**Лекция 12:**

**Создание модели хранилища данных на основе корпоративной модели данных**

**Цель лекции**

Изучив материал настоящей лекции, вы будете знать:

* что такое *корпоративная модель данных* ;
* как преобразовать *корпоративную модель данных* в модель хранилища данных;
* основные элементы *корпоративной модели данных* ;
* *уровни представления корпоративной модели данных* ;
* *алгоритм преобразования корпоративной модели данных в многомерную модель хранилища данных* ;

и научитесь:

* разрабатывать модели хранилища данных на основе *корпоративной модели данных* организации;
* разрабатывать схему "звезда" с помощью CASE-средств;
* секционировать таблицы *многомерной модели* с помощью CASE-средств.

**Литература**: [[2](https://intuit.ru/studies/courses/599/455/literature#literature.2)], [[27](https://intuit.ru/studies/courses/599/455/literature#literature.27)].

**Корпоративная модель данных**

**Введение**

Ядром любого ХД является его модель данных. Без модели данных будет очень сложно организовать данные в ХД. Поэтому разработчики ХД должны потратить время и силы на разработку такой модели. Разработка модели ХД ложится на плечи проектировщика ХД.

По сравнению с проектированием OLTP-систем, методика проектирования ХД имеет ряд отличительных особенностей, связанных с ориентацией структур данных хранилища на решение задач анализа и информационной поддержки процесса принятия решений. Модель данных ХД должна обеспечивать эффективное решение именно этих задач.

Отправной точкой в проектировании ХД может служить так называемая *корпоративная модель данных* (*corporate data* model или *enterprise data* model, *EDM*), которая создается в процессе проектирования OLTP-систем организации. При проектировании *корпоративной модели данных* обычно предпринимается попытка создать на основе бизнес-операций такую структуру данных, которая бы собрала и синтезировала в себе все информационные потребности организации.

Таким образом, *корпоративная модель данных* содержит в себе необходимую информацию для построения модели ХД. Поэтому на первом этапе, если такая модель есть в организации, *проектировщик ХД может начать проектирование ХД с решения задачи преобразования* *корпоративной модели данных* *в модель ХД*.

**Корпоративная модель данных**

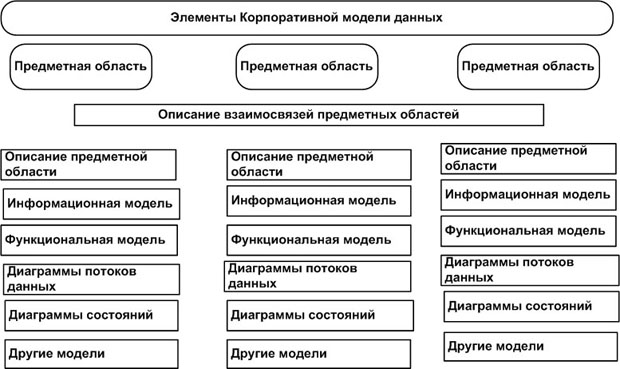
Как решить задачу преобразования *корпоративной модели данных* в модель ХД? Чтобы решить эту задачу, нужно иметь эту модель, т.е. *корпоративная модели данных* должна быть построена и *документирована*. И нужно понять, *что* из этой модели и *как* должно трансформироваться в модель ХД.

Уточним с позиций проектировщика ХД понятие *корпоративной модели данных*. **Под** ***корпоративной моделью данных*** **понимают многоуровневое, структурированное описание предметных областей организации, структур данных предметных областей, бизнес-процессов и бизнес-процедур, потоков данных, принятых в организации, диаграмм состояний, матриц "данные-процесс" и других модельных представлений, которые используются в деятельности организации**. Таким образом, в широком смысле слова, *корпоративная модель данных* представляет собой совокупность моделей различного уровня, которые характеризуют (моделируют на некотором абстрактном уровне) деятельность организации, т.е. содержание *корпоративной модели* напрямую зависит от того, какие модельные конструкции были включены в нее в данной организации.

Основными элементами *корпоративной модели данных* являются:

* описание предметных областей организации (определение сфер деятельности);
* взаимоотношения между определенными выше предметными областями;
* информационная модель данных (*ERD*-модель или модель "сущность-связь");
* для каждой предметной области описание:
  + ключей сущностей;
  + *атрибутов сущностей*;
  + подтипов и *супертипов*;
  + связей между сущностями;
  + группировки атрибутов;
  + взаимосвязей между предметными областями;
* функциональная модель или модель бизнес-процессов;
* диаграммы потоков данных;
* диаграммы состояний;
* другие модели.

Таким образом, *корпоративная модель данных* содержит сущности, атрибуты и отношения, которые представляют информационные потребности организации. На [рис. 16.1](https://intuit.ru/studies/courses/599/455/lecture/10176?page=1#image.16.1) изображены основные элементы *корпоративной модели данных*.

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_01.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_01.jpg)  
**Рис. 16.1.**Основные элементы корпоративной модели данных

**Уровни представления корпоративной модели данных**

*Корпоративная модель данных* подразделяется в соответствии с предметными областями, которые представляют группы сущностей, относящихся к поддержке конкретных нужд бизнеса. Некоторые предметные области могут покрывать такие специфические бизнес-функции, как управление контрактами, другие — объединять сущности, описывающие продукты или услуги.

Каждая логическая модель должна соответствовать существующей предметной области *корпоративной модели данных*. Если логическая модель не соответствует данному требованию, в нее должна быть добавлена модель, определяющая предметную область.

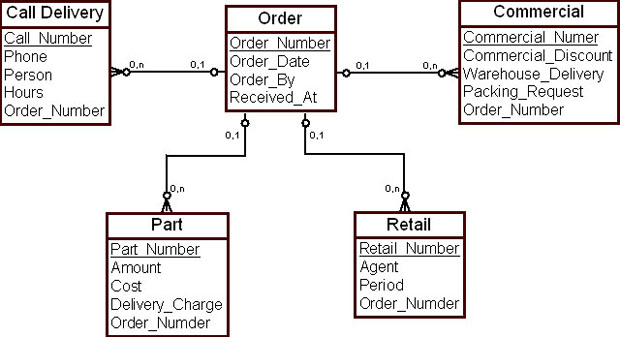
*Корпоративная модель данных* обычно имеет несколько уровней представления. На самом **высоком уровне** (high level) *корпоративной модели данных* располагается описание основных предметных областей организации и их взаимосвязей на уровне сущностей. На [рис. 16.2](https://intuit.ru/studies/courses/599/455/lecture/10176?page=1#image.16.2) приведен фрагмент *корпоративной модели данных* верхнего уровня.



**Рис. 16.2.**Фрагмент корпоративной модели данных верхнего уровня

На схеме, приведенной на рисунке, представлено четыре предметных области: "Покупатель" ( **Customer** ), "Счет" ( **account** ), "Заказ" ( **Order** ) и "Товар" ( **Product** ). Как правило, на верхнем уровне представления модели указываются только *прямые связи* между предметными областями, которые, например, фиксируют следующий факт: покупатель оплачивает счет на заказ товаров. Подробная информация и косвенные взаимосвязи на этом уровне *корпоративной модели* не приводятся.

На следующем, **среднем уровне** (mid level) *корпоративной модели данных* показывается подробная информация об объектах предметных областей, т. е. ключи и *атрибуты сущностей*, их взаимосвязи, подтипы и супертипы и т.д. Для каждой предметной области модели верхнего уровня существует одна модель среднего уровня. На [рис. 16.3](https://intuit.ru/studies/courses/599/455/lecture/10176?page=1#image.16.3) изображен средний уровень представления *корпоративной модели* для фрагмента предметной области "Заказ".

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_03.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_03.jpg)  
**Рис. 16.3.**Средний уровень представления корпоративной модели данных для фрагмента предметной области "Заказ"

Из [рис. 16.3](https://intuit.ru/studies/courses/599/455/lecture/10176?page=1#image.16.3) видно, что предметная область "Заказ" ( **Order** ) включает в себя несколько сущностей, определенных через их атрибуты, и взаимосвязей между ними. Представленная модель позволяет ответить на такие вопросы, как дата заказа, кто сделал заказ, кто отправил заказ, кто получает заказ и ряд других. Из приведенной схемы видно, что в данной организации выделяют два типа заказов – заказы по рекламной акции ( **Commersial** ) и заказы по розничной торговле ( **Retail** ).

Заметим, что *корпоративная модель данных* может представлять различные аспекты деятельности организации и с различной степенью детализации и завершенности. Если *корпоративная модель* представляет все аспекты деятельности организации, она еще называется **моделью данных организации** (*enterprise data* model).

С точки зрения проектирования ХД важным фактором в принятии решения создания модели ХД из *корпоративной модели данных* является состояние *завершенности* *корпоративной модели данных*.

*Корпоративная модель данных* организации обладает характеристикой эволюционности, т.е. она постоянно развивается и совершенствуется. Некоторые предметные области *корпоративной модели данных* могут быть хорошо проработаны, для некоторых работа еще может быть и не начата. Если фрагмент предметной области не проработан в *корпоративной модели данных*, то и нет возможности использовать эту модель как отправную точку проектирования ХД.

