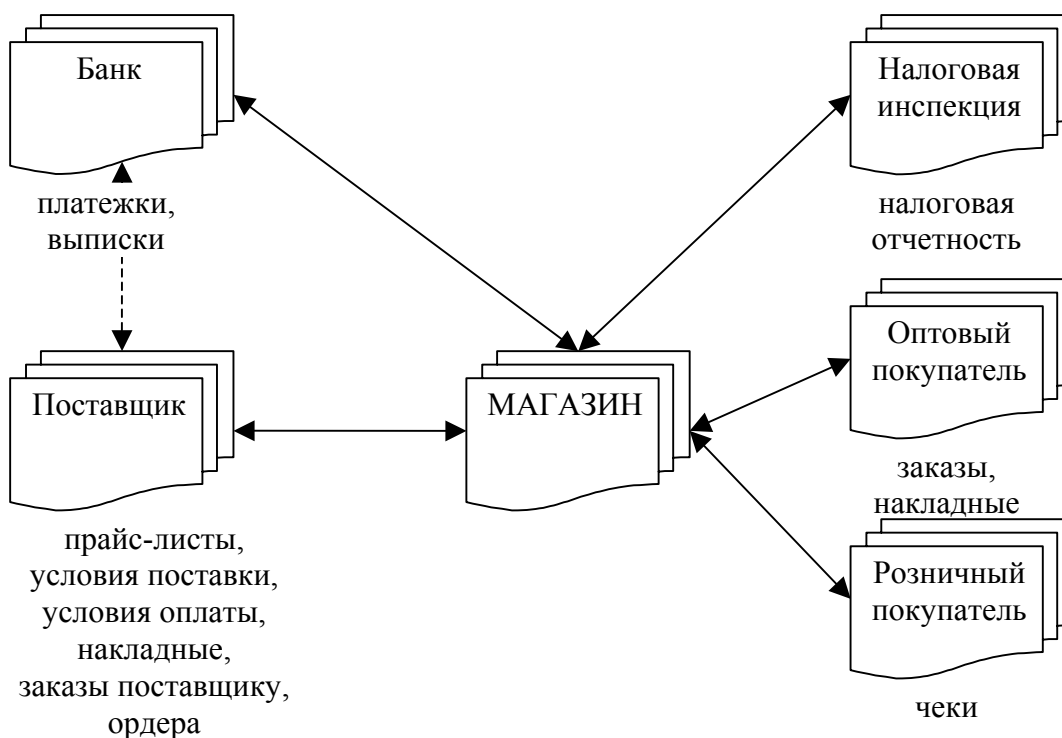
Рассмотрим проблему построения корпоративной информационной системы для сети крупных территориально распределенных торговых предприятий в сфере розничной торговли.

Интегрированная система автоматизации сети торговых предприятий является сложным комплексом аппаратно-программных средств. Эффективность, надежность и функциональные возможности системы во многом зависят от эффективности корпоративной информационной системы (КИС).

## АНАЛИЗ ПРОБЛЕМЫ

Концептуальные положения для проектирования КИС определяются задачами, стоящими перед магазином, как перед структурной единицей сферы товарно-денежного обращения.

Магазин взаимодействует со следующими внешними объектами: поставщики, банки, налоговая инспекция, оптовый и розничный покупатель.

