Министерство сельского хозяйства Российской Федерации
Департамент научно-технологической политики и образования
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Красноярский государственный аграрный университет»

Н.В. Титовская

ХРАНИЛИЩА ДАННЫХ

Методические указания к лабораторным работам
Электронное издание

ФГОС ВО

Направление подготовки 09.04.03 «Прикладная информатика»
Направленность (профиль) «Цифровые технологии в АПК»

Курс: 1
Семестр: 1
Форма обучения: очная
Квалификация выпускника: магистр

Красноярск 2025

Рецензент

Н.А. Тод, к.э.н., доцент кафедры «Логистики» ИЭиУ АПК Красноярского ГАУ

Титовская, Н.В.

Хранилища данных [Электронный ресурс]: метод. указания к лабораторным работам/ Н.В. Титовская; Краснояр. гос. аграр. ун-т. – Красноярск, 2025. – 39 с.

Содержат задания на выполнение лабораторных работ по курсу «Хранилища данных», краткое содержание лабораторных работ, основные требования к оформлению отчета по выполненным лабораторным работам, рекомендуемую литературу, приложение с вариантами заданий.

Предназначено для магистрантов 1-го курса, обучающихся по направлению 09.04.03 «Прикладная информатика», направленность (профиль) «Цифровые технологии в АПК» (1-й семестр).

Печатается по решению редакционно-издательского совета Красноярского государственного аграрного университета

Оглавление

Введение 4
Системы поддержки принятия решений OLAP 5
Отличия хранилищ от обычных баз данных 6
Общие свойства хранилищ 10
Данные хранилища 14
Компоненты хранилища 17
Методика (методология) построения хранилищ данных 19
Лабораторная работа по реализации Хранилищ данных 28
Литература 39

Введение

Хранилища данных (Datawarehouse) и оперативный анализ данных (On-LineAnalyticalProcessing, OLAP) – новые информационные технологии, которые обеспечивают аналитикам, управленцам и руководителям высшего звена возможность изучать большие объемы взаимосвязанных данных при помощи быстрого интерактивного отображения информации на разных уровнях детализации с различных точек зрения в соответствии с представлениями пользователя о предметном пространстве.

Основная **цель** хранилищ данных (ХД) - создание единого логического представления данных, содержащихся в разнотипных базах данных (БД) или в единой модели корпоративных данных.

Другими словами хранилище данных создается с **целью** интеграции в одном месте, согласования и, возможно, агрегации ранее разъединенных детализированных данных, таких как:

- исторических архивов,
- данных из оперативных систем,
- данных из внешних источников, а также:
- разделения наборов данных, используемых для оперативной обработки, и наборов данных, используемых для решения задач поддержки принятия решений,
- обеспечения всесторонней информационной поддержки максимальному кругу пользователей.

Сегодня хранилища данных и OLAP становятся неотъемлемой частью современных корпоративных систем поддержки принятия решений. Это одно из наиболее динамично развивающихся направлений индустрии создания программного обеспечения.

Хранилище данных — это предметно-ориентированная, интегрированная, вариантная по времени, не разрушаемая совокупность данных, предназначенная для поддержки принятия управленческих решений.

Другое определение ХД :

Хранилище данных — ориентированная на поддержку управленческих решений автоматизированная система, состоящая из организационной структуры, технических средств, базы или совокупности базы данных (БД) и ПО, которое выполняет, как правило, следующие функции:

- извлечение данных из разрозненных источников, их трансформация и загрузка в хранилище;
- администрирование данных и хранилища;
- извлечение данных из хранилища, аналитическая обработка и представление данных конечным пользователям.

Ральф Кимбалл (Ralph Kimball), один из авторов концепции хранилищ данных, описывал хранилище данных как «место, где люди могут получить доступ к своим данным» (см., например, Ralph Kimball, «The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses», John Wiley & Sons, 1996 и «The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse», John Wiley & Sons, 2000). Он же сформулировал и основные **требования к хранилищам данных**:

- поддержка высокой скорости получения данных из хранилища;
- поддержка внутренней непротиворечивости данных;
- возможность получения и сравнения так называемых срезов данных (slice and dice);
- наличие удобных утилит просмотра данных в хранилище;
- полнота и достоверность хранимых данных;
- поддержка качественного процесса пополнения данных.

Системы поддержки принятия решений OLAP

Системы поддержки принятия решений обычно обладают средствами предоставления пользователю агрегатных данных для различных выборок из исходного набора в удобном для восприятия и анализа виде. Как правило, такие агрегатные функции образуют многомерный (и, следовательно, нереляционный) набор данных (нередко называемый гиперкубом или метакубом), оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные[1]. Вдоль каждой оси данные могут быть организованы в виде иерархии, представляющей различные уровни их детализации. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных.

Технология комплексного многомерного анализа данных получила название OLAP (On-Line Analytical Processing). OLAP — это ключевой компонент организации хранилищ данных. Концепция OLAP была описана в 1993 году Эдгаром Коддом, известным исследователем баз данных и автором реляционной модели данных (см. E.F. Codd, S.B. Codd, and C.T.Salley, Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Technical report, 1993). В 1995 году на основе требований, изложенных Коддом, был сформулирован так называемый тест FASMI (Fast Analysis of Shared Multidimensional Information — быстрый анализ разделяемой многомерной информации), включающий следующие требования к приложениям для многомерного анализа:

- предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;
- возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;
- многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;
- многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это — ключевое требование OLAP);
- возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

Следует отметить, что OLAP-функциональность может быть реализована различными способами, начиная с простейших средств анализа данных в офисных приложениях и заканчивая распределенными аналитическими системами, основанными на серверных продуктах

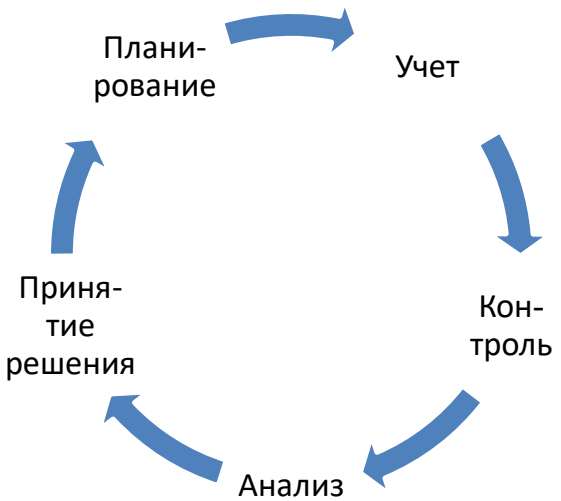
Отличия хранилищ от обычных баз данных

Типичное хранилище данных, как правило, отличается от обычной реляционной базы данных. Во-первых, обычные базы

данных предназначены для того, чтобы помочь пользователям выполнять повседневную работу, тогда как хранилища данных предназначены для принятия решений. Например, продажа товара и выписка счета производятся с использованием базы данных, предназначенной для обработки транзакций, а анализ динамики продаж за несколько лет, позволяющий спланировать работу с поставщиками, с помощью хранилища данных.

Во-вторых, обычные базы данных подвержены постоянным изменениям в процессе работы пользователей, а хранилище данных относительно стабильно: данные в нем обычно обновляются согласно расписанию (например, еженедельно, ежедневно или ежечасно — в зависимости от потребностей). В идеале процесс пополнения представляет собой просто добавление новых данных за определенный период времени без изменения прежней информации, уже находящейся в хранилище.

И в-третьих, обычные базы данных чаще всего являются источником данных, попадающих в хранилище. Кроме того, хранилище может пополняться за счет внешних источников, например статистических отчетов.



OLAP — это надстройка над OLTP и использует транзакционные системы в качестве источников данных.

В контуре управления взаимосвязаны 5 функций (по кольцу) :

- планирование
- учет
- контроль
- анализ
- принятие решений.

Рис. 2. Контур управления

Отмечают 2 типа контура:

- системы оперативной обработки транзакций
- системы класса поддержки принятия решений

Рис. 3. Распределение функционала между аналитическими и транзакционными информационными системами.

Развитие хранилищ данных обусловлено:

- созданием развитого ПО оперативного анализа данных и нерегламентированных запросов пользователей;
- появлением новых типов БД на основе многомерной модели и параллельной обработки запросов, которые опирались на достижения в области параллельных компьютеров;
- появлением ПО промежуточного слоя, обеспечившие связь между разнотипными БД;
- резким снижением стоимости хранения информации.

При перенесении данных из оперативной системы в хранилище перед загрузкой они преобразуются. Различного рода несоответствия в кодировании, типах данных и других «свойствах», присущих исходной системе, устраняются. Это также отличный повод для анализа данных исходной системы и приведения в соответствие всех расхождений реального состояния данных с их типами и кодами, представленными в документации. Вообще говоря, построение хранилища данных открывает возможность избавиться от нежелательных «свойств» оперативной системы.

Другим важным свойством, отличающим хранилище данных от оперативной системы, является то, что оно не разрушается. В то время как оперативная система выполняет над хранимыми данными операции обновления, удаления и вставки, в хранилище помещается большой объем данных, которые, будучи раз загруженными, уже никогда более не подвергаются каким-либо изменениям. Конечно,

редкие исключения из этого правила бывают. Характерной особенностью хранилища данных является то, что два разных корпоративных пользователя, выполняющие один и тот же запрос к хранилищу данных в разное время, получат один и тот же результат. Это исключает ситуации, при которых незапланированное извлечение данных и генерация отчетов приводят к различным результатам.

Еще одна особенность хранилища данных – независимость от времени. Если оперативная система содержит только текущие данные, то системы хранилищ данных содержат как исторические данные, так и данные, которые имели статус текущих при последней загрузке хранилища. Временные рамки данных, содержащихся в хранилище, изменяются в широких пределах в зависимости от типа системы. Однако обычно временные рамки данных, находящихся в хранилище, лежат в пределах от 15-ти месяцев до пяти лет. Данные большей давности, как правило, переносятся в архив на магнитной ленте или CDROM, если, конечно, их присутствие в хранилище данных больше не требуется.

Системы оперативных данных и информационные системы на основе хранилищ данных обладают рядом противоположных характеристик, которые лучше всего сравнивать непосредственно одну с другой. В таблице 1.1. приведен краткий перечень основных свойств систем каждого типа.

Таблица 1.1. Сравнительные характеристики хранилищ данных и оперативных систем

Системы хранилищ данных	**Оперативные системы**
Используются руководством	Используются работниками «переднего края»
Стратегическое значение	Тактическое значение
Поддерживают стратегические направления развития бизнеса	Поддерживают повседневную деятельность
Используются для интерактивного анализа	Используются для обработки транзакций
Предметно-ориентированные	Ориентированны на приложения
Хранят исторические данные	Хранят только текущие данные
Непредсказуемые запросы	Предсказуемые запросы

В настоящее время хранилища данных построены для столь большого числа предметных областей, что их невозможно здесь перечислить. Масштабы и способ использования этих хранилищ данных изменяются в широких пределах в зависимости от типа организации и вида деловой информации, для поддержки которых они разрабатывались. Вот некоторые из наиболее распространенных областей применения хранилищ данных.

- Анализ рисков.
- Финансовый анализ.
- Анализ случаев мошенничества.
- Маркетинг взаимоотношений.
- Управление активами.
- Анализ стереотипов поведения клиентов.

Общие свойства хранилищ

Хранилище данных играет в первую очередь роль интегратора и аккумулятора исторических данных. Структура организации хранилища ориентирована на предметные области. Предметно-ориентированное хранилище содержит данные, поступающие из различных оперативных БД и внешних источников. Хранилище представляет собой совокупность данных, отвечающую следующим характеристикам:

- ориентированность на предметную область или ряд предметных областей,
- интегрированность,
- зависимость от времени (поддержка хронологии),
- постоянство.

Ориентированность на предметную область

Первая особенность хранилища данных заключается в его ориентированности на предметный аспект. Предметная направленность контрастирует с классической ориентированностью прикладных приложений на функциональность и процессы.