Степень завершенности *корпоративной модели* может быть нивелирована в проектировании ХД следующим образом. Поскольку процесс разработки ХД обычно разбивается во времени на последовательность этапов, процесс его проектирования можно синхронизировать с *процессом завершения* разработки отдельных фрагментов *корпоративной модели данных* организации.

На самом низком *уровне представления корпоративной модели данных* показывается информация о физических характеристиках объектов БД, соответствующих *логической модели данных* среднего *уровня представления корпоративной модели данных*.

**Алгоритм преобразования корпоративной модели данных в модель хранилища данных**

**Общее описание алгоритма**

Допустим, что *корпоративная модель данных* разработана в организации и документирована. Какие действия должен выполнить проектировщик ХД, чтобы преобразовать ее в модель ХД?

Как правило, на основе *корпоративной модели данных* разрабатываются ХД для информационной поддержки процесса принятия решений. Такие ХД составляют фундамент систем поддержки принятия решений (*СППР*, DSS). С этой точки зрения возникает основной вопрос, на который проектировщики ХД для таких систем должны дать ответ: какие данные (*атрибуты сущностей*) *корпоративной модели данных* следует сохранить в модели ХД, а какие можно не сохранять?

После отбора данных для модели ХД проектировщик должен рассмотреть вопрос о временных зависимостях в отобранных данных. Это связано с тем, что ХД для *СППР* обычно хранят временные ряды (исторические данные), отражающие изменение значений атрибутов модели во времени.

Поскольку объемы сохраняемых данных в ХД очень велики, проектировщик должен решить вопрос, хранить или нет в ХД производные атрибуты (или вычисляемые поля). Сохранение в ХД вычисленных значений позволит увеличить производительность обработки запросов.

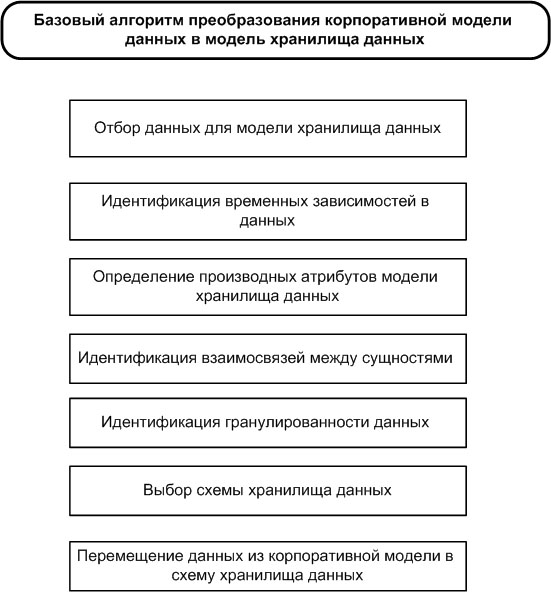
После того как проектировщик ХД рассмотрел сущности и их атрибуты, он должен обратить свое внимание на взаимосвязи между данными и решить вопрос о том, какие взаимосвязи между данными *корпоративной модели данных* следует перенести в модель ХД.

Следующий вопрос, который должен решить проектировщик ХД, не имеет прямого отношения к *корпоративной модели данных*, но должен быть решен до того, как проектировщик начнет моделировать данные по одной из типовых схем для ХД — "звезда" или "снежинка". Это вопрос об уровне структуризации (детализации) данных или *гранулированности данных* (Data *granularity*). Напомним, что **уровень структуризации данных — это степень детализации хранимых данных, оптимальная с точки зрения решения информационно-аналитических задач в рамках предметной области ХД**.

Далее проектировщик ХД может приступить к формированию схемы ХД типа "звезда" из таблиц *корпоративной модели данных*.

Таким образом, имеем базовый алгоритм преобразования *корпоративной модели данных* в модель ХД ([рис. 16.4](https://intuit.ru/studies/courses/599/455/lecture/10176?page=2#image.16.4)):

* выбрать данные данных *корпоративной модели*, которые следует хранить в ХД;
* исследовать временные зависимости данных и, если необходимо, добавить элемент времени в ключи сущностей ХД;
* добавить в модель производные элементы данных;
* преобразовать взаимосвязи между данными;
* определить уровень структуризации ( *гранулированности* ) данных в ХД;
* объединить данные из таблиц *корпоративной модели данных* в таблицах выбранной схемы ХД.



**Рис. 16.4.**Базовый алгоритм преобразования корпоративной модели данных в модель хранилища данных

В модификации этого алгоритма, согласно У. Инмону (W.H Inmon), проектировщику ХД предлагается решить еще два вопроса.

Первый вопрос — о выявлении периодических групп данных или массивов данных и представлении их в модели ХД (отношения "многие ко многим").

Второй вопрос — о разделении атрибутов согласно параметрам стабильности, известном в дисциплине обработки данных как вопрос о "трех массивах". Исходный массив данных с точки зрения алгоритмов обработки разбивается на три массива: очень медленно обновляемый, периодически обновляемый и очень быстро обновляемый.

Рассмотрим основные шаги алгоритма подробнее.

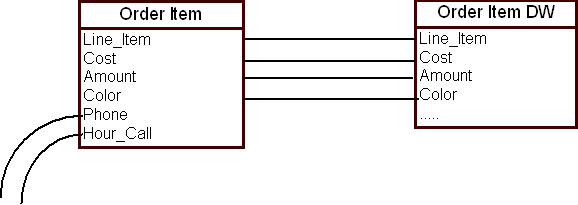
**Отбор данных из корпоративной модели**

Никаких унифицированных, формальных процедур отбора атрибутов *корпоративной модели данных* для сохранения их в модели ХД пока не предложено, хотя исследования в этом направлении проводятся. Основным критерием отбора является ответ на вопрос: какова вероятность того, что этот атрибут будет использоваться в процессе принятия решений?

Ответ на этот вопрос можно получить непосредственно, как результат анкетирования или опроса заинтересованных лиц, а именно лиц, принимающих решения. Если возможность всестороннего анкетирования отсутствует, то проектировщик ХД пытается исключить из рассмотрения следующие данные:

* данные, время жизни которых в *корпоративной модели данных* очень мало с точки зрения временных масштабов ХД;
* данные, которые не входят во временные зависимости, сохраняемые в ХД;
* данные, которые имеют значение или смысл только при оперативной обработке данных.

На [рис. 16.5](https://intuit.ru/studies/courses/599/455/lecture/10176?page=2#image.16.5) приведен пример отбора элементов данных сущности "Позиции заказа" ( **Order Item** ) *корпоративной модели данных* для модели ХД.



**Рис. 16.5.**Отбор данных корпоративной модели данных для модели ХД

Маловероятно, что атрибуты "Номер телефона" ( **Phone** ) и "Время вызова" ( **Hour\_call** ) будут иметь значение для принятия решений спустя несколько месяцев после прохождения заказа. Поэтому эти атрибуты можно не включать в модель ХД.

Атрибуты "Позиция заказа" ( **line\_item** ) — другими словами, закупаемый товар, "Стоимость" ( **cost** ) и "Количество" ( **amount** ) — практически всегда будут необходимы для анализа продаж. Эти атрибуты обязательно должны быть включены в модель ХД.

Вероятно, что атрибут "Цвет товара" ( **color** ) будет использоваться службой маркетинга при планировании маркетинговых и рекламных мероприятий. Для некоторых товаров цвет может и не иметь значения с точки зрения анализа. Предположим, что проектировщик ХД решил включить этот атрибут в модель ХД.

**Добавление временных элементов**

В *корпоративной модели данных* OLTP-системы сущности могут отражать текущее состояние моделируемого объекта и не отслеживать историю изменения этих состояний. С точки зрения принятия решений может быть важным поведение объекта предметной области (представленного в модели этой сущностью) во времени, и модель ХД должна отражать историю смены состояний такого объекта.

Тогда при импорте данных из OLTP-системы в ХД будет выполняться мгновенный снимок данных сущности, который будет фиксировать состояние объекта в момент импорта.

Проектировщик ХД должен ввести в модель ХД *временную метку*, соответствующую времени сохранения снимка в ХД. На [рис. 16.6](https://intuit.ru/studies/courses/599/455/lecture/10176?page=2#image.16.6) приведен пример добавления атрибута " *Временная метка* " ( **Time\_of\_snapshot** ) в модель ХД для сущности "Покупатель" ( **Customer** ) как часть ключа сущности.



**Рис. 16.6.**Добавление временной метки в модель ХД

Если в сущностях, переносимых из *корпоративной модели данных* в модель ХД, временные атрибуты входят в *составной ключ* сущности, то никаких действий проектировщика ХД по изменению ключа сущности не требуется.

**Производные элементы данных**

При проектировании БД OLTP-систем производные (выводимые) элементы данных, как правило, не включаются в структуру данных, и следовательно, *корпоративная модель данных* редко содержит такие данные. Производные данные не включаются в *корпоративную модель данных*, чтобы максимально исключить избыточность данных и сократить число элементов данных в модели.