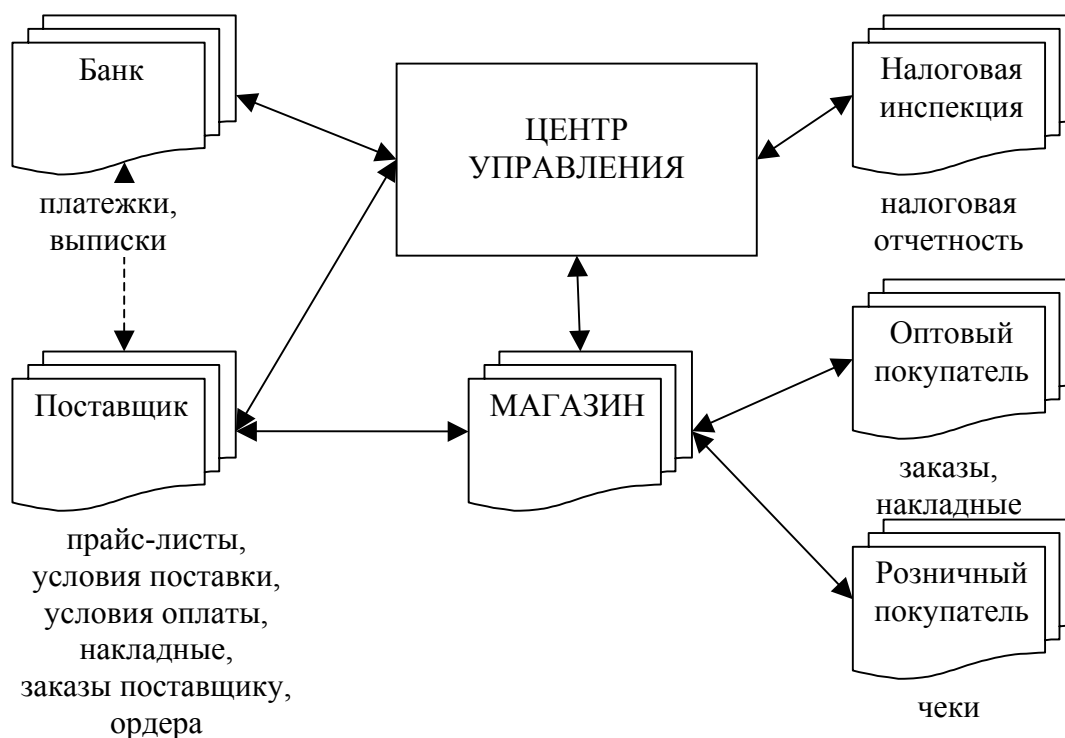


**Рисунок 17 Структура взаимодействия сети торговых предприятий с внешними объектами**

В современных условиях экономически целесообразно выделить в качестве отдельного подразделения центр управления сетью торговых



предприятий, который возьмет на себя часть функций по работе с поставщиками и полностью освободит магазины от взаимодействия с банками и налоговой инспекцией.

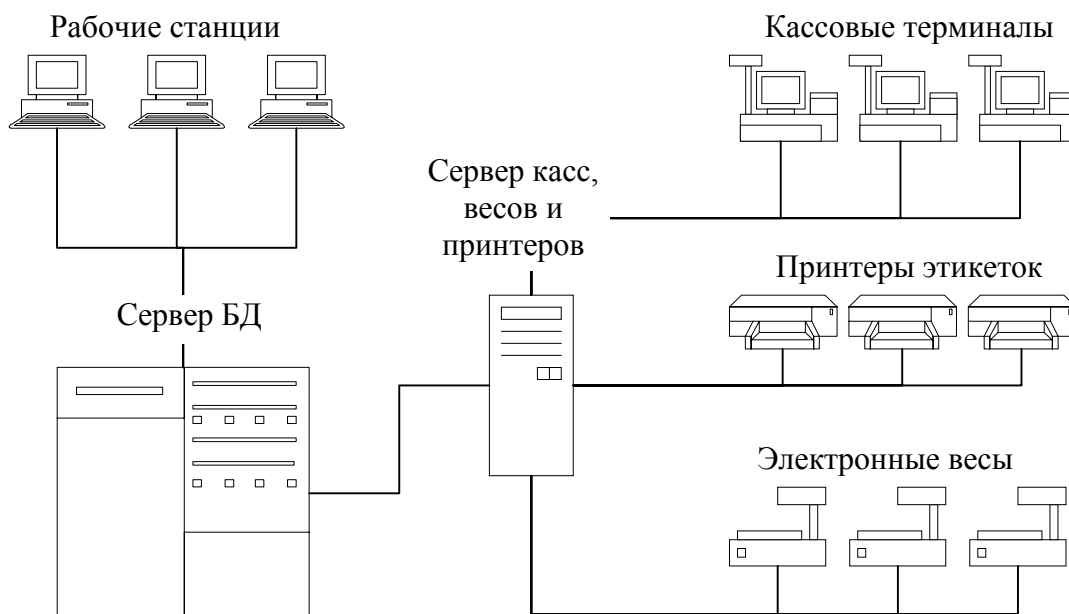


**Рисунок 18 Модифицированная структура взаимодействия**

Предложенная структура может работать только при полной автоматизации каждого магазина и при налаженном оперативном обмене информацией. То есть, требуется корпоративная информационная система способная решить поставленные задачи.