Приложения всегда оперируют функциями, такими, например, как открытие сделки, кредитование, выписка накладной, зачисление на счет и т.д. Хранилище данных организовано вокруг фактов и

предметов, таких, как сделка, сумма кредита, покупатель, поставщик, продукт и т.д.

Интегрированность

Наиболее важный аспект хранилища данных состоит в том, что данные, находящиеся в хранилище, интегрированы.

Интегрированность проявляется во многих аспектах:

- в согласованности имен,
- в согласованности единиц измерения переменных,
- в согласованности структур данных,
- в согласованности физических атрибутов данных и др.

Контраст между интеграцией данных в хранилище данных и в прикладном окружении иллюстрируется следующим образом.

Первая причина возможного **рассогласования** приложений заключается в наличии множества средств разработки. Каждое средство разработки диктует определенные правила, часть из которых индивидуальна для данного средства. Не секрет, что каждый разработчик предпочитает одни средства разработки другим. Если два разработчика используют различные средства разработки, они, как правило, применяют индивидуальные особенности средств, а значит, возникает вероятность несогласованности между создаваемыми системами.

Вторая причина возможного **рассогласования** приложений заключается в существовании множества способов построения приложения. Способ построения конкретного приложения зависит от стиля разработчика, от времени, когда это приложение разрабатывалось, а также от ряда факторов, характеризующих конкретные условия разработки приложения. Все это отражается на используемых способах задания ключевых структур, способах кодирования, обозначения данных, физических характеристиках данных и т.д. Таким образом, если два разработчика создают различные способы построения приложений, имеется высокая вероятность того, что полной согласованности между системами не будет.

Интеграция данных по единицам измерения атрибутов состоит в следующем. Разработчики приложений к вопросу о способе задания размеров продукции могут подходить несколькими путями. Размеры могут задаваться в сантиметрах, дюймах, ядрах и т.д. Каков бы ни был источник данных, если информация поступит в хранилище, она

должна быть приведена к одним и тем же единицам измерения, принятым в качестве стандарта в хранилище.

Зависимость от времени

Все данные в хранилище в определенный момент времени **совместны** (непротиворечивы). Для оперативных систем эта базовая характеристика данных соответствует совместности данных в момент доступа. Когда в оперативной среде осуществляется доступ к данным, ожидается, что данные имеют совместные значения только в момент доступа к ним.

Зависимость от времени хранилища данных проявляется в следующем. Данные в хранилище представлены за временной промежуток от года до 10 лет. В оперативной среде представление данных осуществляется в промежутке от текущего значения до нескольких десятков дней. Приложения с высокой производительностью для обеспечения эффективного процесса транзакций должны работать с минимальным количеством данных. Следовательно, оперативные приложения ориентированны на короткий временной промежуток.

Другое проявление зависимости хранилища данных от времени заключается в его структуре. Каждая структура хранилища включает – явно или неявно – элемент времени.

Третье проявление зависимости хранилища данных от времени состоит в неукоснительном выполнении правила, что данные, однажды корректно в хранилище записанные, не могут быть обновлены. Хранилище данных с точки зрения практического использования представляет собой большую серию моментальных снимков. Естественно, если моментальный снимок данных был сделан некорректно, он может быть изменен. Но если был получен корректный моментальный снимок, то, однажды сделанный, он в последующем изменению не подлежит. Оперативные данные, будучи корректны в момент доступа к ним, могут обновляться по мере необходимости.

Постоянство

Четвертая определяющая характеристика хранилища данных – это постоянство. В оперативной среде операции обновления, добавления, удаления и изменения производятся над записями регулярно. Базовые манипуляции с данными хранилища ограничены начальной загрузкой данных и доступом к ним. В хранилище данных обновление данных не производится. Исходные (исторические)

данные, после того как они были согласованны, верифицированы и внесены в хранилище данных, остаются неизменными и используются исключительно в режиме чтения.

Существуют важные последствия различия обработки данных в оперативной среде и обработки в хранилище данных. На уровне проектирования хранилища данных необходимость в поддержке механизмов, обеспечивающих корректность обновлений, отпадает – обновления в хранилище данных не производятся. Это означает, что на физическом уровне проектирования при решении проблемы нормализации и физической денормализации доступ к данным может оптимизироваться без каких-либо ограничений. Другое последствие простоты работы с данными хранилища касается технологии работы с данными. Технология работы с данными в оперативной среде отличается большей сложностью. Она поддерживает функции оперативного резервного копирования и восстановления, обеспечивает целостность данных, включает механизмы разрешения конфликтов и тупиковых ситуаций. Для обработки информации в хранилище данных указанные функции не столь критичны.

Характеристики хранилища данных – ориентированность на предметную область при проектировании, интегрированность данных, зависимость от времени и простота управления данными – определяют среду, которая существенно отличается от классической транзакционной среды.

Источником почти всех данных среды хранилища данных являются оперативные среды. Может возникнуть ощущение, что существует огромная избыточность данных в обеих средах. Однако на практике избыточность данных в средах минимальна, поскольку:

- При передаче данных из оперативной среды в хранилище данных эти данные фильтруются. Многие данные вообще никогда не выгружаются из оперативной среды. В хранилище данных передается только информация, используемая для обработки в системе поддержки принятия решений.
- Временной горизонт в средах существенно различается. Данные в оперативной среде всегда являются текущими. Данные в хранилище имеют хронологию. С точки зрения временного горизонта пересечение между оперативной средой и средой хранилища данных минимально.
- Хранилище данных содержит агрегированные (итоговые) данные, которые никогда не включаются в оперативную среду.

- Передача данных из оперативной среды в хранилище данных сопровождается фундаментальными преобразованиями. Большинство данных при поступлении в хранилище видоизменяется.

Данные хранилища

В общем случае модель данных современных Систем Поддержки Принятия Решений (СППР) строится на основе **пяти классов данных**:

- источники данных,
- хранилища данных (в узком смысле),
- оперативный склад данных,
- витрины данных,
- метаданные.

Источники данных

Источниками данных хранилища служат оперативные транзакционные системы, которые обслуживают повседневную учетную деятельность компании. Необходимость включения той или иной транзакционной системы в качестве источника определяется бизнес-требованиями к СППР. Исходя из этих же требований, в качестве источников данных, могут быть рассмотрены внешние системы, в том числе и Интернет. Детальные данные из источников могут либо напрямую поступать в хранилище, либо предварительно агрегироваться до требуемого уровня обобщения.

Хранилище данных (в узком смысле)

Хранилище данных (в узком смысле) представляет собой предметно-ориентированную базу или совокупность БД, извлекаемых из источников, которые организованы по сегментам, отражающим конкретную предметную область бизнеса: производство, правило, детальные слабо агрегированные данные.

Оперативный склад данных (Operational Data Store - ODS)

В литературе существуют разные определения этого класса данных. В частности под оперативным складом данных можно подразумевать технологический элемент хранения данных в СППР, который служит буфером между транзакционными источниками данных и хранилищем. Как было уже отмечено ранее, данные, прежде чем попасть в хранилище, должны быть преобразованы в

единые форматы, очищены, объединены и синхронизированы. Например, данные, необходимые для поддержки принятия решения, могут существовать в транзакционной системе более короткое время (часы, дни), чем период пополнения данных хранилища (дни, недели). Или семантически однородные данные поступают из транзакционных систем в разное время. В этом случае оперативный склад данных служит аккумулятором данных, поступающих от источников, перед их загрузкой в хранилище. В отличие от хранилища данных информация в складе данных может изменяться со временем в соответствии с изменениями, происходящими в источниках данных.

Оперативный склад данных создается как промежуточный буфер между оперативными системами и хранилищем данных. Эта конструкция, аналогичная конструкции хранилища данных. Идентичность оперативного склада и хранилища данных состоит в их предметной ориентированности и хранении детальных данных. Отличие от хранилища данных состоит в том, что оперативный склад данных:

- имеет изменяемое содержимое,
- содержит только детальные данные,
- содержит текущие значения данных.

Детальные данные — это данные из оперативных и внешних систем, не подвергавшиеся операциям обобщения, суммирования, т.е. данные, не изменившие своей семантики. Из оперативных систем и внешних источников данные поступают в оперативный склад, проходя процессы трансформации.

Данные оперативного склада регулярно обновляются. Каждый раз, когда данные изменяются в оперативных системах и внешних источниках, соответствующие им данные из оперативного склада также должны быть изменены. Частота обновления оперативного склада зависит как от частоты обновления источников, так и от регламента загрузки данных в склад.

Витрины данных (Data mart)

Функционально ориентированные витрины данных представляют собой структуры данных, обеспечивающие решение аналитических задач в конкретной функциональной области или подразделении компании, например управление прибыльностью, анализ рынков, анализ ресурсов и проч. Иногда эти структуры хранения данных называют также киосками данных. Витрины

данных можно рассматривать как маленькие хранилища, которые создаются с целью информационного обеспечения аналитических задач конкретных управленческих подразделений компании.

Как правило, витрина содержит значительно меньше данных, охватывает всего несколько предметных областей и имеет более короткую историю. Витрины данных можно представить в виде логически или физически разделенных подмножеств хранилищ данных. Обычно они строятся для обслуживания нужд определенной группы пользователей.

Источником данных для витрин служат данные хранилища, которые, как правило, агрегируются и консолидируются по различным уровням иерархии. Детальные данные могут также помещаться в витрину или присутствовать в ней в виде ссылок на данные хранилища.

Различные витрины данных содержат разные комбинации и выборки одних и тех же детализированных данных хранилища. Важно, что данные витрины поступают из центрального хранилища данных — единого "источника истины".

Метаданные

Метаданные — это любые данные о данных. Метаданные играют важную роль в построении Систем Поддержки Принятия Решений (СППР). Одновременно это один из наиболее сложных и недостаточно практически проработанных объектов. В общем случае можно выделить по крайней мере три аспекта метаданных, которые должны присутствовать в системе.

1. **С точки зрения пользователей:**
    - метаданные для бизнес-аналитиков,
    - метаданные для администраторов,
    - метаданные для разработчиков.
2. **С точки зрения предметных областей:**
    - структуры данных хранилища,
    - модели бизнес-процессов,
    - описания пользователей,
    - технологические и пр.
3. **С точки зрения функциональности системы:**
    - метаданные о процессах трансформации,
    - метаданные по администрированию системы,

- метаданные о приложениях, метаданные о представлении данных
- пользователям.

Присутствие трех перечисленных аспектов метаданных подразумевает, что, например, прикладные пользователи и разработчики системы будут иметь различное видение технологических аспектов трансформации данных из источников: прикладные пользователи - семантику, состав и периодичность пополнения хранилища данными из источника, разработчики - ER-диаграммы, правила трансформации и интерфейс доступа к данным источника.

В настоящее время отсутствует единая промышленная технология проектирования, создания и сопровождения метаданных. Поэтому вопросы, связанные с управлением метаданными, рассматриваются отдельно, применительно к каждому конкретному проекту построения СППР.

Компоненты хранилища

Хранилище на самом верхнем уровне **состоит**, как правило, из трех подсистем:

- подсистемы загрузки данных,
- подсистемы обработки запросов и представления данных,
- подсистемы администрирования хранилища.

Подсистема загрузки данных

Данная подсистема представляет собой ПО, которое в соответствии с определенным регламентом извлекает данные из источников и приводит их к единому формату, определенному для хранилища. Данная подсистема отвечает за формализованную логическую согласованность, качество и интеграцию данных, которые загружаются из источников в оперативный склад данных. Каждый источник данных требует разработки собственного загрузочного модуля. Каждый модуль должен решать два класса задач:

- Начальной загрузки ретроспективных данных,
- Регламентного пополнения хранилища данными из источников.

Данная подсистема также по регламенту извлекает детальные данные из оперативного склада, производит их агрегирование, консолидацию, трансформацию и помещает данные в хранилище и

витрины данных. Именно в данной подсистеме должны быть определены все бизнес-модели консолидации данных по иерархическим измерениям и вычисления зависимых бизнес-показателей по независимым исходным данным.