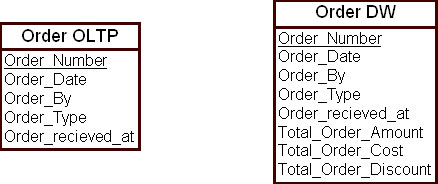
Вопрос, нужно ли включать производные данные в модель ХД, является важным с точки зрения объема вычислений при обработке данных в ХД. Если размер ХД очень большой или растет очень быстро, то целесообразно включить в модель данных такие элементы данных, которые, будучи один раз вычислены при загрузке ХД, с течением времени своих значений не изменяют.

Проектировщику следует иметь в виду, что включение новых элементов данных в модель ХД приводит к излишнему увеличению размера ХД. Поэтому в модель ХД следует включать такие производные элементы данных, которые предполагается использовать достаточно часто.

Таким образом, критерий отбора производных полей для включения в модель ХД состоит в выполнении следующих двух условий:

* частота использования производных полей приложениями ХД достаточно высока;
* вычисленные значения элементов данных после занесения их в ХД не изменяются со временем.

На [рис. 16.7](https://intuit.ru/studies/courses/599/455/lecture/10176?page=3#image.16.7) проиллюстрирована процедура добавления производных полей в модель ХД на примере сущности "Заказ" ( **Order** ).



**Рис. 16.7.**Добавление производных полей в модель ХД

Введенные в модель ХД производные поля сущности "Заказ" — это атрибуты "Суммарное количество заказа" ( **total\_order\_amount** ), "Итоговая стоимость заказа" ( **total\_order\_cost** ), "Итоговая скидка по заказу" ( **total\_order\_discount** ). После закрытия заказа эти атрибуты не будут изменять свои значения. С другой стороны, при анализе продаж организации в рамках ХД вероятность частого использования этих полей в запросах велика. Поэтому было целесообразно включить эти атрибуты в модель ХД.

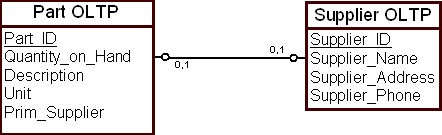
**Трансформация взаимосвязей между данными**

Одним из сложных вопросов проектирования ХД является вопрос о трансформации взаимосвязей *корпоративной модели данных* во взаимосвязи данных ХД. Проектировщик ХД должен ясно понимать суть этой проблемы, чтобы принять адекватные проектные решения, отражающие взаимосвязи в данных.

В *корпоративной модели данных*, как правило, предполагается, что каждая связь между сущностями фиксирует только одно бизнес-правило предметной области БД OLTP-системы. Эта взаимосвязь между сущностями корректна на момент доступа к данным, если данные адекватно отражают состояние сущностей в предметной области. Такой "моментный" характер интерпретации связи в *корпоративной модели данных* может потерять смысл при рассмотрении этой взаимосвязи между данными с точки зрения временной ретроспективы, которая характерна для модели ХД.

Рассмотрим вопрос, как взаимосвязь между сущностями *корпоративной модели* может быть представлена в ХД, когда данные из БД OLTP-системы загружены в ХД.

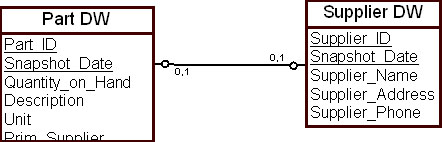
В качестве примера расследуем взаимосвязь между сущностями "Комплектующие" ( **Part OLTP** ) и "Поставщик" ( **Supplier OLTP** ). Эта взаимосвязь отражает бизнес-правило: каждое изделие поставляется от определенного поставщика. В частности, в организации может быть принято бизнес-правило: каждое изделие имеет основного поставщика. Тогда ограничение целостности состоит в том, что в БД не существует изделий, у которых нет основного поставщика. Это взаимосвязь действительна на момент доступа к данным ([рис. 16.8](https://intuit.ru/studies/courses/599/455/lecture/10176?page=3#image.16.8)).



**Рис. 16.8.**Взаимосвязь между изделием и поставщиком в оперативной БД

Допустим, что данные загружаются в ХД по принципу периодического моментального снимка. Периодический моментальный снимок таблицы с комплектующими деталями делается по концу отчетного периода — конец недели, конец месяца, конец квартала и т.д. Данные, описывающие изделие, захватываются в оперативной БД и помещаются в ХД в эти моменты времени. В том числе и основной поставщик изделия на момент взятия снимка, т.е. **связь представляется в ХД через значения своих данных на момент взятия снимка**.

Сущность "Комплектующие" меняет свое состояние достаточно быстро, поскольку изделия постоянно поставляются и отгружаются со склада. Сущность "Поставщик" меняет свое состояние гораздо медленнее, поэтому периодический моментальный снимок таблицы с поставщиками делается, например, раз в квартал. Таким образом, занесенный в ХД поставщик является по существу последним поставщиком данного изделия на момент взятия снимка ([рис. 16.9](https://intuit.ru/studies/courses/599/455/lecture/10176?page=3#image.16.9)).



**Рис. 16.9.**Трансформация взаимосвязи между изделием и поставщиком в ХД после периодического моментального снимка

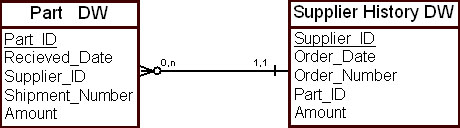
Главный недостаток таких периодических моментальных снимков состоит в том, что они могут неадекватно отражать состояние взаимосвязей между сущностями. В течение недели изделие может иметь пять основных поставщиков (т.е. менять их каждый рабочий день), но моментальный снимок не будет отражать таких изменений. Поставщик за квартал может отгрузить много различных изделий, но моментальный снимок покажет только последнюю сделанную отгрузку.

Из примера видно, что для того чтобы адекватно представить взаимосвязь между данными в ХД, требуется использовать историческую модель представления данных для рассмотренных сущностей.

В исторической модели данных сущность трактуется как совокупность своих последовательных состояний. Для исторической таблицы "Комплектующие" ( **Part DW** ) при получении изделия фиксируется вся подробная информация, в том числе и поставщик изделия. При таком подходе взаимосвязь между двумя сущностями, "Комплектующие" ( **Part DW** ) и "Поставщик" ( **Supplier DW** ), представлена в ХД полностью.

Если для поставщиков также поддерживается историческая таблица, то в нее заносятся данные о каждом заказе на получаемые изделия, сделанном поставщику. Таким образом, в ХД будут зафиксированы все отгрузки поставщика.

Пример дискретной исторической модели представления взаимосвязи приведен на [рис. 16.10](https://intuit.ru/studies/courses/599/455/lecture/10176?page=3#image.16.10).



**Рис. 16.10.**Представление взаимосвязи между изделием и поставщиком в ХД в исторической модели данных

Таким образом, при выборе формы представления взаимосвязи между данными в ХД проектировщик должен учитывать следующие правила:

1. историческая трактовка данных, относящихся к сущности;
2. изменение *качественного* содержания бизнес-правила *корпоративной модели данных* при представлении его в ХД.

**Определение уровня структуризации (гранулированности) данных**

Уровень структуризации или детализации данных ( *гранулированность* ), как уже указывалось в предыдущих лекциях, является важной характеристикой ХД. В процессе проектирования проектировщик ХД принимает решение об уровне структуризации данных. При использовании *корпоративной модели данных* проектировщик решает вопрос: достаточен ли уровень структурированности данных в этой модели для представления в модели ХД? И если нет, то устанавливает необходимый уровень структуризации данных.

Для определения уровня структуризации данных в модели ХД проектировщик пытается получить от аналитиков предметной области ответы на следующие типовые вопросы:

* Какие элементы данных будут суммироваться?
* За какой период времени данные будут суммироваться (день, месяц, год)?
* Соответствует ли выбранный уровень структуризации данных для ХД уровню структуризации в *корпоративной модели*?

При определении степени детализации фактов проектировщик должен обратиться, в первую очередь, к изучению бизнес-процессов организации, для которой разрабатывается модель ХД. Как правило, каждый бизнес-процесс требует отдельной *многомерной модели* с уникальным определением *гранулированности* фактов.

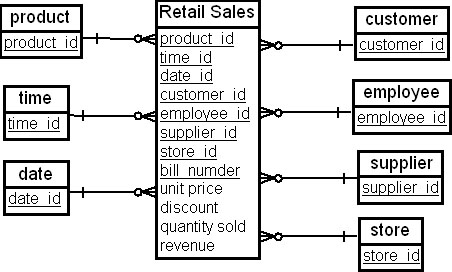
С точки зрения проектировщика ХД каждый бизнес-процесс состоит из нескольких фактов и измерений, которые отличаются от других бизнес-процессов. Давайте на примере рассмотрим применение этого простого принципа к определению *гранулированности* фактов в модели ХД.