Что касается структуры аппаратных средств, то она призвана, в первую очередь, обеспечивать работу торгового процесса. То есть необходимо организовать эффективное функционирование устройств front-office, а именно: кассовых терминалов, электронных весов и принтеров этикеток. Структуру аппаратных средств можно считать эффективной, если она делает возможным оперативный обмен данными и непрерывную работу устройств front-office. В

связи с этим вводится выделенный сервер управления кассами, весами и принтерами.



**Рисунок 19 Структура основных аппаратных средств**

Сервер касс, весов и принтеров берет на себя все функции по обмену данными с устройствами front-office, в том числе авторизацию пластиковых карт.

## **ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ИС ТОРГОВОГО ПРЕДПРИЯТИЯ**

В системе применяется натурально-стоимостная система учета, т.е. учет товаров осуществляется как в количественном, так и в стоимостном выражении. Система обеспечивает ведение всей деятельности предприятия. Заложена возможность настройки системы при изменении бизнес-правил или плана счетов.

Система построена в виде набора модулей, которые связаны посредством ядра системы. Такой способ организации позволяет конфигурировать систему для учета потребностей предприятия, а также добавлять новые модули для дальнейшего расширения функциональности системы.

## **Ядро системы**

Ядро системы играет роль фундамента, на котором, как здание из кирпичиков, из модулей формируется система. Ядро состоит из следующих элементов:

- Журнал товарных операций
- Журнал денежных операций
- Главная бухгалтерская книга
- Книга товаров
- Настройки и бизнес-правила
- Права доступа

### ***Главная бухгалтерская книга и журнал денежных операций***

Главная бухгалтерская книга представляет собой план счетов. Все счета имеют одинаковую структуру. По каждому счету отображается информация о текущих остатках денежных средств, оборотах. Каждый счет имеет уровень учета – уровень, по которому производится учет данного счета главной бухгалтерской книге. Высший уровень суммирования счетов главной бухгалтерской книги может суммировать счета низших уровней. Текущее состояние счета находит отражение в статусе. Счет может быть действующим (статус 0-8), или замороженным (статус - 9). Если счет заморожен, то система не позволяет делать проводки ни по одному лицевому счету, открытому по данному счету. План счетов отображает текущее состояние всех финансов предприятия.

Журнал денежных операций содержит информацию обо всех движениях денежных средств. В журнале фиксируются следующие основные данные:

- Дата и время совершения операции
- Код и вид операции
- Номера корреспондирующих счетов

- Сумма операции
- Вид и идентификатор документа

На основании журнала денежных операций и текущего состояния плана счетов может быть получена информация о состоянии плана счетов на любой момент времени.

### ***Книга товаров и журнал товарных операций***

Книга товаров представляет собой классифицированный набор карточек товаров. Карточка товара содержит следующую информацию о товаре:

- Наименование
- Признак весовой/штучный
- Ссылки на справочники производителей, стран, единиц измерения и прочие
- Список штрих-кодов
- Цена реализации
- Код товарной группы
- Текущие остатки
- Ставки НДС и налога с продаж

Книга товаров отражает текущее состояние всех товаров предприятия.

Журнал товарных операций содержит информацию обо всех движениях товаров. В журнале фиксируются следующие основные данные:

- Дата и время совершения операции
- Код и вид операции
- Ссылка на карточку товара
- Количество операции
- Суммы операции в учетных ценах и ценах реализации
- Вид и идентификатор документа

На основании журнала товарных операций и текущего книги товаров может быть получена информация о состоянии книги товаров на любой момент времени.

### ***Права доступа, настройки и бизнес-правил***

Система на уровне ядра поддерживает условия безопасности. К числу мер относятся: ограничение прав доступа к системе и к отдельным ее функциям, регистрация в журналах и документах информации об ответственном пользователе.

Ядро системы, кроме прочего, включает в себя настройки и бизнес-правила. Причем бизнес-правила могут быть изменены пользователем. Таким образом, система может быть перенастроена для учета изменений потребностей предприятия.