Подсистема обработки запросов и представления данных

Оперативный склад, хранилище и витрины данных являются инфраструктурой, которая обеспечивает хранение и администрирование данных. Для извлечения данных, их аналитической обработки и представления конечным пользователям служит специальное ПО. Как правило, можно выделить три типа данного ПО:

- Программное обеспечение **регламентированной отчетности**, которое характеризуется заранее предопределенными запросами данных и их представлениями бизнес-пользователям. От данного ПО не требуется быстрого времени реакции. Из соображений стоимости эффективности для его реализации в наибольшей степени подходит технология **ROLAP** (см. далее).
- Программное обеспечение **нерегламентированных запросов** пользователей. Это ПО – основной способ общения бизнес-аналитиков с хранилищем, при котором каждый последующий запрос к данным и вид их представления определяются, как правило, результатами предыдущего запроса. Для приложений данного типа требуется высокая скорость обработки запросов (единицы секунд). Данное ПО реализуется технологией **MOLAP** (см. далее) и специальными инструментами построения сложных нерегламентированных запросов с интуитивно понятным для бизнес-аналитиков графическим интерфейсом.
- Программное обеспечение **добычи знаний**, которое реализует сложные статистические алгоритмы и алгоритмы искусственного интеллекта, предназначенные для поиска скрытых в данных закономерностей, представления этих закономерностей, представления этих закономерностей в виде моделей и многовариантного прогнозирования по ним развития ситуаций по схеме «Что если …?».

Конечно, как правило, такое деление носит весьма условный характер, а границы между соответствующими приложениями могут быть размыты [2].

Подсистема администрирования хранилища

К ведению данной подсистемы относятся все задачи, связанные с поддерживанием системы и обеспечением ее устойчивой работы и расширения. Можно выделить, по крайней мере, четыре класса задач, расширение которых должна обеспечивать данная подсистема:

- Администрирование данных, которое включает в себя регулярное пополнение данных из источников, если необходимо, ручной ввод, сверка и корректировка данных в оперативном складе. Администрирование данных ведется, как правило, бизнес-пользователями, а ответственность распределяется по предметно-ориентированным сегментам.
- Администрирование хранилища данных. В задачу администрирования хранилища входят все вопросы, связанные с поддержанием архитектуры хранилища, его эффективной и бесперебойной работы, защитой и восстановлением данных после сбоев.
- Администрирование доступа к данным обеспечивает сопровождение профилей пользователей, разграничение доступа к конфиденциальным данным, защиту информации от несанкционированного доступа.
- Администрирование метаданных системы.

Методика (методология) построения хранилищ данных

Существуют различные **подходы к стратегии построения** корпоративного хранилища данных (ХД):

- построение сверху вниз,
- снизу вверх,
- динамическая интеграция данных и др.

Считается, что **наиболее эффективным подходом является подход**, при котором в процессе разработки и внедрения хранилища данных осуществляется его пошаговое наращивание на основе единой системы классификаторов и общей среды передачи и хранения данных – **спиральная модель процесса разработки (рис. 4)**.

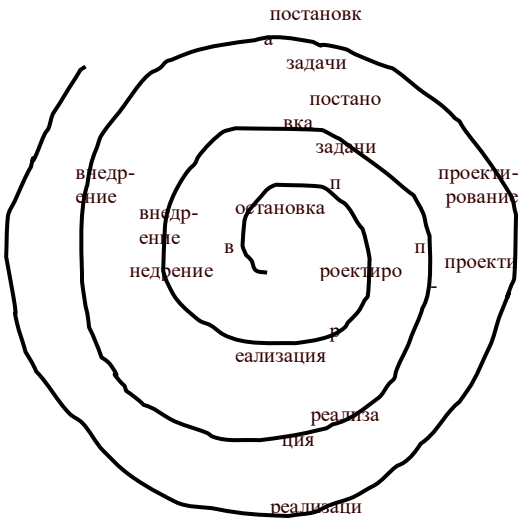


Рис. 4а. Спиральная модель разработки

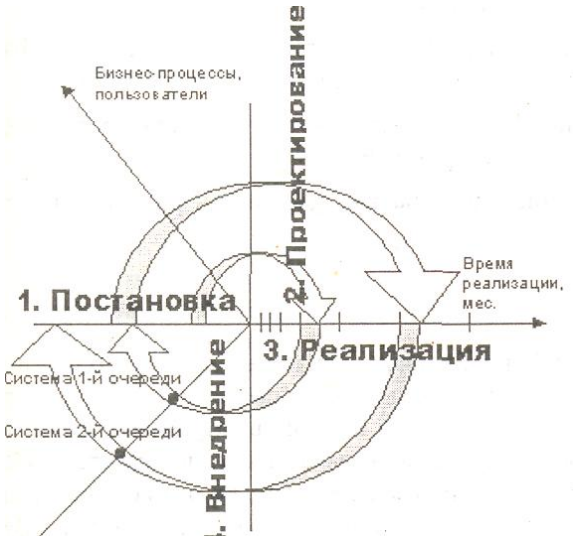


Рис. 4б. Стратегия построения СППР

На каждом шаге развертывания осуществляется реализация одной или ограниченного числа витрин данных по следующему технологическому циклу (стадиям создания):

- постановка задачи,
- проектирование,
- реализация,
- внедрение.

Стратегия пошагового наращивания позволяет по завершении каждого цикла ввести в кратчайшие сроки в промышленную эксплуатацию законченную систему, с определенной ограниченной функциональностью. Небольшие масштабы каждого проектного цикла существенно уменьшают потери при возможных проектных ошибках по сравнению с полномасштабным проектированием и созданием системы в целом. Кроме того, поскольку в каждом цикле применяются одни и те же методологические и технологические подходы, а также средства разработки, то время реализации каждой новой витрины будет сокращаться за счет повышения опыта проектной группы и постепенной отладки механизма взаимодействия между заказчиком и разработчиком системы.

Постановка задачи

Системно-аналитическое обследование

Этап обследования начинается с согласования и утверждения заказчиком плана и программы обследования. В процессе обследования выполняются следующие виды работ:

- проводятся интервью с основными участниками проекта со стороны компании-заказчика и лицами, ответственными за принятие управленческих решений;
- уточняется организационная структура, фиксируются организационные и функциональные рамки проекта;
- выявляются и документируются особенности и недостатки существующих информационных решений;
- формализуется схема бизнеса компании с учетом функциональных рамок;
- производится сбор существующих отчетных материалов и прочих официальных документов, имеющих непосредственное отношение к реализации проекта.

По итогам обследования уточняются стратегические и оперативные задачи управления компанией, решение которых должна обеспечивать СППР, формализуются цели и задачи создания системы. **Цель этапа анализа** – получение моделей данных и описание процедур принятия управленческих решений.

Техническое задание

Техническое задание (ТЗ) – один из ключевых документов проекта, который определяет требования к созданию СППР и порядок этого создания. Как правило, если время разработки системы превышает двенадцать месяцев, то целесообразно вводить очередность и, соответственно, сначала разрабатывать на основе концепции ТЗ систему первой очереди, которая может быть реализована за 3 месяца. В противном случае динамично развивающиеся условия бизнеса, постоянно совершенствующиеся информационные технологии приведут к тому, что, когда полномасштабная система будет реализована, она уже морально устареет. Если проект достаточно масштабен, то помимо основного ТЗ на систему в целом могут разрабатываться и частные ТЗ на ее отдельные компоненты.

Проектирование

На данной стадии проектных работ, на основе анализа требований к системе, сформулированных в ТЗ, разрабатываются основные архитектурные решения. **Архитектура информационной системы** рассматривается в четырех аспектах:

- Логическая архитектура. Представляет архитектуру системы с точки зрения пакетов базовых классов и их взаимосвязей. Определяются автоматизируемые процессы и функции,

необходимые для достижения поставленных целей, которые затем разделяются на задачи, подлежащие реализации на стадии разработки.

- Архитектура процессов. Применительно к СППР, определяет информационное обеспечение системы – состав и содержание процессов преобразования и передачи данных.
- Компонентная архитектура. Представляет архитектуру ПО системы, ее декомпозицию на подсистемы и компоненты.
- Техническая архитектура. Описывает физические узлы системы и связи между ними.

Автоматизируемые процессы и функции

Система Поддержки Принятия Решений (СППР) по виду автоматизированной деятельности относятся к системам обработки и передачи информации. **Объектами автоматизации** являются технические процессы, связанные с информационным обеспечением управленческой и аналитической деятельности руководящего персонала и специалистов подразделений и высшего руководства компании. **Целями системы** являются:

- **Интеграция** ранее разъединенных детализированных данных:
  - О исторических архивов,
  - О данных из оперативных систем,
  - О данных из внешних источников.
- **Разделение** наборов данных, используемых для оперативной обработки, и наборов данных, используемых для решения задач поддержки принятия решений.
- Обеспечение всесторонней информационной **поддержки** максимальному кругу исследователей.

Для реализации поставленных целей в рамках системы подлежат **автоматизации** следующие процессы:

- Сбор данных.
- Преобразование данных:
  1. Очистка данных.
  2. Согласование данных.
  3. Унификация данных.
  4. Агрегирование данных.
- Хранение данных:
  1. Промежуточное хранение данных.
  2. Накопление исторических данных.

- Предоставление данных потребителям.
- Сопровождение метаданных.

Информационное обеспечение

В общем случае **информационное обеспечение системы** состоит из **пяти классов данных:**

- источников данных,
- оперативного склада данных,
- хранилища данных,
- витрины данных,
- репозитария метаданных.

Проектирование информационного обеспечения системы осуществляется сверху вниз. На основе анализа прецедентов использования системы, выявленных на этапе системно-аналитического обследования, определяются представления данных конечным прикладным пользователям системы: состав показателей и их разрезы. Осуществляется сегментация представлений данных в соответствии с их проблемной ориентацией. На основе групп представлений витрин должны быть определены:

- **Измерения**, их иерархии и уровень детализации. Например, для временного измерения должен быть определен минимальный интервал времени (день, неделя, месяц), по которому будут индексироваться показатели в витрине.
- **Базовые показатели**, измерения, их индексирующие, и правила агрегирования каждого показателя по иерархиям. Правила агрегирования по иерархическому измерению зависят от показателя. Например, если для дохода от продаж агрегирование по времени осуществляется простым суммированием, то при исследовании цены продукции агрегирование по времени может быть реализовано в виде среднего, максимального или минимального значения за период агрегации.
- **Производные показатели** и формулы их вычисления на основе базовых показателей.

Выбор конкретного способа представления витрин (ROLAP, MOLAP или HOLAP — см. далее) выполняется, как правило, на стадии реализации системы.

Выявленные измерения и показатели служат исходными данными для проектирования хранилища.

В первую очередь обобщаются все выявленные разрезы и их иерархии. На их основе проектируется бизнес-пространство хранилища. Измерения, как правило, тесно связаны со структурированной нормативно-справочной информацией компании. Например, измерениями хранилища часто служат организационная структура компании, справочник административно-территориального деления, план финансовых статей компании и пр.

На пространстве, которое задается бизнес-измерениями, проектируются базовые и производные показатели, которые должны находиться в хранилище. Для больших систем целесообразно проводить сегментацию хранилища по предметным областям.

На следующем этапе выполняется анализ результатов обследования источников данных. При выборе подходящего источника во внимание принимаются следующие вопросы:

- Если имеется более одного источника, следует ли определить, какой из них лучше?
- Какие преобразования необходимо выполнить, чтобы приготовить источник к загрузке в хранилище?
- Согласуются ли структура источника и структура хранилища?
- Насколько согласуются данные источника с нормативно-справочной информацией?
- Что будет происходить, если источник имеет несколько месторасположений?
- Насколько аккуратны данные источника?
- Как источник обновляется?
- Каковы возраст и перспективность источника?
- Насколько полны данные?
- Что потребуется для интеграции данных источника в поток загрузки?
- Какова технология хранения данных в источнике?
- Насколько эффективно может осуществляться доступ к источнику?

На основе выполненного анализа принимаются следующие **архитектурные решения:**

- Определяются состав, содержание и источники потоков данных, которые будут поступать из источников в хранилище.

- Определяются преобразования, которые должны быть выполнены над данными при загрузке, а также периодичность загрузки данных в хранилище.
- При необходимости проектируются структуры оперативного склада данных и транзитных файлов.
- Выявляются данные, которые отсутствуют в источниках информационного хранилища. Для таких данных, как правило, проектируются процедуры и регламенты ручного ввода.