Предположим, что нам нужно построить *многомерную модель* ХД для розничной торговли в сети магазинов, расположенных в различных регионах (бизнес-процесс). Первый вопрос, который интересует руководство организации, это анализ сбыта товаров во всех магазинах. Второй вопрос, который интересует руководство, — анализ причин возврата товаров по всем магазинам и анализ всех поставщиков с учетом процента возврата товаров по каждому из них.

Существуют два различных определения *гранулированности* фактов для определенного выше бизнес-процесса. Для анализа сбыта товаров *гранулированность* может быть определена следующим образом (Случай 1): на каждый товар, включенный в один счет покупки.

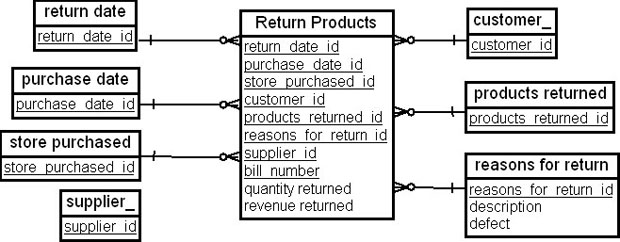
Для анализа причин возврата товаров *гранулированность* может быть определена следующим образом (Случай 2): конкретный товар, возвращенный покупателем в любой из магазинов.

В первом случае в качестве измерений будут выступать "Товар" ( **product** ), "Время" ( **time** ), "Покупатель" ( **customer** ), "Дата" ( **date** ), "Продавец" ( **employee** ), "Поставщик" ( **supplier** ) и "Магазин" ( **store** ). В качестве фактов могут быть выбраны следующие: "Цена товара" ( **unit price** ), "Скидка" ( **discount** ), "Количество проданных товаров" ( **quantity sold** ) и "Доход" ( **revenue** ). Схема "звезда" с таблицей фактов "Розничная продажа" ( **Retail Sales** ) для этого случая показана на [рис. 16.11](https://intuit.ru/studies/courses/599/455/lecture/10176?page=3#image.16.11).



**Рис. 16.11.**Схема "звезда" для модели ХД учета сбыта товаров по сети магазинов розничной торговли

Во втором случае в качестве измерений будут выступать "Дата возврата" ( **return date** ), "Дата покупки" ( **purchase date** ), "Покупатель" ( **customer** ), "Магазин, который продал бракованный товар" ( **store purchased** ). "Возвращенный товар" ( **products returned** ), "Причина возврата" ( **reasons for return** ), "Поставщик" ( **supplier** ). В качестве фактов могут быть выбраны "Потерянный доход от возврата" ( **revenue returned** ) и "Количество возвращенных товаров" ( **quantity returned** ). Схема "звезда" с таблицей фактов "Возвращенный товар" ( **Return Products** ) для этого случая показана на [рис. 16.12](https://intuit.ru/studies/courses/599/455/lecture/10176?page=3#image.16.12).

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_12.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_12.jpg)  
**Рис. 16.12.**Схема "звезда" для модели ХД учета возврата товаров по сети магазинов розничной торговли

Как видно из сказанного, одному бизнес-процессу может отвечать несколько многомерных моделей. Сколько создавать таблиц фактов для одного бизнес-процесса — определяет проектировщик ХД. Ясно, что для каждого бизнес-процесса должна быть одна модель (схема "звезда" или "снежинка"). Если *гранулированность* фактов различна, то их следует помещать в различные таблицы.

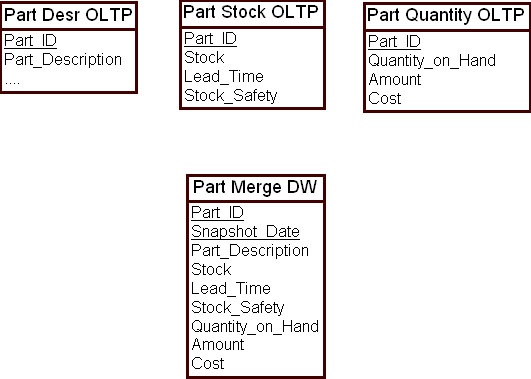
**Перенесение атрибутов из таблиц корпоративной модели данных в таблицы хранилища данных**

На следующем шаге преобразования *корпоративной модели данных* в модель ХД проектировщик объединяет набор таблиц *корпоративной модели данных* в таблицы ХД в рамках выбранной *многомерной модели*. Таблицы *корпоративной модели данных*, как правило, нормализованы в соответствии с требованиями реляционной модели. Объединение нескольких нормализованных таблиц в одну является *денормализацией* схемы БД. Для ХД большого размера *денормализация* выполняется исходя из соображений производительности. Объединение таблиц экономит дисковое пространство и увеличивает производительность обработки запросов.

Несмотря на соображения производительности, проектировщик ХД при объединении таблиц должен помнить о том, что таблицы имеет смысл объединять, если:

* таблицы имеют общий ключ;
* частота совместного использования данных из нескольких таблиц в запросах высока;
* данные добавляются в таблицы одинаковым образом.

Рассмотрим пример на [рис. 16.13](https://intuit.ru/studies/courses/599/455/lecture/10176?page=3#image.16.13). В *корпоративной модели данных* имеются три нормализованные таблицы, содержащие данные об изделиях. Все три таблицы БД OLTP-системы, показанные на рисунке сверху, имеют один и тот же первичный ключ "Номер изделия" ( **Part\_ID** ). Поэтому имеет смысл в модели ХД объединить эти таблицы в одну "Комплектующие объединенные" ( **Part Merge DW** ), как показано на рисунке внизу.



**Рис. 16.13.**Объединение таблиц корпоративной модели данных в модели ХД

**Массивы данных**

Нормализованные таблицы *корпоративной модели данных* не содержат периодических групп и массивов данных. Таковы требования реляционной модели. Однако для ХД использование таких элементов данных может быть целесообразным. По своему усмотрению проектировщик ХД может рассмотреть вопрос о создании массивов данных в реляционной модели. При принятии решения о создании массивов данных можно руководствоваться следующими соображениями — массив данных имеет смысл создавать, если:

* размерность массива постоянна или предсказуема;
* размер массива не очень велик;
* элементы массива используются совместно.

С точки зрения физического проектирования схемы базы данных это называется разбиением или *секционированием таблиц*. Разбиение и *секционирование* являются схожими методами, но различаются по механизму реализации. Разбиение — это логический прием, а *секционирование* — встроенный механизм диалекта SQL. Для ХД обычно используется *секционирование*.

*Секционирование* выполняется из соображений повышения производительности обработки запросов. Обоснование использования секционирования для схем ХД, кратко говоря, состоит в следующем. Таблица фактов, как правило, одна. Временной атрибут, как правило, входит как часть ключа. Время по своей природе имеет предсказуемую структуру измерения — день, неделя, месяц, квартал, год и т.д. Поэтому целесообразно рассмотреть возможность *секционирования таблицы* фактов по единицам измерения времени.

Например, телефонные компании часто выставляют счета своим абонентам за разговоры в течение месяца. Поэтому можно рассмотреть возможность представления таблицы оплаты счетов абонентами как массива из двенадцати элементов, каждый из которого отвечает месяцу года. Хотя на практике информационные службы телефонных компаний используют такой подход редко, с точки зрения анализа задолженности клиентов он оказывается весьма эффективным.

Примеры *секционирования таблиц* были рассмотрены в ["Знакомство с CASE инструментом"](https://intuit.ru/studies/courses/599/455/lecture/10166).

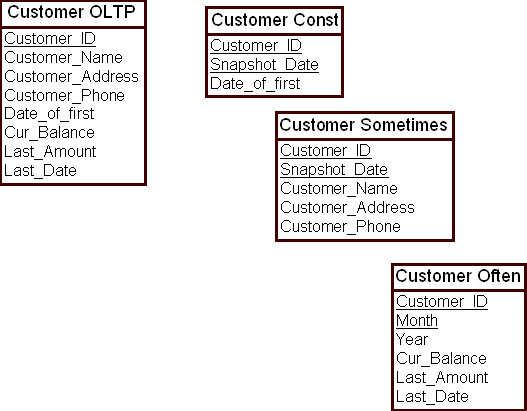
**Группировка данных в соответствии с частотой использования**

Реляционная модель (а, следовательно, и построенная на ее основе *корпоративная модель данных* — тоже) не различает скорость изменения данных внутри таблицы. ХД имеют одной из своих главных функций накопление данных, и скорость поступления данных для них имеет значение. Другими словами, ХД чувствительны к скорости добавления данных.

Чтобы нивелировать чувствительность к скорости добавления данных в ХД, таблицы следует организовывать следующим образом: медленно меняющиеся данные помещать в одну таблицу, а быстро меняющиеся — в другую. Т.е. скорость изменения количества данных в каждой из таблиц ХД должна быть приблизительно одинаковой.