### **Основные модули системы**

Информационная система торгового предприятия состоит из следующих основных модулей:

- Управление финансами
- Управление заказами и поставками
- Управление складом
- Управление торговым залом
- Управление продажами
- Управление производством полуфабрикатов
- Управление эксплуатацией имущества предприятия
- Управление персоналом и расчет заработной платы
- Справочники
- Отчеты и аналитика

### ***Управление финансами***

Модуль управления финансами позволяет выполнять в системе все операции связанные с денежными средствами. Особое внимание уделено автоматизации расчетов с поставщиками и финансовому планированию.

### ***Управление заказами и поставками***

Модуль предназначен для управления заказами и поставками. Система позволяет на основании анализа автоматически формировать план поставок и заказы поставщикам, а также осуществлять контроль над их выполнением. Особое внимание уделено ускорению приемки товара и обеспечению непрерывности товарного запаса при минимуме складских площадей.

### ***Управление складом***

Модуль управления складом выполняет функции учета товаров на складе.

### ***Управление торговым залом***

Модуль предназначен для управления торговым залом. Одна из основных функций – оптимизация размещения товара в торговом зале. В системе активно используется понятие ячейки торгового зала. Это позволяет вовремя и быстро производить выкладку товара. Кроме этого в систему заложены алгоритмы расчета размеров ячеек торгового зала для достижения таких целей, как уменьшение количества внеплановых выкладок, повышение эффективности использования торговых площадей, и другие.

### ***Управление продажами***

Предназначение модуля управления продажами повышение качества обслуживания покупателя и получение максимального дохода. Модуль выполняет функции обмена данными с кассами и электронными весами, планирования продаж, расчет цен реализации и прочие.

### ***Управление производством полуфабрикатов***

Модуль управления производством полуфабрикатов предназначен для автоматизации формирования планов производства и контроля их выполнения.

В систему заложены алгоритмы расчета себестоимости полуфабрикатов и контроля расхода сырья.

#### ***Управление эксплуатацией имущества предприятия***

Модуль предназначен для управления эксплуатацией имущества предприятия. К основным функциям относятся учет имущества предприятия, автоматическое начисление амортизации, планирование и контроль расходных материалов и прочие.

#### ***Управление персоналом и расчет заработной платы***

Модуль выполняет такие функции, как формирование и контроль графиков работы персонала, учет рабочего времени и расчет заработной платы, и другие.

#### ***Справочники***

К справочникам относятся как всевозможные классификаторы (например, товарные группы), так и группы объектов (например, поставщики)

#### ***Отчеты и аналитика***

Модуль отчеты и аналитика выполняет функции по формированию отчетности и анализу функционирования предприятия для принятия управленческих решений.

### **РАСЧЕТ ПАРАМЕТРОВ ФУНКЦИОНИРОВАНИЯ РАБД**

Распределенная база данных построена на основании положений описанных выше. На основании предложенных алгоритмов был произведен расчет параметров функционирования РаБД для минимизации стоимости эксплуатации. На основании следующих исходных данных.

СУБД: Enterprise Progress 9.1B28

Сервер: Intel 2 x Pentium III 800 RAM 512 RAID 5 OS Windows 2000 Server.

Клиент: Celeron 700 RAM 64 OS Windows 98

$T = 43200$  минут (период)

$v^\Phi = 1560$  (мощность потока исходных данных БД филиала)

$v^\Pi = 30000$  (мощность потока запросов данных пользователями филиала)

$q^\Pi = 50000$  (число запросов пользователей)

$N = 3$  (число филиалов)

$V^Ц = 5000$  (объем справочников центра)

$V_{IV}^\Phi = 25000$  (обрабатываемый объем данных для расчета локального управления)

$V_y^\Phi = 4000$  (объем коррекции данных после расчета локального управления)

$k_{agr}^\Phi = 0.05$  (коэффициент агрегирования БД филиала)

$k_{cn}^\Phi = 0.1$  (относительное сокращение объема данных при выполнении запроса через сервер приложений)

$k_{\odot\odot}^\Phi = 30$  (отношение скорости записи к скорости чтения БД)

$k^{ADSL} = 5$  (отношение скоростей приема и передачи по сети ADSL)

$s_{cem}^{ADSL} = 1.53e-6$  (время передачи по сети ADSL)