Общая структура репозитария хранилища является своего рода отражением главной цели его построения, а именно максимально полно и быстро удовлетворить потребности пользователей в той или иной информации. В зависимости от потребностей пользователей в информации можно выделить следующие ее **основные типы**:

- **Персональную информацию** – эта информация, используемая пользователями со строго определенными обязанностями и информационными потребностями. Обычно требует большой предварительной обработки, или, другими словами, имеет высокий уровень агрегации. Чаще всего храниться в МБД.
- **Информацию по бизнес-темам** – информация, относящаяся к определенной тематике, например, как финансовая деятельность организации. Для организаций имеющих близкие функциональные и организационные структуры, ее можно определить как информацию для подразделения (например, для финансовой службы). Имеет более широкий спектр, как в предметных областях, так и вовремени, но вместе с тем напрямую используется реже, чем персонализированная информация. Обычно храниться в смешанных структурах: МБД и реляционных таблицах.
- **Детальные данные** – самая подробная информация, доступная в хранилище данных. Обычными пользователями применяется весьма редко, только в случае необходимости подробного уточнения информации. Обычно является полем деятельности аналитиков по добыче знаний (или поиску скрытых зависимостей в больших объемах информации). Обычно храниться в реляционных структурах.

- Старые детальные данные – это, по сути, тот же самый низкий уровень агрегирования, что и у текущих детальных данных, - выделяются в особый тип по следующей причине. С одной стороны, старые детальные данные часто требуют больших ресурсов для хранения, а с другой – они со временем, например, через несколько лет, необходимы очень редко. Решением в данном случае является использование более дешевых и емких способов хранения, например лент или библиотек.

Компонентная архитектура

Система на самом верхнем уровне состоит, как правило, из двух видов ПО: общего и специального.

К общему ПО относятся:

- ***ПО промежуточного слоя***, которое обеспечивает сетевой доступ к приложениям и БД. Сюда относятся сетевые и коммуникационные протоколы, драйверы, системы обмена сообщениями и пр.
- ***ПО загрузки и предварительной обработки данных***. Этот уровень включает в себя набор средств для загрузки данных из OLTP-систем и внешних источников. Проектируется, как правило, в сочетании с дополнительной обработкой: проверкой данных на чистоту, консолидацией, форматированием, фильтрацией и пр.
- ***Серверное ПО***. Представляет собой ядро всей системы. Оно включает в себя:
    - Серверы реляционных БД,
    - Серверы МБД,
    - Серверы приложений (поисковые, аналитической обработки, добычи знаний и др.).
    - **Специальное ПО** представляет собой совокупность программ, разрабатываемых при создании Систем Поддержки Принятия Решений (СППР). Они объединяются в следующие **подсистемы:**
    - Подсистему загрузки данных,
    - Подсистему обработки запросов и представления данных,
    - Подсистему администрирования.

В этой части должны быть спроектированы модули, составляющие подсистему, и алгоритмы отдельных процедур, входящих в их состав.

Техническая архитектура

Серверное ПО работает под управлением серверов приложений и серверов БД на UNIX- или NT-платформах или мэйнфреймах. **Клиентское ПО**, устанавливается на ПК конечных пользователей. В последние годы наметилось стремительное внедрение технологии «тонкого» клиента, при которой на ПК пользователя находится лишь Web-броузер, а вся функциональность клиентского ПО загружается с сервера приложений в виде JavaScript- программ или апплетов. Техническая архитектура во многом зависит от масштабови требований, предъявляемых к ее производительности и надежности. В зависимости от этого серверные компоненты системы могут располагаться на одном компьютере или на нескольких. Сегменты хранилища и витрины данных в больших системах могут располагаться на нескольких компьютерах.

Реализация

Данная стадия проекта непосредственно связана с разработкой и тестированиемкомпонентов информационного и специального ПО системы в соответствии с разработанной на этапе проектирования архитектурой.

К основным результатам работы на этом этапе следует отнести:

- Непосредственно саму систему в виде общего и специального ПО, баз данных.
- План внедрения системы, который должен определять все работы по внедрению системы у заказчика, включая упаковку системы, доставку ее заказчику, инсталляцию системы на технических средствах заказчика, тестирование и доработку.
- Набор тестов, которые должны быть выполнены после установки системы у заказчика.
- Пользовательскую документацию и учебные материалы для пользователей системы.

Внедрение

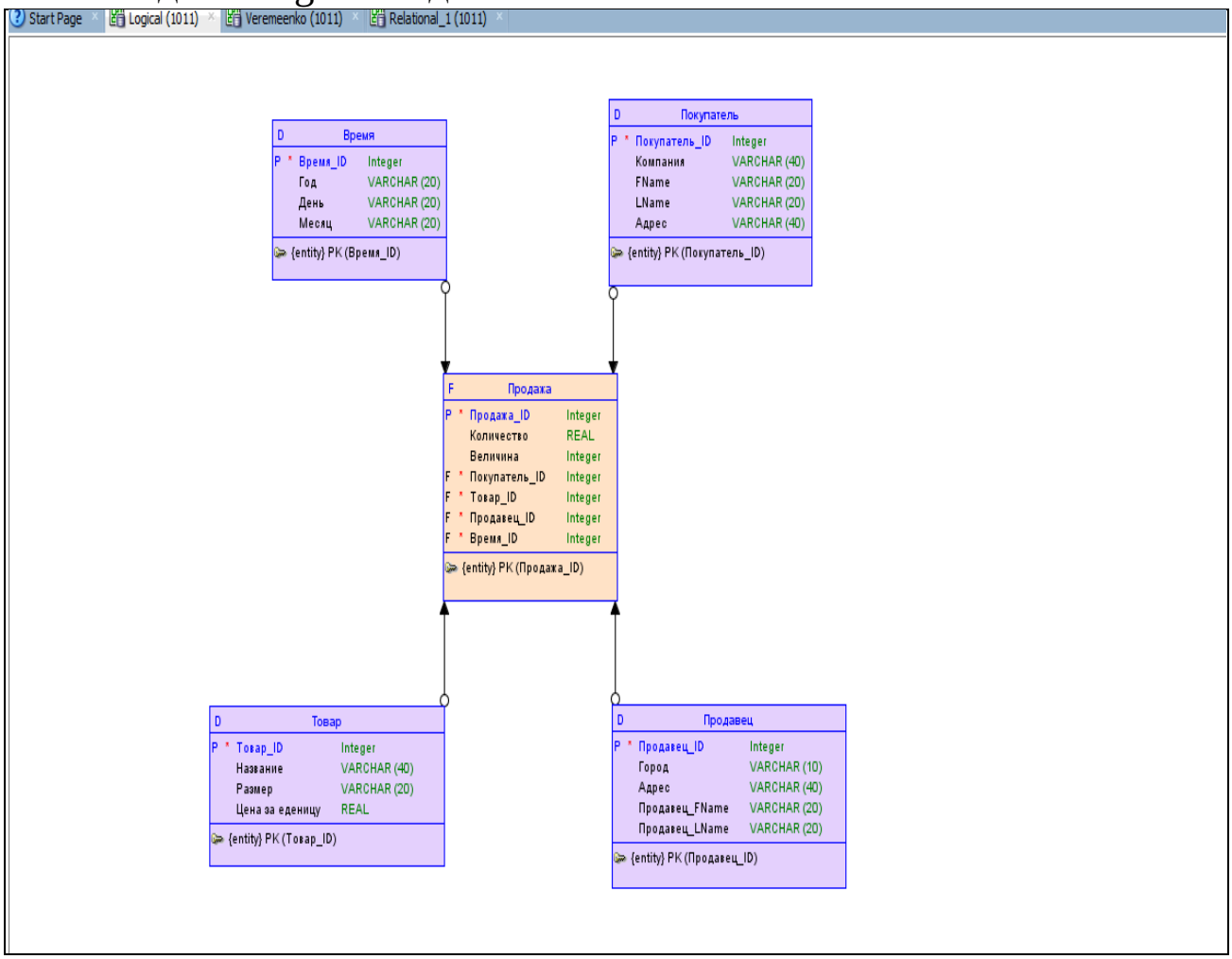
Данная фаза состоит в выполнении работ, предусмотренных планом внедрения, который был разработан на предыдущей фазе.

На стадии развертывания осуществляются монтаж и установка системы и отдельных ее компонентов у заказчика. Осуществляется первоначальная загрузка хранилища необходимыми данными, выполняется опытная эксплуатация системы. Кроме того, на стадии развертывания осуществляется обучение пользователей и

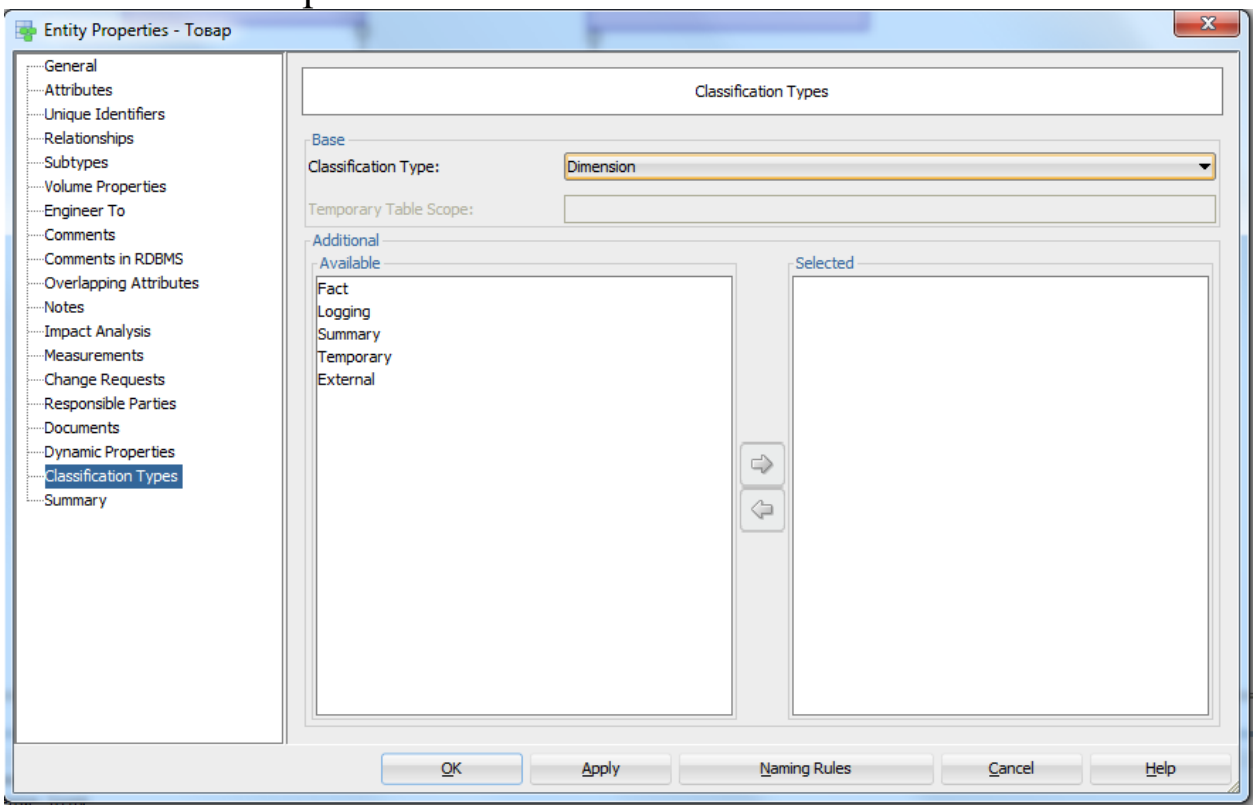
сотрудников службы технической поддержки. Окончанием данного этапа считается момент перехода к производственной эксплуатации хранилища.