Рассмотрим пример. В *корпоративной модели данных* имеются данные о покупателях, которые были объединены в одну таблицу – "Покупатели OLTP" ( **Customer OLTP** ). Среди атрибутов этой таблицы есть данные, которые никогда не меняются, например, значения атрибута "Дата первой покупки" ( **First\_Date** ). Есть и данные, которые иногда меняются, например, значение атрибута "Телефон покупателя" ( **Customer\_Phone** ). Также здесь существуют данные, которые меняются часто, например, значение атрибута "Дата последней покупки" ( **Last\_Date** ).

Такую таблицу целесообразно разбить на три таблицы. Первая таблица содержит атрибуты с постоянными данными; вторая содержит атрибуты, значения которых иногда меняются; третья — атрибуты, значения которых меняются часто. На [рис. 16.14](https://intuit.ru/studies/courses/599/455/lecture/10176?page=3#image.16.14) показана схема ХД, которая учитывает принцип изменчивости значений атрибутов.



**Рис. 16.14.**Принцип изменчивости данных в модели ХД

Таким образом, мы рассмотрели все восемь шагов базового алгоритма преобразования данных *корпоративной модели* в модель ХД.

Первый этап алгоритма перекидывает мостик от *корпоративной модели данных* к модели ХД. Этапы со второго по шестой по своей сути представляют процесс многомерного проектирования, который далее мы рассмотрим с позиции использования CASE-средств проектирования многомерных моделей. Седьмой и восьмой шаги непосредственно связаны с обеспечением производительности ХД на уровне проектных решений.

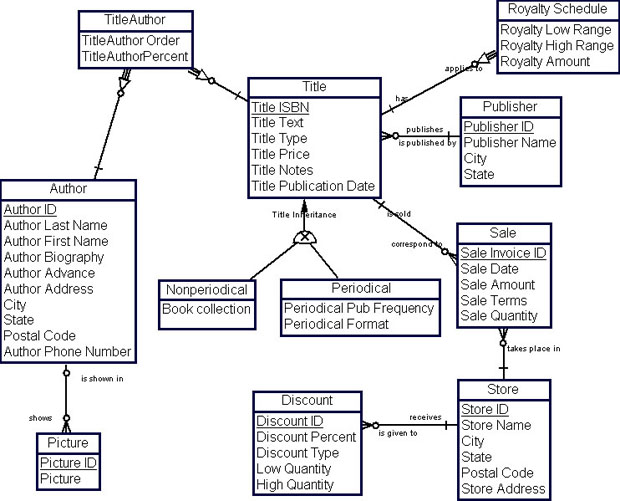
**Использование CASE-инструментов для создания многомерной модели на основе корпоративной модели данных**

**Корпоративная модель данных учебного примера**

Мы будем рассматривать применение CASE-средств на примере продукта PowerDesigner компании Sybase для моделирования ХД на основе СУБД MS SQL Server компании Microsoft.

Допустим, что руководство компании по розничной торговле издательской продукцией приняло решение о реализации ХД (а именно *киоска данных*) для анализа продаж изданий и анализа вклада авторов в общий доход компании. К моменту принятия решения о разработке *киоска данных* у ИТ-подразделения компании существовала *корпоративная модель данных* и операционная система обработки данных на СУБД MS SQL Server.

Логическая схема корпоративной модели издательской компании приведена на [рис. 16.15](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#image.16.15).

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_15.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_15.jpg)  
**Рис. 16.15.**Логическая схема корпоративной модели издательской компании

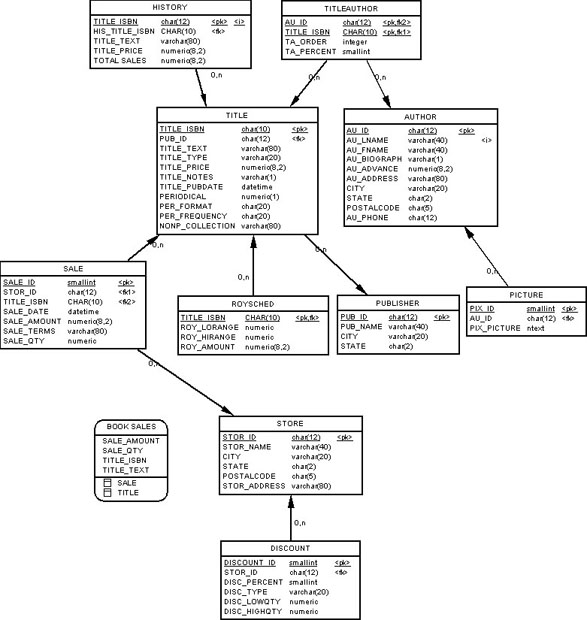
*Корпоративная модель данных* компании документирована. Все сущности, *атрибуты сущностей* и домены описаны. Пример описания сущности "Автор" ( **Autor** ) приведен в [табл. 16.1](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#table.16.1).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица 16.1. Сущность "Автор" (Autor) | | | | | |
|  | **Имя атрибута** | **Код атрибута** | **Тип данных** | **Домен** | **Ключ** |
| 1 | Author ID (Идентификатор автора) | AU\_ID | A12 | Алфавитно-цифровой | Да, NOT NULL |
| 2 | Author Last Name (Фамилия) | AU\_LNAME | VA40 | Алфавитный | NOT NULL |
| 3 | Author First Name (Имя) | AU\_FNAME | VA40 | Алфавитный | NOT NULL |
| 4 | Author Biography (Биография автора) | AUTHOR BIOGRAPHY | TXT | Текст |  |
| 5 | Author Advance (Доплаты авторам) | AU\_ADVANCE | MN8,2 | Десятичное число |  |
| 6 | Author Address (Адрес автора) | AU\_ADDRESS | VA80 | Текст |  |
| 7 | City (Населенный пункт) | CITY | VA20 | Алфавитный |  |
| 8 | State (Область) | STATE | VA80 | Алфавитный |  |
| 9 | *Postal Code* (Почтовый индекс) | POSTALCODE | N | Целочисленный |  |
| 10 | Author Phone Number (Телефон автора) | AU\_PHONE | A12 | Алфавитно-цифровой | NOT NULL |

Идентифицируем бизнес-процессы предметной области *киоска данных* по представленной на [рис. 16.15](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#image.16.15) модели "сущность-связь" и применим алгоритм преобразования *корпоративной модели*, описанный в предыдущих разделах настоящей лекции.

Для проектирования *многомерной модели* нам потребуется еще одни документ – физическая модель операционной системы обработки данных компании, которая приведена на [рис. 16.16](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#image.16.16).

Она нам нужна для правильного выбора наименований полей и их типов при перенесении в *многомерную модель* данных.

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_16.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_16.jpg)  
**Рис. 16.16.**Физическая схема корпоративной модели

**Идентификация бизнес-процессов компании и отбор данных**

Из задач, поставленных руководством компании по розничной торговле издательской продукцией, следует, что мы имеем два бизнес-процесса для модели ХД: продажи изданий через магазины и продажи изданий конкретных авторов.

Для бизнес-процесса продажи изданий через магазины (бизнес-процесс 1) мы имеем следующие сущности *корпоративной модели данных*:

* сущность-*супертип* "Наименование изданий" ( **Тitle** ) с подчиненными ей подтипами "Периодические издания" ( **Periodical** ) и "Непериодические издания" ( **Nonperiodical** );
* сущность "Продажи" ( **Sale** );
* сущность "Магазин" ( **Store** ).

Для бизнес-процесса продажи изданий конкретного автора (бизнес-процесс 2) мы имеем следующие сущности *корпоративной модели данных*:

* сущность "Автор" ( **Author** );
* сущность "Продажи" ( **Sale** );
* сущность "Издание Автор" ( **TitleAuthor** ).

Применение алгоритма преобразования *корпоративной модели* в *многомерную модель* данных разберем на примере бизнес-процесса 1. Преобразование для бизнес-процесса 2 выполняется аналогично.

Заметим, что факты в *многомерной модели* отражают связи "многие ко многим" между измерениями. Это означает, что внешние ключи в таблицах фактов следует искать в отношениях, разрешающих такие связи.

Хорошим правилом будет разбить выделенные сущности на два класса: транзакционные сущности, которые фиксируют бизнес-операции, и нетранзакционные сущности, которые хранят описания бизнес-операций и бизнес-процессов. Первые претендуют быть прототипами таблиц фактов, а вторые – таблиц измерений *многомерной модели*.

Для бизнес-процесса 1 такое разбиение представлено в [табл. 11.2](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#table.16.2).

|  |  |
| --- | --- |
| Таблица 16.2. Разбиение сущностей на типы | |
| **Транзакционные сущности** | **Нетранзакционные сущности** |
| "Продажи" (Sale) - описывает бизнес-операцию продажи издания | "Магазин" (Store) описывает магазин как объект бизнес-процесса, через который осуществляется продажа |
|  | "Наименование изданий" (Тitle) с подчиненными ей "Периодические издания" (Periodical) и "Непериодические издания" (*Nonperiodical*) описывает издания, которые продаются в магазинах |

Таким образом, для бизнес-процесса 1 имеем таблицу фактов "Продажи" (Sale) и две таблицы измерений "Магазин" (Store) и "Наименование изданий" (Тitle), при этом обратим внимание на то, что в физической модели была выполнена *денормализация* сущности "Наименование изданий" включением в нее атрибутов подчиненных сущностей "Периодические издания" ( **Periodical** ) и "Непериодические издания" ( **Nonperiodical** ).