$s_{cem}^\Phi = 2.38e-8$  (время передачи по сети филиала)

$s_{\odot\odot}^\Phi = 1.05e-5$  (время обработки записи СУБД филиала)

$s^\Pi = 1.66e-5$  (время обработки записи ЦП рабочей станции пользователя)

$s_{agr}^\Phi = 4.16e-6$  (время агрегирования записи)

$s_{cn}^\Phi = 2.00e-6$  (время предобработки записи на сервере БД)

$S_{agr}^\Phi = 0.007$  (время записи блока агрегированных данных)

$S_y^\Phi = 0.05$  (время записи блока локального управления)

$V_{абон}^{ADSL} = 2.10e+6$  (предоплаченный сетевой трафик)

$c^{ADSL} = 1.72e-5$  (плата за превышение сетевого трафика)



$$T_{ynp}^{\Phi} = 60 \text{ (допустимая задержка локального управления)}$$

$$c_{ynp}^{\Phi} = 0.29 \text{ (плата за задержку локального управления)}$$

$$T_{ynp}^{\Pi} = 1440 \text{ (допустимая задержка центрального управления)}$$

$$c_{ynp}^{\Pi} = 0.13 \text{ (плата за задержку центрального управления)}$$

$$T_{cp}^{\Pi} = 1 \text{ (допустимое время обработки запроса пользователя)}$$

$$c_{cp}^{\Pi} = 1000 \text{ (плата за задержку обработки запроса пользователя)}$$

При помощи встроенных возможностей Microsoft Excel 97 был получен следующий результат:

$$T^{\Phi} = 60$$

$$T^{\Pi} = 300$$

$$T_y^{\Pi} = 0$$

$$T_y^{\Phi} = 0$$

$$z = 0$$

$$C^{ADSL} + C_{ynp}^{\Phi} + C_{ynp}^{\Pi} + C_{cp}^{\Pi} = 0$$

Таким образом, функционирование будет производиться по следующему плану:

Каждые 60 минут (1 час) осуществляется агрегирование данных филиала. С периодичностью 360 минут (6 часов) агрегированные данные филиалов тиражируются в центр управления. С периодичностью 360 минут (6 часов) от центра управления к филиалам тиражируется глобальное управление. Локальное управление выполняется с периодичностью 60 минут (1 час). Вся обработка запросов пользователей осуществляется на сервере базы данных.

## СПИСОК ЛИТЕРАТУРЫ

1. Артемьев В.И. Обзор способов и средств построения информационных приложений. М.: Открытые системы, Системы управления базами данных, 1996, №5
2. Новиков И. Практика – критерий истины. М.: Открытые системы, Мир ПК, 2001, №9
3. Саймон Алан Р. Стратегические технологии баз данных. М.: Финансы и статистика, 1999
4. Кульба В.В., Ковалевский С.С., Косяченко С.А., Сиротюк В.О. Теоретические основы проектирования оптимальных структур распределенных баз данных. М.:Синтег, 1999
5. Гари Хансен, Джеймс Хансен Базы данных: Разработка и управление. М.: Бином, 2000
6. Гайфуллин Б.Н., Обухов И.А. Автоматизированные системы управления предприятиями стандарта ERP/MRP II. М.: Богородский печатник, 2000.
7. Отоцкий Л., Савин А. Семь критериев выбора ERP-систем для России. М.: Открытые системы, Открытые системы, 1998, №4
8. Когаловский В. Производственное планирование от Гантта до ERP. Что такое OPT, Just-in-time, CIM, CALS? Какова их связь с ERP и MRP II? М.: Открытые системы, Директор информационной службы, 2000, №5
9. Коржов В. Система ERP для малых и средних предприятий. М.: Открытые системы, Computerworld, 2000, №15
10. Ильина М. «Правильная» автоматизация для России. М.: Открытые системы, Computerworld, 2001, №15-16
11. Громько О. Осторожно: системы ERP. М.: Открытые системы, Открытые системы, 2001, №7-8

12. Олсон Майкл А. Выбор и реализация встроенных баз данных. М.: Открытые системы, Системы управления базами данных, 2000, №11

Редактор выпуска Глазкова Е.Н.  
Издательство ИМАШ РАН  
Тираж 100 экземпляров. 2002 год