Лабораторная работа по реализации Хранилищ данных

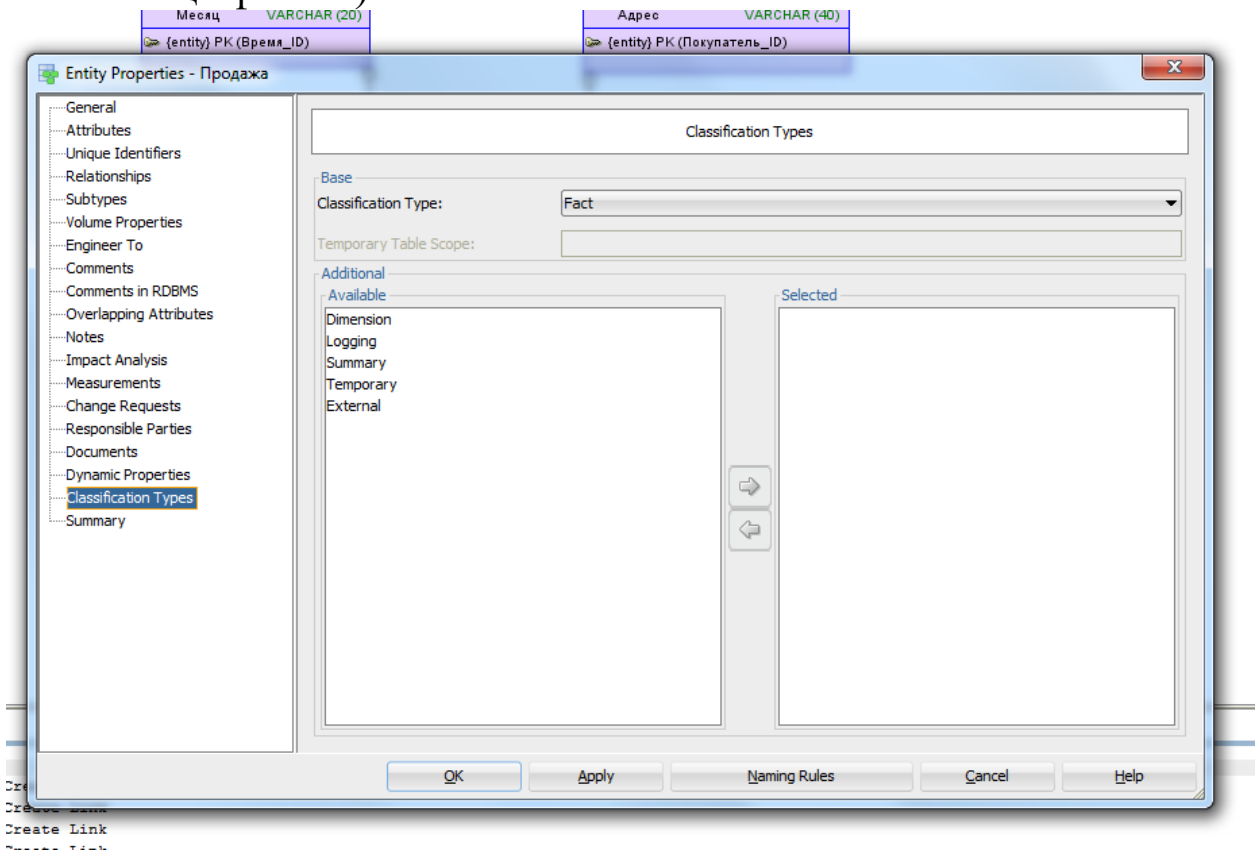
1. Создаем Logical модель в Datamodeler.



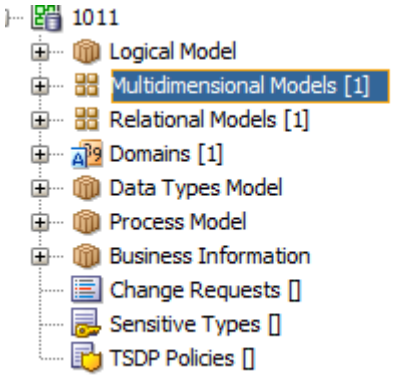
2. Сущностям Товар, Продавец, Покупатель, Время properties – выставляем настройки:



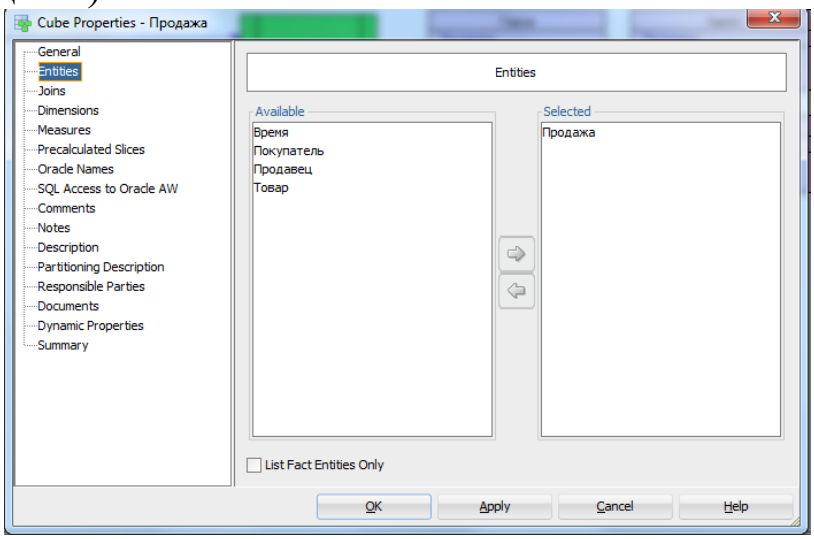
Для сущности Продажа выбираем Classification type – Fact.(т.к это таблица фактов)



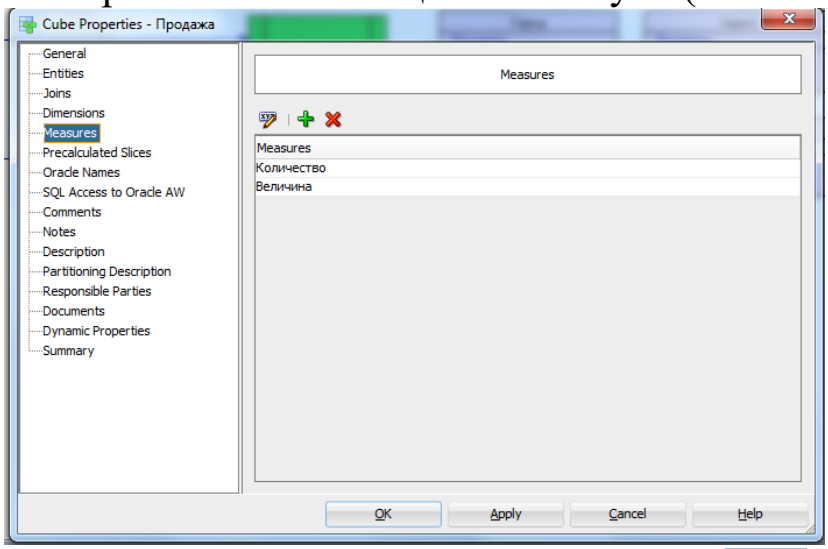
3. Создаем новую multidementional model (МП - new multidementional model)



4. Переходим в новую, созданную нами, многомерную модель
5. На панели инструментов выбираем NEW CUBE. 
   Называем его ПРОДАЖИ
6. Entities – выбираем сущность с которой связываем куб (Продажи)

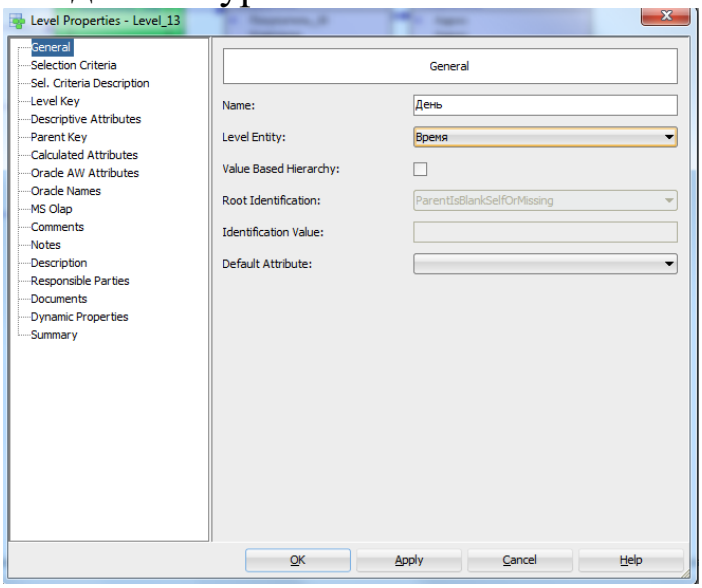


7. Выбираем состовляющие эл-ты куба (MEASURES).

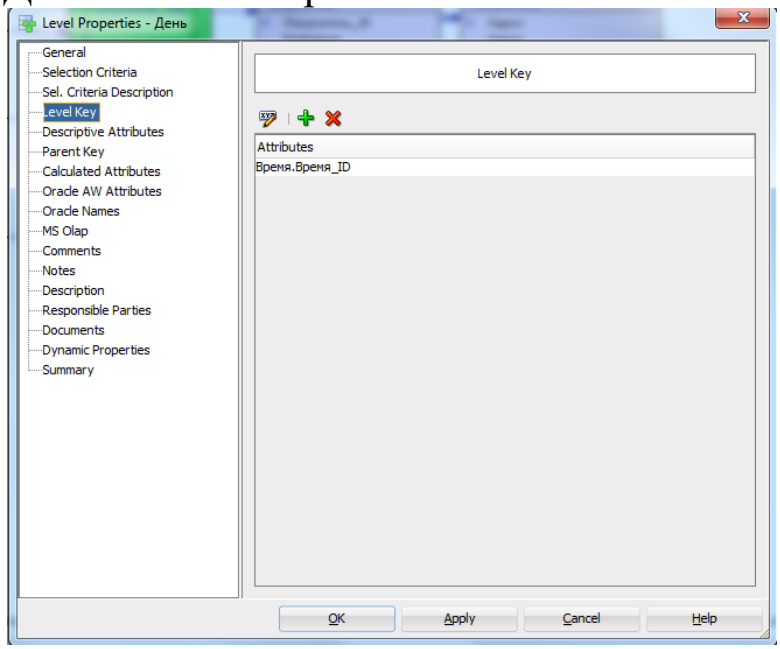


8. Создаем уровень измерения New Level 

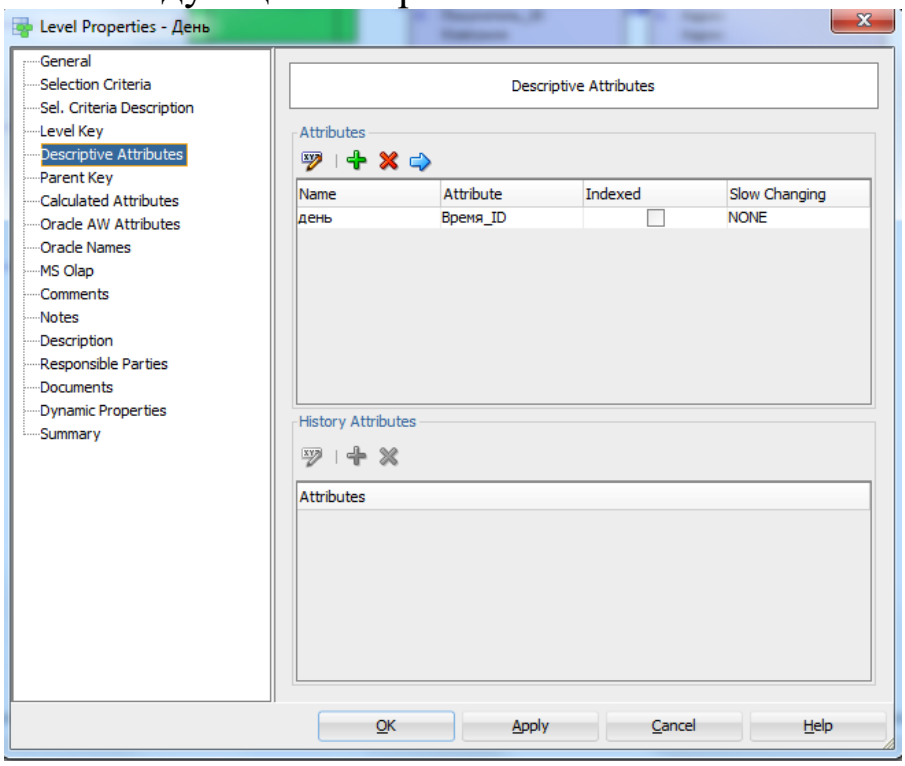
Назовем его День. Выставляем следующие настройки. В поле level entity выбираем таблицу измерений от которого будет зависеть данный уровень.