Следуя алгоритму преобразования *корпоративной модели*, отберем *атрибуты сущностей*, которые целесообразно представить в *многомерной модели* *киоска данных*. Результаты работы целесообразно свести в таблицу, как показано в [табл. 11.3](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#table.16.3).

|  |  |  |
| --- | --- | --- |
| Таблица 16.3. Отобранные атрибуты | | |
| **Таблица модели ХД** | **Таблица внешней БД** | **Отобранные атрибуты** |
| Sale | Sale | SALE\_ID (Идентификатор продажи) |
| STOR\_ID (Идентификатор магазина) |
| TITLE\_ISBN (ISBN издания) |
| SALE\_DATE (Дата продажи) |
| SALE\_AMOUNT (Сумма) |
| SALE\_TERMS (Условия продажи) |
| SALE\_QTY (Количество) |
| Store | Store | STOR\_ID (Идентификатор магазина) |
| STOR\_NAME (Название магазина) |
| CITY (Населенный пункт) |
| STATE (Область) |
| POSTALCODE (Почтовый индекс) |
| STOR\_ADDRESS (Адрес магазина) |
| Тitle | Тitle | TITLE\_ISBN (ISBN издания) |
| PUB\_ID (Идентификатор издателя) |
| TITLE\_TEXT (Название издания) |
| TITLE\_TYPE (Тип издания) |
| TITLE\_PRICE (Цена издания) |
| TITLE\_NOTES (Признак примечаний) |
| TITLE\_PUBDATE (Дата выпуска издания) |
| PERIODICAL (Признак периодичности издания) |
| PER\_FREQUENCY (Частота выпуска издания) |
| NONP\_COLLECTION (Непериодическое издание) |

Предположим, что мы будем создавать транзакционную таблицу фактов "Продажи" (Sale) для нашего *киоска данных*. Тогда временная зависимость данных будет определяться датой продажи, т. е. значением атрибута "Дата продажи" ( **SALE\_DATE** ) (второй этап алгоритма преобразования *корпоративной модели данных* ). Поэтому никаких дополнительных *временных меток* в нашу *многомерную модель* мы вводить не будем, так же, как и не будем вводить в модель производные поля (агрегаты).

Рассмотрим теперь взаимосвязи между данными (четвертый этап преобразования *корпоративной модели данных* ). В нашем случае *транзакционной таблицы фактов* преобразование связей не требуется. Таблица фактов фиксирует историю ежедневных продаж изданий в магазинах.

Определим теперь уровень *гранулированности* фактов (пятый этап преобразования *корпоративной модели данных* ). Конкретное издание продается в конкретном магазине в определенный день. Это бизнес-правило определяет уровень *гранулированности* фактов: в таблице фактов для каждой продажи будет существовать одна строка, фиксирующая продажу конкретного издания в конкретном магазине в конкретный день.

В нашем случае отсутствуют массивы данных, поэтому шестой этап алгоритма преобразования мы выполнять не будем.

Теперь мы обладаем необходимыми данными, чтобы построить *многомерную модель* данных. Выберем для нашей *многомерной модели* схему "звезда".

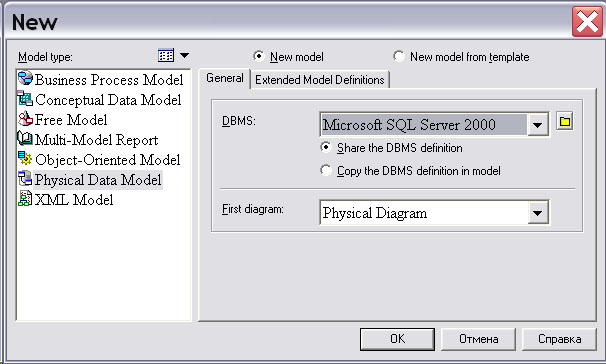
*Секционирование таблиц* фактов будет рассмотрено далее, после построения *многомерной модели*.

Рассмотрим использование алгоритма преобразования *корпоративной модели данных* компании при помощи CASE-средства PowerDesigner компании Sybase.

**Создание многомерной модели данных в PowerDesigner**

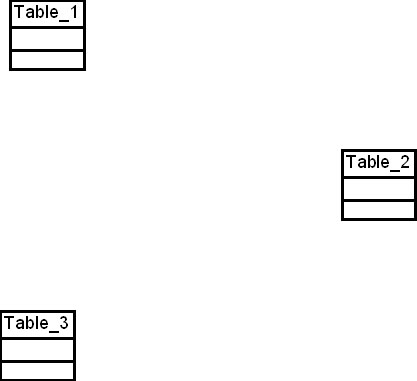
Сначала мы создадим *многомерную модель* *киоска данных* для анализа продаж изданий.

Для этого запустим CASE-инструментарий PowerDesigner, выберем функцию " **Создать новую физическую модель данных** " (Пункты меню **File \to New** ). В появившемся на экране диалоговом окне " **New** " выберем в списке "Тип модели" ( **Model type** ) элемент "*Физическая модель данных*" ( **Physical Data Model** ), из выпадающего списка СУБД ( **DBMS** ) выберем **MS SQL Server**, а из списка "Первая диаграмма" ( **First Diagram** ) выберем "Диаграмма физической модели" ( **Physical Diagram** ) и нажмем на кнопку " **OK** " ([рис. 16.17](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#image.16.17)).



**Рис. 16.17.**Создание многомерной модели: выбор СУБД

Далее создадим таблицы, которые будут таблицей фактов и двумя таблицами измерений, используя палитру инструментов ( **Palette** ). Чтобы создать таблицу, необходимо навести курсор мыши на пиктограмму таблицы на палитре инструментов, щелкнуть по пиктограмме левой кнопкой мыши, перенести курсор на рабочее пространство, еще раз щелкнуть левой кнопкой мыши ([рис. 16.18](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#image.16.18)) и выбрать на палитре инструментов пиктограмму указателя, чтобы перейти к выполнению других функций.

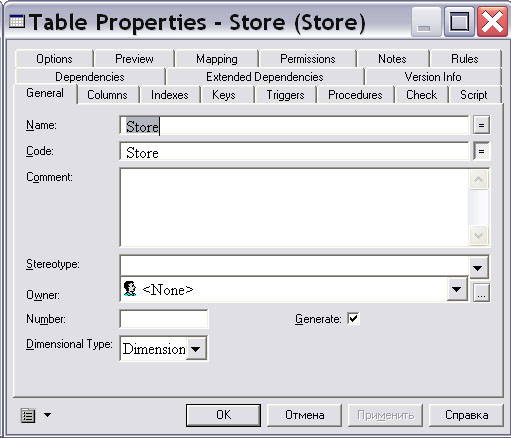


**Рис. 16.18.**Создание многомерной модели: создание таблиц модели

Теперь перенесем данные из таблиц *корпоративной модели данных* в объекты *многомерной модели* данных (т.е. выполним шестой этап алгоритма преобразования *корпоративной модели данных* ).

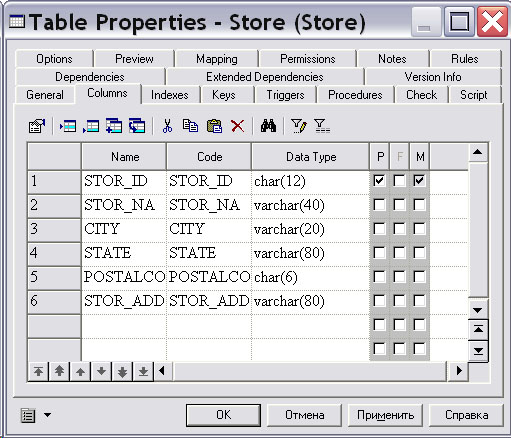
**Определение атрибутов многомерной модели данных в PowerDesigner**

Начнем с определения элементов таблиц измерений. Для примера рассмотрим измерение "Магазин" ( **Store** ). Двойным щелчком мыши на диаграмме таблицы раскроем диалоговое окно "Свойства таблицы" ( **Table Properties …**), в поле "Имя" ( **Name** ) вкладки "Общие" ( **General** ) наберем на клавиатуре " **Store** ". Далее в списке "Тип таблицы" ( **Dimensional Type** ) выберем тип таблицы "Измерение" ( **Dimension** ) и нажмем на кнопку " **Применить** " ([рис. 16.19](https://intuit.ru/studies/courses/599/455/lecture/10176?page=5#image.16.19)).