9. Добавляем первичный ключ таблицы Level key -



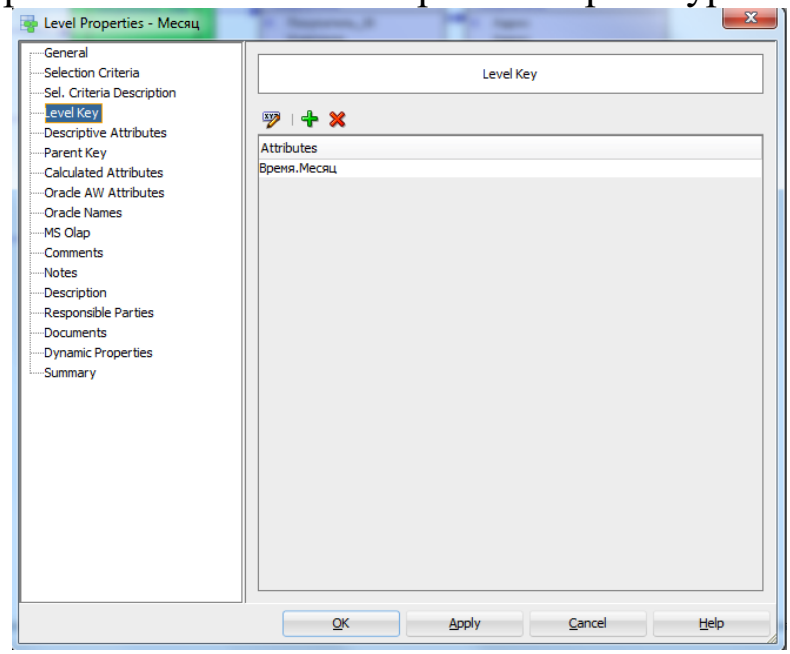
10. Следующая настройка



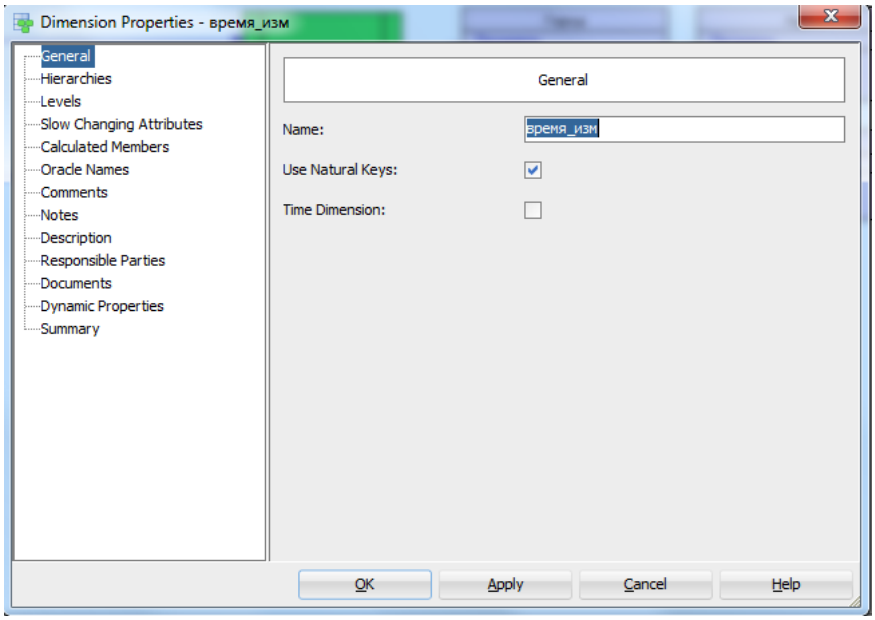
11. Получаем следующий уровень:

12. По тому же принципу создаем другие уровни того же измерения (месяц и год).

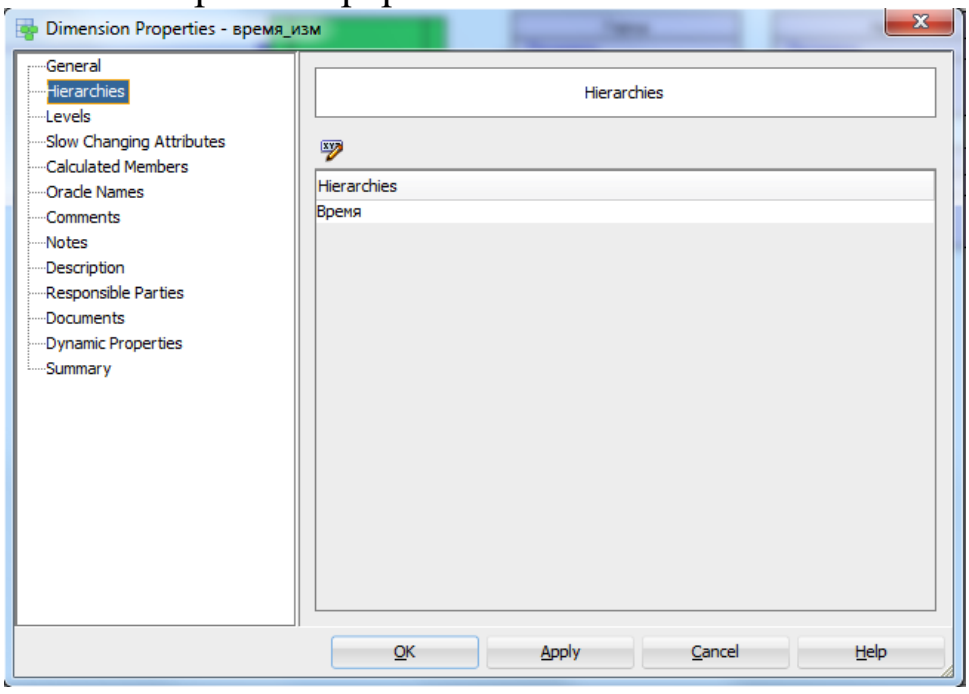
Настройки будут отличаться в разделе Level Key,т.к первичный ключ мы выбрали в первом уровне.



13. Создаем измерение New dimension 
Назовем его время_изм

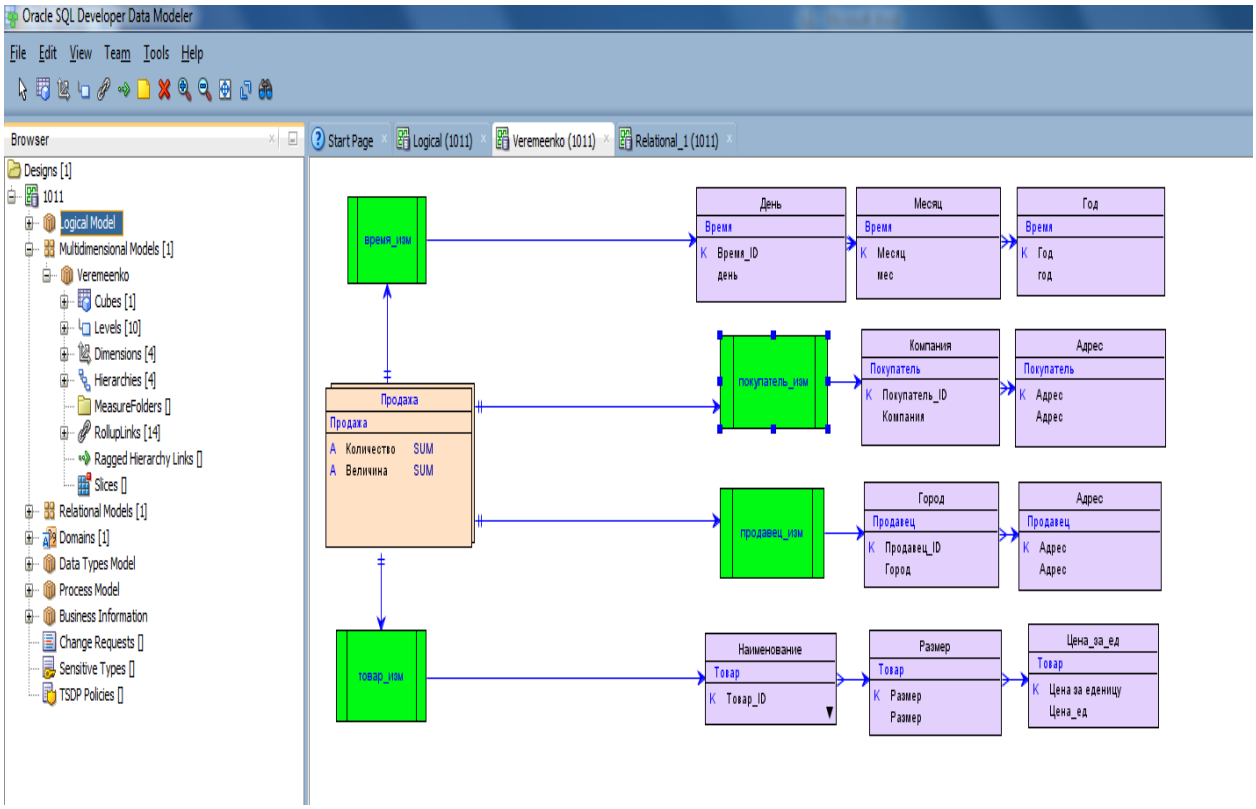


14. Выбираем Иерархию

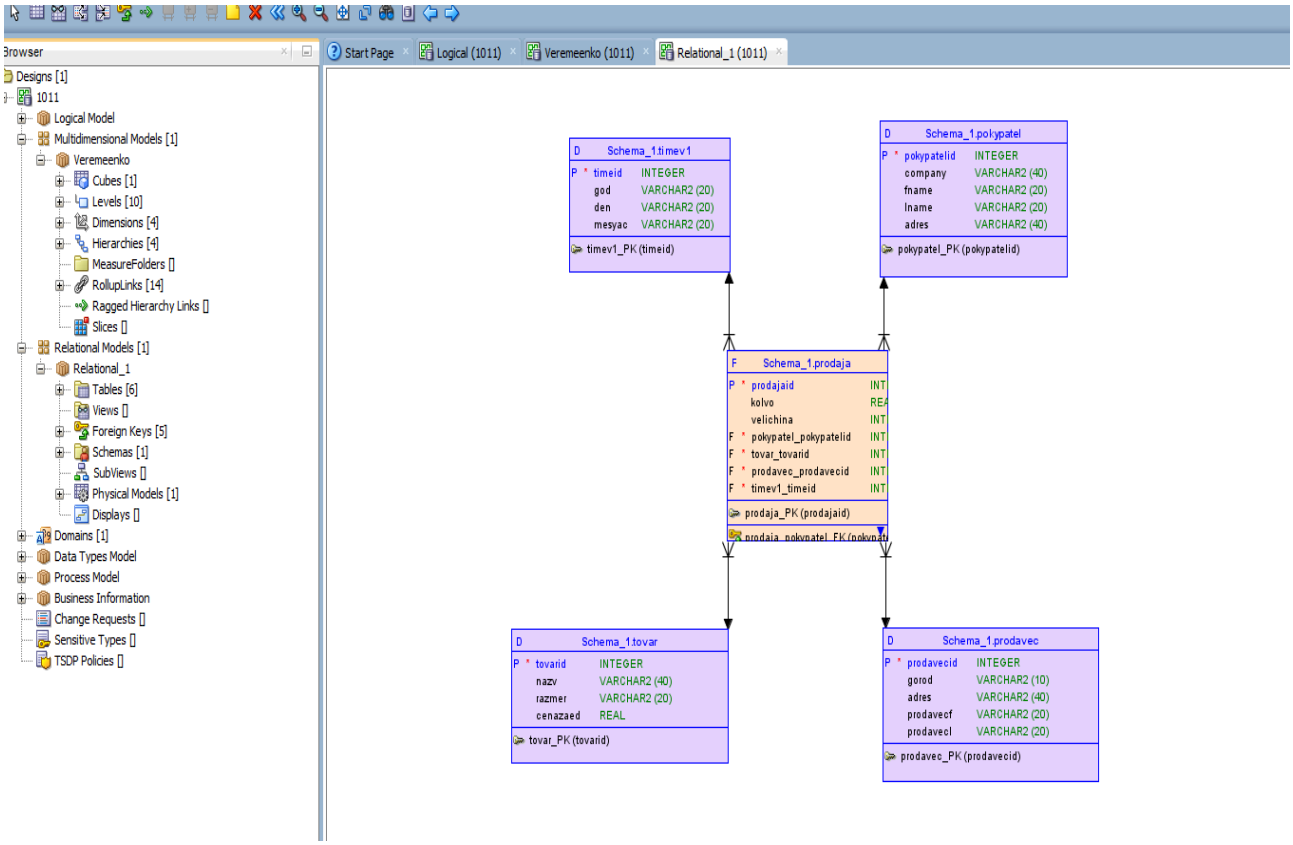


15. С помощью элемента New link  связываем созданные нами элементы.