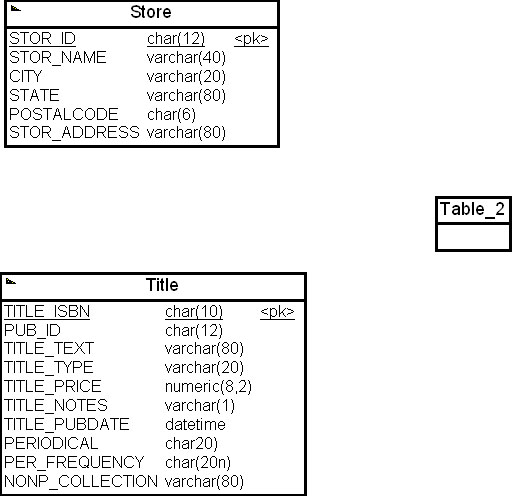
**Рис. 16.19.**Создание многомерной модели: определение наименования таблицы измерений

Теперь перейдем на вкладку "Колонки" ( **Columns** ). В открывшейся таблице наберем на клавиатуре имена атрибутов измерения (см. [табл. 11.3](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#table.16.3) и [рис. 16.20](https://intuit.ru/studies/courses/599/455/lecture/10176?page=5#image.16.20)), установим первичным ключом измерения колонку **STOR\_ID** и нажмем на кнопку " **OK** ".



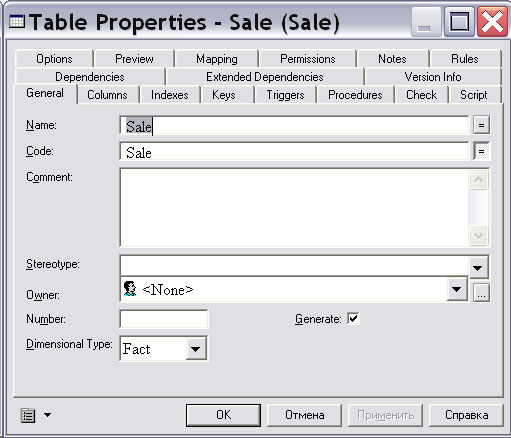
**Рис. 16.20.**Создание многомерной модели: определение атрибутов таблицы измерений

Выполнив аналогичные операции для измерения "Наименование изданий" ( **Title** ) с первичным ключом **TITLE\_ISBN**, получим фрагмент схемы, показанный на [рис. 16.21](https://intuit.ru/studies/courses/599/455/lecture/10176?page=5#image.16.21).



**Рис. 16.21.**Создание многомерной модели: определение таблиц измерений

Теперь перейдем к определению полей таблицы фактов. Двойным щелчком мыши на диаграмме таблицы раскроем диалоговое окно "Свойства таблицы" ( **Table Properties …** ), в поле "Имя" ( **Name** ) вкладки "Общие" ( **General** ) наберем на клавиатуре " **Sale** ". Далее в списке "Тип таблицы" ( **Dimensional Type** ) выберем тип таблицы "Факт" ( **Fact** ) и нажмем на кнопку " **Применить** " ([рис. 16.22](https://intuit.ru/studies/courses/599/455/lecture/10176?page=5#image.16.22)).



**Рис. 16.22.**Создание многомерной модели: определение таблицы фактов

Далее перейдем на вкладку "Колонки" ( **Columns** ). В открывшейся таблице наберем на клавиатуре имена атрибутов и метрик факта (см. [табл. 11.3](https://intuit.ru/studies/courses/599/455/lecture/10176?page=4#table.16.3) и [рис. 16.20](https://intuit.ru/studies/courses/599/455/lecture/10176?page=5#image.16.20)) и нажмем на кнопку " **ОК** " ([рис. 16.23](https://intuit.ru/studies/courses/599/455/lecture/10176?page=5#image.16.23)).

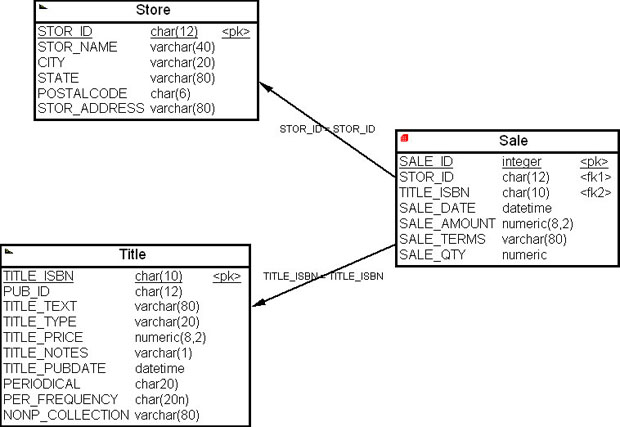
[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_23.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_23.jpg)  
**Рис. 16.23.**Создание многомерной модели: определение полей таблицы фактов

Первичным ключом таблицы фактом выбрано поле **SALE\_ID**. Теперь перейдем к установлению связей между таблицей фактов и таблицами измерений, т. е. отразим в *многомерной модели* результаты четвертого и пятого этапов алгоритма преобразования *корпоративной модели данных*.

**Определение связей многомерной модели данных в PowerDesigner**

Для того чтобы установить связь между таблицами, необходимо навести курсор мыши на пиктограмму связи палитры инструментов, щелкнуть по ней левой кнопкой мыши, установить курсор мыши на таблицу (в нашем случае — на таблицу фактов), не отпуская левой кнопки мыши, перенести курсор на другую таблицу (в нашем случае таблицу измерений) и отпустить кнопку мыши ([рис. 16.24](https://intuit.ru/studies/courses/599/455/lecture/10176?page=5#image.16.24)).

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_24.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_24.jpg)  
**Рис. 16.24.**Создание многомерной модели: определение связей между таблицами измерений и таблицей фактов

Связь между таблицами показывается на диаграмме стрелкой, которая в данном случае подписана условием соединения таблиц. В таблице фактов появятся первичные ключи измерений как внешние ключи таблицы фактов.

**Обсуждение примера**

Таким образом, мы выполнили построение *многомерной модели* *киоска данных* на основе *корпоративной модели данных*.

Скрипт для создания *киоска данных* в СУБД MS SQL Server приведен ниже.

/\* Table: Sale \*/

create table Sale (

SALE\_ID integer not null,

STOR\_ID char(12) null,

TITLE\_ISBN char(10) null,

SALE\_DATE datetime null,

SALE\_AMOUNT numeric(8,2) null,

SALE\_TERMS varchar(80) null,

SALE\_QTY numeric null,

constraint PK\_SALE primary key (SALE\_ID)

)

go

/\* Table: Store \*/

create table Store (

STOR\_ID char(12) not null,

STOR\_NAME varchar(40) null,

CITY varchar(20) null,

STATE varchar(80) null,

POSTALCODE char(6) null,

STOR\_ADDRESS varchar(80) null,

constraint PK\_STORE primary key (STOR\_ID)

)

go

/\* Table: Title \*/

create table Title (

TITLE\_ISBN char(10) not null,

PUB\_ID char(12) null,

TITLE\_TEXT varchar(80) null,

TITLE\_TYPE varchar(20) null,

TITLE\_PRICE numeric(8,2) null,

TITLE\_NOTES varchar(1) null,

TITLE\_PUBDATE datetime null,

PERIODICAL char20) null,

PER\_FREQUENCY char(20n) null,

NONP\_COLLECTION varchar(80) null,

constraint PK\_TITLE primary key (TITLE\_ISBN)

)

go

alter table Sale

add constraint FK\_SALE\_REFERENCE\_STORE foreign key (STOR\_ID)

references Store (STOR\_ID)

go

alter table Sale

add constraint FK\_SALE\_REFERENCE\_TITLE foreign key (TITLE\_ISBN)

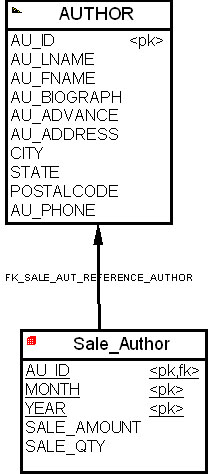
references Title (TITLE\_ISBN)

go

Листинг .

Построенный кисок данных предназначен для анализа продаж компании розничной торговли издательской продукцией через сеть своих магазинов (бизнес-процесс 1).

Проводя аналогичные рассуждения для бизнес-процесса 2, мы получим схему *многомерной модели*, состоящую из таблицы фактов "Продажи по автору" ( **Sale\_Author** ) и измерения "Авторы" ( **Author** ), как показано на [рис. 16.25](https://intuit.ru/studies/courses/599/455/lecture/10176?page=6#image.16.25).



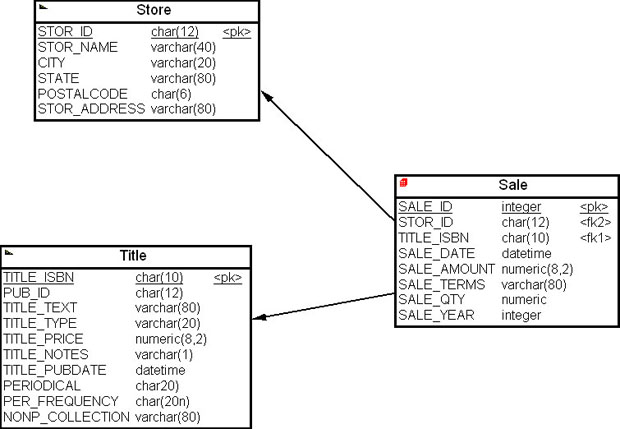
**Рис. 16.25.**Многомерная модель киоска данных для анализа продаж по автору

Заметим, что в этой *многомерной модели* *киоска данных* *гранулированность* определена как продажи автора за месяц, а поля таблицы фактов "Сумма продаж" ( **SALE\_AMOUNT** ) и "Количество продаж" ( **SALE\_QTY** ) являются производными полями, вычисляемыми как соответствующие суммы за месяц.