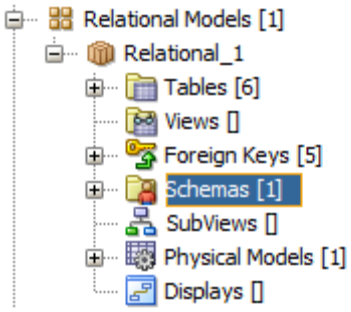
16. По тому же принципу создаем все остальные измерения и связи



17. Создаем из логической реляционную модель с помощью кнопки  . Engineer

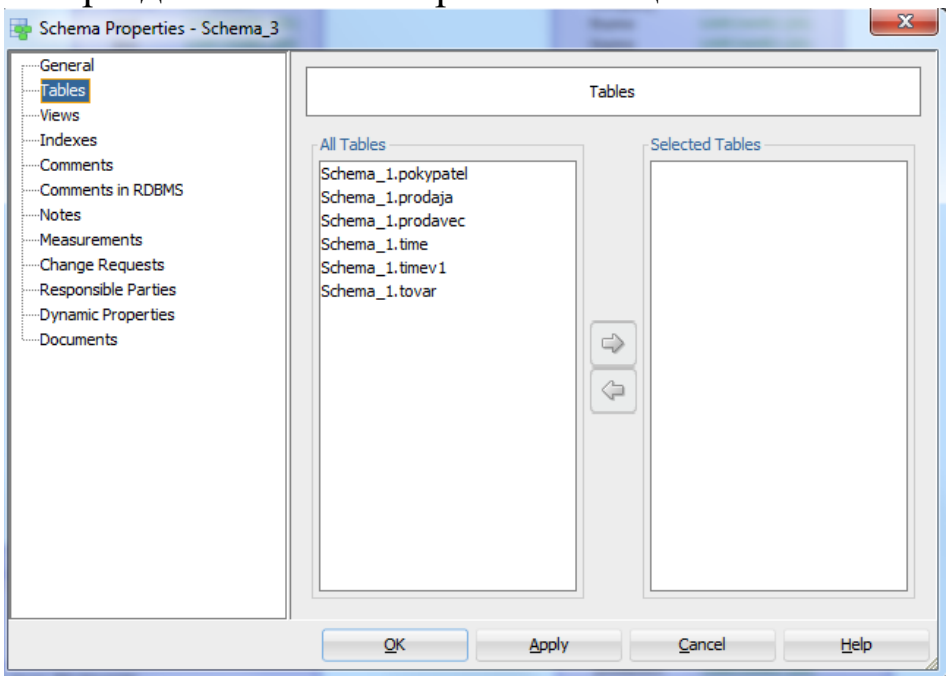


18. В окне Browser создаем новую схему

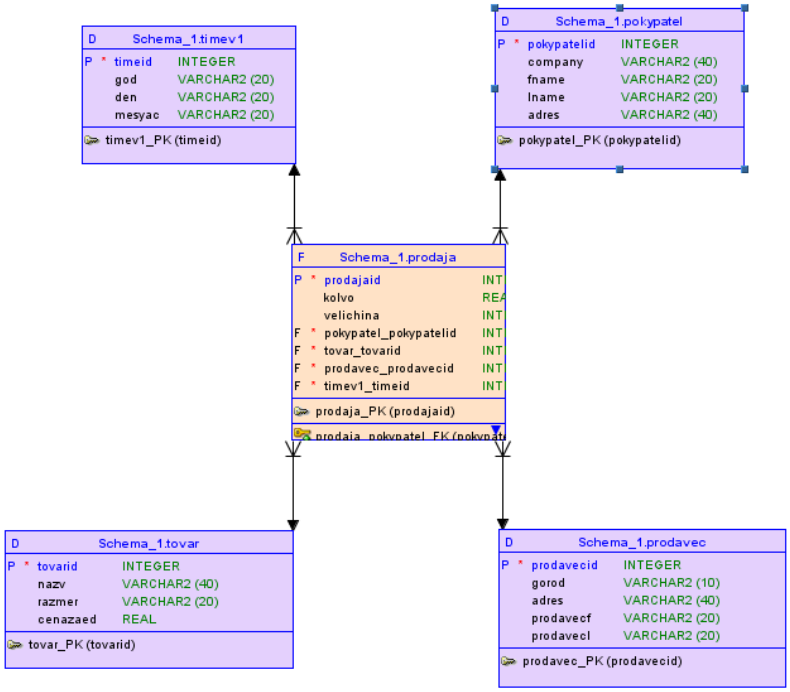


Schemas – МП – New schema

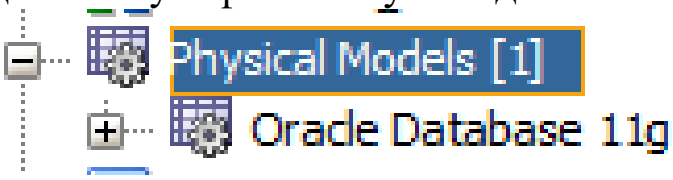
19. В разделе tables выбираем таблицы



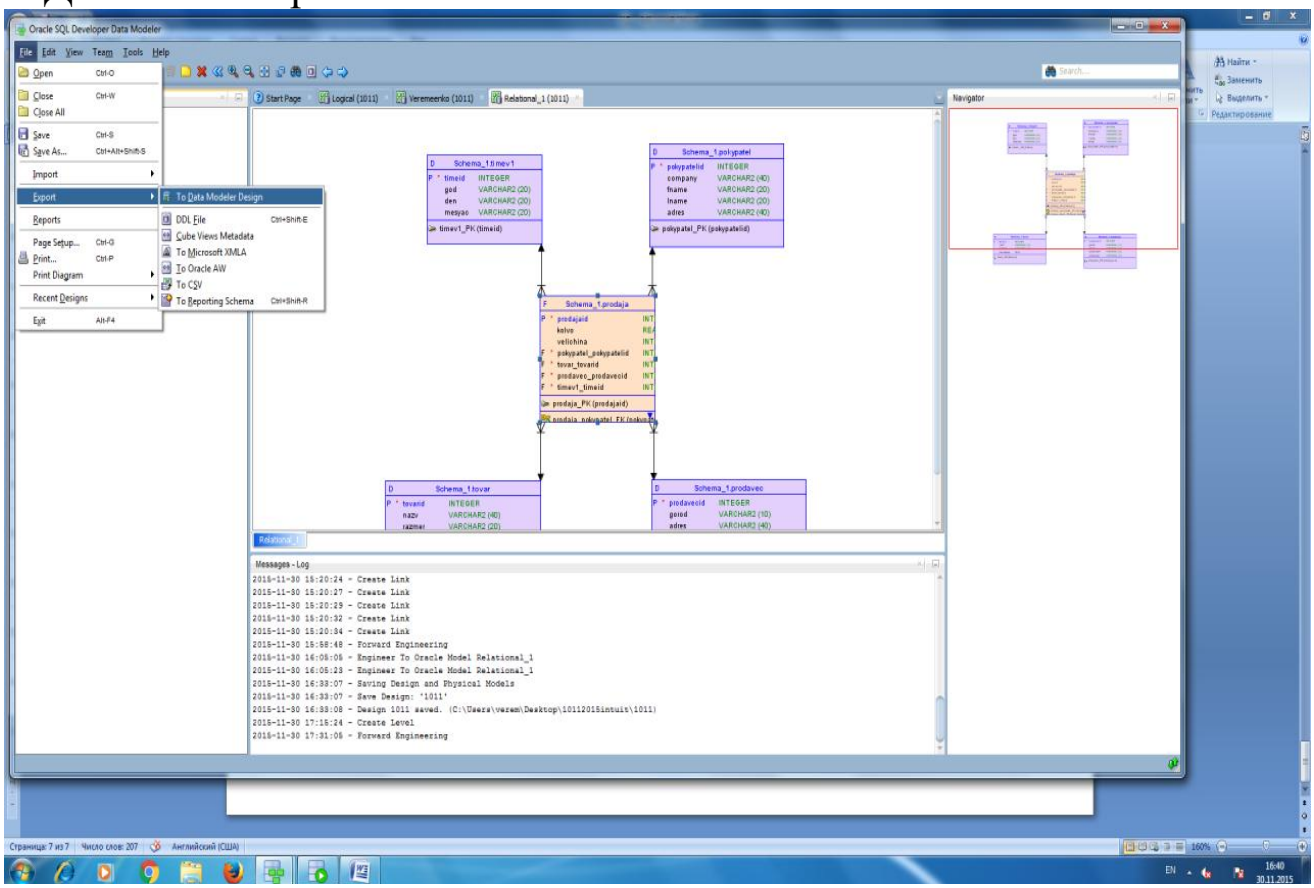
20. Получаем следующий вид



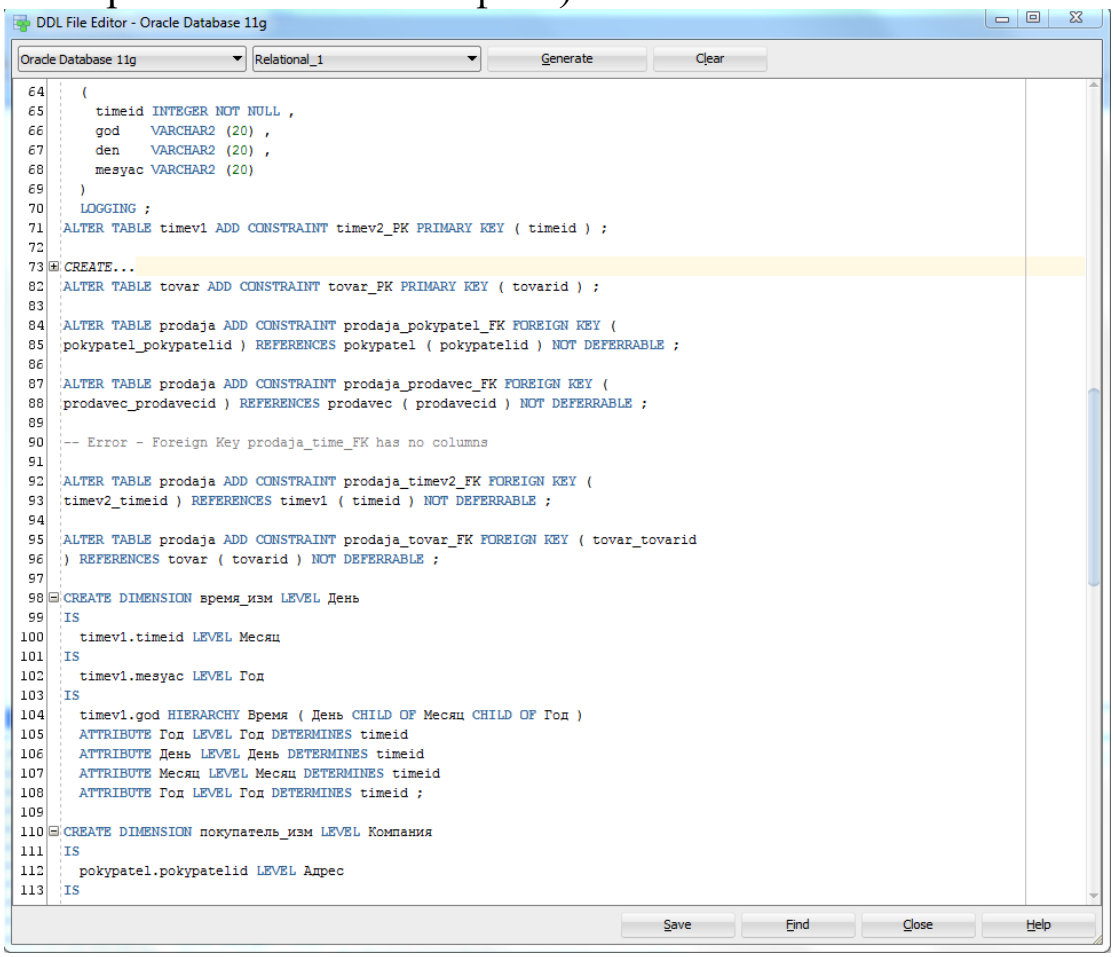
21. Создаем новую физическую модель.New physical model