**Секционирование таблиц многомерной модели данных в PowerDesigner**

Рассмотрим, как *CASE*-средства помогают проектировщику данных выполнять *секционирование таблиц* *многомерной модели* данных. Обратимся к примеру из предыдущего раздела настоящей лекции и рассмотрим *многомерную модель* *киоска данных* на [рис. 16.24](https://intuit.ru/studies/courses/599/455/lecture/10176?page=5#image.16.24).

Предположим, что маркетинговая служба анализирует постоянно продажи изданий текущего года, а по годам *анализ* продаж выполняется один раз в год. Тогда целесообразно с целью увеличения производительности запросов выполнить горизонтальное *секционирование таблицы* фактов "Продажи" ( **Sale** ) по годам. Для этого введем в таблицу фактов *атрибут* "Год продажи" ( **SALE\_YEAR** ) ([рис. 16.26](https://intuit.ru/studies/courses/599/455/lecture/10176?page=6#image.16.26)). Предположим также, что киоск данных будет наполняться данными, начиная с 2007 года.

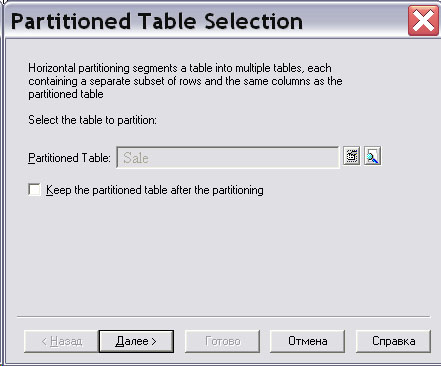
[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_26.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_26.jpg)  
**Рис. 16.26.**Добавление ключа секционирования в таблицу фактов "Продажи" (Sale)

Заметим, что вместо этого действия мы могли ввести еще одну таблицу измерения "Время" с атрибутом "Год" и установить *связь* этого измерения с таблицей фактов модели.

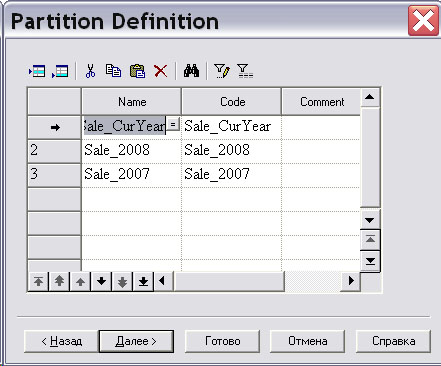
Напомним, что горизонтальное *секционирование таблицы* состоит в разбиении исходной таблицы на несколько таблиц, каждая из которых содержит *подмножество* строк исходной таблицы.

Для того чтобы выполнить горизонтальное *секционирование таблицы* фактов "Продажи" ( **Sale** ) при помощи *CASE* PowerDesigner, можно щелкнуть правой кнопкой мыши на таблице фактов и из всплывающего *меню* выбрать *пункт* "Горизонтальное *секционирование*" ( **Horizontal Partitioning** ). На экране появится *диалоговое окно* "Выбор таблицы для секционирования" ( **Partitioned Table Selection** ). Выбрав таблицу "Продажи" ( **Sale** ), нажмите на кнопку " **Далее** " ([рис. 16.27](https://intuit.ru/studies/courses/599/455/lecture/10176?page=6#image.16.27)).



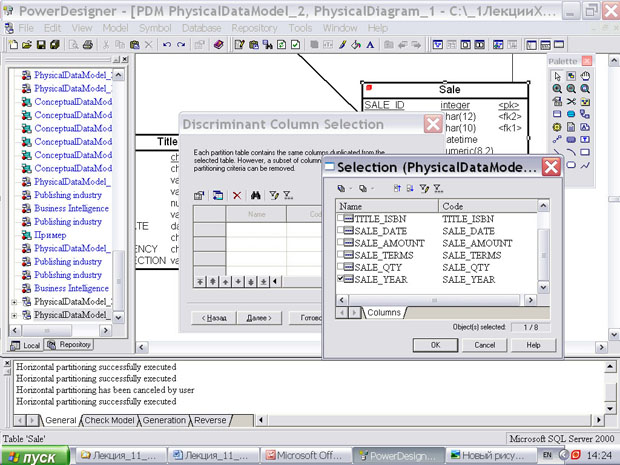
**Рис. 16.27.**Выбор таблицы для секционирования

На экране появится *диалоговое окно* "*Определение* секций" ( **Partition Definitions** ), в котором необходимо определить имена секций (пусть это будут имена **Sale\_CurYear**, **Sale\_2008** и **Sale\_2007** ) и нажать на кнопку " **Далее** " ([рис. 16.28](https://intuit.ru/studies/courses/599/455/lecture/10176?page=6#image.16.28)).



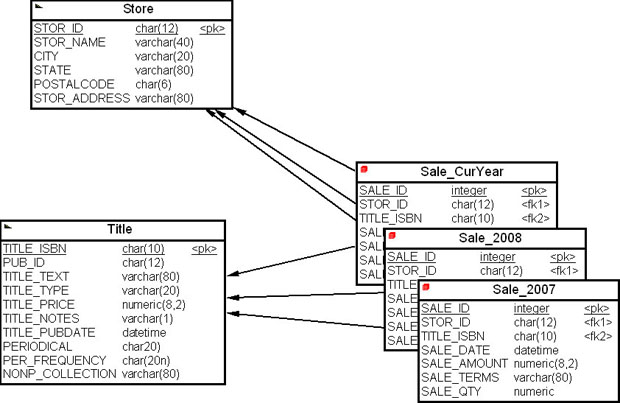
**Рис. 16.28.**Определение имен секций

На экране появится *диалоговое окно* "Выбор дискриминантной колонки" ( **Discriminant Column Selection** ). Дискриминантная колонка – это колонка таблицы, которая входит в критерий секционирования, поэтому ее не обязательно хранить в секциях. Щелкнув левой кнопкой мыши на пиктограмме " **Добавить строку** ", вы активизируете *диалоговое окно* "Выбор" ( **Selection** ). Выберем в списке колонок колонку "Год продажи" ( **SALE\_YEAR** ), по которой будет выполняться *секционирование*, и нажмем кнопки " **ОК** " и " **Далее** " ([рис. 16.29](https://intuit.ru/studies/courses/599/455/lecture/10176?page=6#image.16.29)).

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_29.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_29.jpg)  
**Рис. 16.29.**Выбор дискриминантной колонки

На экране появится *диалоговое окно* "*Информация* о секционировании" ( **Partitioning Information** ). Далее нажмем кнопку " **Готово** ". После расстановки на рабочем пространстве секций получим *многомерную модель* данных нашего *киоска данных* с секционированной таблицей "Продажи" ( **Sale** ), как на [рис. 16.30](https://intuit.ru/studies/courses/599/455/lecture/10176?page=6#image.16.30).

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_30.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/16/files/11_30.jpg)  
**Рис. 16.30.**Многомерная модель киоска данных учебного примера с секционированной таблицей "Продажи" (Sale)

Таким образом, мы на примере рассмотрели, как можно использовать *CASE*-средство для преобразования *корпоративной модели данных* в *многомерную модель* ХД.

**Резюме**

При создании ХД в масштабе организации *корпоративная модель данных*, как правило, выступает отправной точкой проектирования модели ХД. В этом случае процедуру проектирования модели ХД можно разбить на следующие шаги.

1. Выбрать данные данных *корпоративной модели*, которые будут сохраняться в ХД.
2. Исследовать временные зависимости данных и, если необходимо, добавить элемент времени в ключи сущностей ХД.
3. Добавить в модель производные элементы данных.
4. Преобразовать взаимосвязи между данными.
5. Определить уровень структуризации ( *гранулированности* ) данных в ХД.
6. Перенести данные из таблиц *корпоративной модели данных* в таблицы выбранной схемы ХД.
7. Выявить периодические группы данных или массивы данных.
8. Разделить атрибуты согласно параметрам стабильности.

С точки зрения конкретного диалекта *SQL*, в рассмотренный *алгоритм* можно добавить еще несколько шагов, также связанных с обеспечением производительности ХД на проектном уровне — в частности, *иерархии измерений* и материализуемые представления.

Выполнять проектирование модели ХД на основе *корпоративной модели данных* организации целесообразно с помощью *CASE*-инструментария. Использование *CASE*-инструментария увеличивает *производительность* труда проектировщика ХД, особенно в случае крупных и средних проектов.