22. Делаем экспорт DDL file

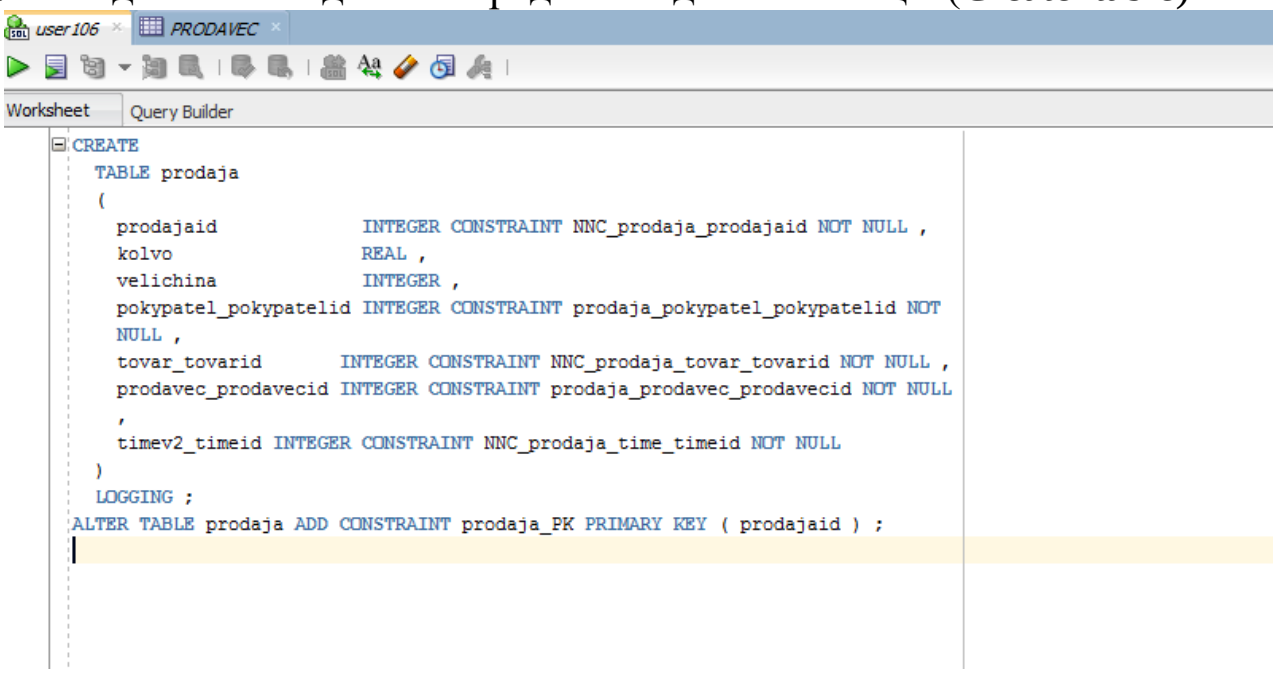


23. Сохраняем код Save DDL file (Для себя рекомендуем скопировать в текстовый файл)




```
  (
    timeid INTEGER NOT NULL ,
    god    VARCHAR2 (20) ,
    den    VARCHAR2 (20) ,
    mesyac VARCHAR2 (20)
  )
  LOGGING ;
ALTER TABLE timev1 ADD CONSTRAINT timev2_PK PRIMARY KEY ( timeid ) ;

CREATE...
ALTER TABLE tovar ADD CONSTRAINT tovar_PK PRIMARY KEY ( tovarid ) ;

ALTER TABLE prodaja ADD CONSTRAINT prodaja_pokypatel_FK FOREIGN KEY (
pokypatel_pokypatelid ) REFERENCES pokypatel ( pokypatelid ) NOT DEFERRABLE ;

ALTER TABLE prodaja ADD CONSTRAINT prodaja_prodavec_FK FOREIGN KEY (
prodavec_prodavecid ) REFERENCES prodavec ( prodavecid ) NOT DEFERRABLE ;

-- Error - Foreign Key prodaja_time_FK has no columns

ALTER TABLE prodaja ADD CONSTRAINT prodaja_timev2_FK FOREIGN KEY (
timev2_timeid ) REFERENCES timev1 ( timeid ) NOT DEFERRABLE ;

ALTER TABLE prodaja ADD CONSTRAINT prodaja_tovar_FK FOREIGN KEY ( tovar_tovarid
) REFERENCES tovar ( tovarid ) NOT DEFERRABLE ;

CREATE DIMENSION время_изм LEVEL День
IS
  timev1.timeid LEVEL Месяц
IS
  timev1.mesyac LEVEL Год
IS
  timev1.god HIERARCHY Время ( День CHILD OF Месяц CHILD OF Год )
  ATTRIBUTE Год LEVEL Год DETERMINES timeid
  ATTRIBUTE День LEVEL День DETERMINES timeid
  ATTRIBUTE Месяц LEVEL Месяц DETERMINES timeid
  ATTRIBUTE Год LEVEL Год DETERMINES timeid ;

CREATE DIMENSION покупатель_изм LEVEL Компания
IS
  pokypatel.pokypatelid LEVEL Адрес
IS
```

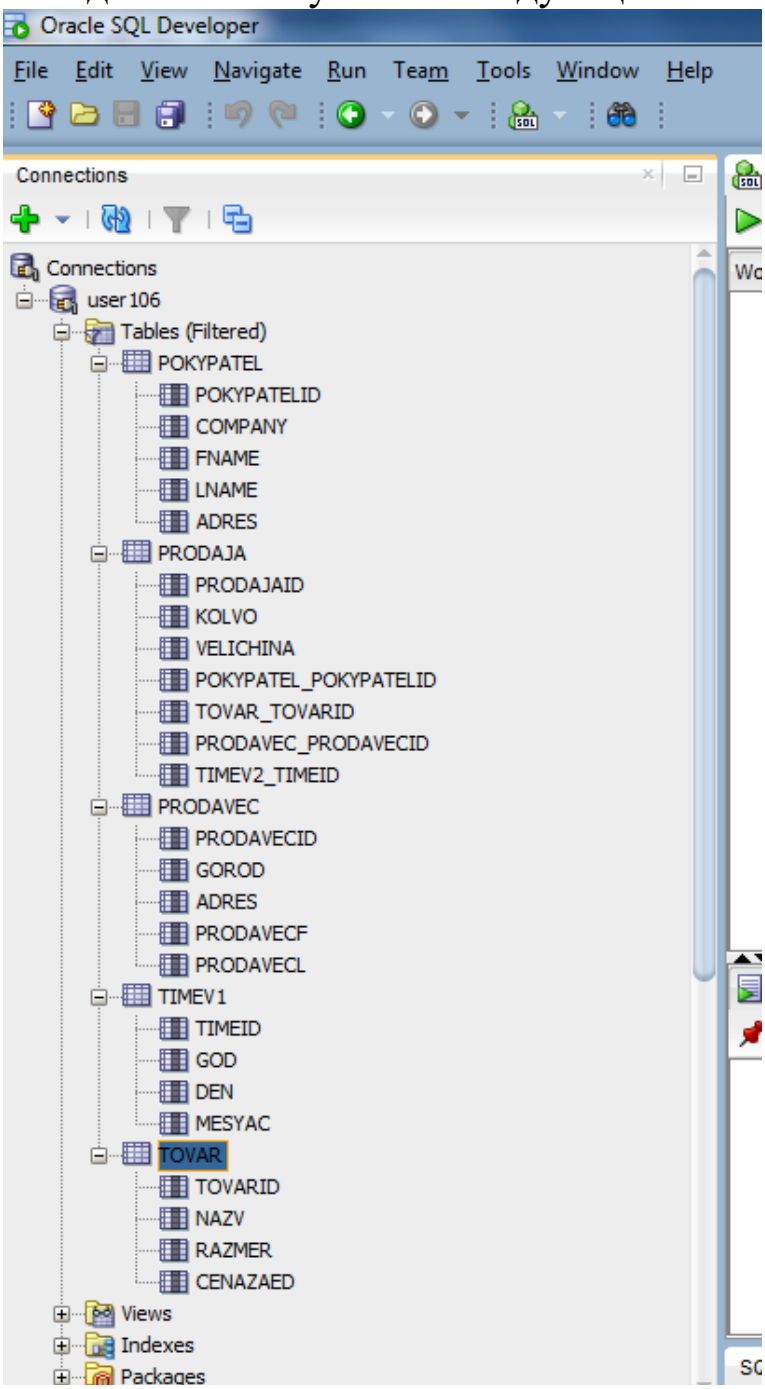
24. Запускаем ORACLE
25. Из данного кода поочередно создаем таблицы.(Create table)




```
CREATE
  TABLE prodaja
  (
    prodajaid             INTEGER CONSTRAINT NNC_prodaja_prodajaid NOT NULL ,
    kolvo                 REAL ,
    velichina             INTEGER ,
    pokypatel_pokypatelid INTEGER CONSTRAINT prodaja_pokypatel_pokypatelid NOT
    NULL ,
    tovar_tovarid       INTEGER CONSTRAINT NNC_prodaja_tovar_tovarid NOT NULL ,
    prodavec_prodavecid INTEGER CONSTRAINT prodaja_prodavec_prodavecid NOT NULL
    ,
    timev2_timeid INTEGER CONSTRAINT NNC_prodaja_time_timeid NOT NULL
  )
  LOGGING ;
ALTER TABLE prodaja ADD CONSTRAINT prodaja_PK PRIMARY KEY ( prodajaid ) ;
```

26. По такому же принципу преобразуем весь оставшийся код в Хранилище данных.

27. В итоге должны получаться следующие таблицы



Литература

Основная

1. Парфенов, Ю. П. Постреляционные хранилища данных : учебное пособие для вузов / Ю. П. Парфенов ; под научной редакцией Н. В. Папуловской. — Москва : Издательство Юрайт, 2024. — 121 с. —https://urait.ru/bcode/472624
2. Орешков, В. И. Хранилища данных и OLAP-технологии : учебное пособие / В. И. Орешков. — Рязань : РГРТУ, 2023. — 64 с. — https://e.lanbook.com/book/16798

Дополнительная литература

3. . Акопов, А. С. Имитационное моделирование : учебник и практикум для вузов / А. С. Акопов. — Москва : Издательство Юрайт, 2024. — 389 с. https://urait.ru/bcode/468919

Методические указания, рекомендации и другие материалы к занятиям

Титовский С.Н., Титовская Н.В.Хранилища данных. Электронный обучающий ресурс. http://e.kgau.ru/course/view.php?id=1059