**Лекция 11:**

**Проектирование и разработка процесса ETL**

**Цель лекции**

Изучив материал настоящей лекции, вы будете знать:

* что такое процесс *ETL*;
* место процесса *ETL* в архитектуре системы бизнес аналитики на основе хранилищ данных;
* что такое реализация *ETL*-процесса с использованием промежуточной области;
* что такое реализация *ETL*-процесса без использования промежуточной области;
* основные *элементы ETL-процесса* ;

и научитесь:

* строить *диаграммы движения данных*, *диаграммы преобразования данных*, *диаграммы управления потоком преобразования данных* ;
* выполнять общее планирование реализации *ETL*-процесса;
* проектировать *ETL*-процессы.

**Введение**

Для того чтобы заставить ХД заработать, необходимо не просто обеспечить взаимодействие многих источников данных – важно тщательно спланировать это взаимодействие. Поэтому процессы *извлечения*, *преобразования* и *загрузки данных* играют важную роль в создании и эксплуатации ХД.

Чтобы процесс *преобразования данных* протекал без сбоев, необходимо обеспечить наличие необходимой документации и метаданных. Процесс *извлечения данных* влияет на *производительность* других систем, поэтому его следует рассматривать в аспекте управления изменениями и конфигурацией систем – источников данных.

**Под аббревиатурой ETL** (*extraction*, *transformation*, loading — *извлечение*, *преобразование* и *загрузка данных* ) **понимается составной процесс переноса данных одного приложения или автоматизированной информационной системы в другие**.

Процесс *ETL* реализуется путем либо разработки приложения *ETL*, либо создания комплекса встроенных программных процедур, либо использования *ETL*-инструментария. Приложения *ETL* извлекают информацию из исходных *БД* источников, преобразуют ее в формат, поддерживаемый *БД* назначения, а затем загружают в эту *БД* преобразованные данные.

Цель любого *ETL*-приложения состоит в том, чтобы своевременно доставить данные из внешних систем в систему, с которой работают пользователи. Как правило, *ETL*-приложения используются при переносе данных внешних источников в ХД систем бизнес-аналитики. Поэтому организация процесса *ETL* является составной частью проекта разработки практически любого ХД.

Проектирование и разработка *ETL*-процесса является одной из самых важных задач проектировщика ХД. Для ХД процесс *ETL* имеет следующие свойства. Во-первых, объем данных, который выбирается из систем источников данных и помещается в ХД, как правило, бывает достаточно большим, до десятков Гб. Во-вторых, процесс *ETL* является необходимой составной частью эксплуатации ХД. *Периодичность* процесса *ETL* определяется не только потребностью пользователя в своевременных данных, но и размером загружаемой порции данных. *По* оценкам специалистов, *ETL*-процесс может занимать до 80% времени. В-третьих, на разных стадиях процесса *ETL* формируются *метаданные* ХД и обеспечивается *качество данных*. В-четвертых, во время процесса *ETL* может произойти потеря данных, поэтому необходимо обеспечивать *контроль* за поступлением данных в ХД. В-пятых, процесс *ETL* обладает свойством восстанавливаемости после сбоев без потери данных.

На [рис. 15.1](https://intuit.ru/studies/courses/599/455/lecture/10175?page=1#image.15.1) показано *место* процесса *ETL* в архитектуре системы бизнес-аналитики на основе ХД.

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_01.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_01.jpg)  
**Рис. 15.1.**Процесс ETL в архитектуре хранилища данных

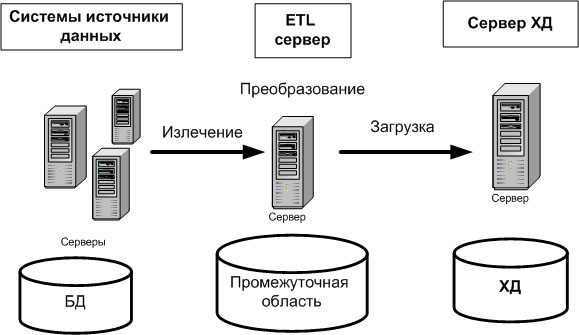
Как видно из рисунка, на процесс *ETL* возложена вся работа *по* подготовке данных для доставки их в ХД, формирование и обновление метаданных ХД, а также *управление данными*, извлеченными в результате *Data* mining.

Таким образом, процесс *ETL* состоит из трех основных стадий.

* ***Извлечение данных*** На этой стадии отбираются и описываются данные внешних источников (начинают формироваться метаданные ХД), которые должны храниться в ХД (релевантные данные).
* ***Преобразование данных*** На этой стадии релевантные данные преобразуются в формат представления данных в ХД, правила *преобразования* сохраняются в метаданных ХД, формируются ключевые поля таблиц физической структуры ХД, выполняется *очистка данных*.
* ***Загрузка данных*** На этой стадии данные загружаются в ХД, выполняется построение агрегатов.

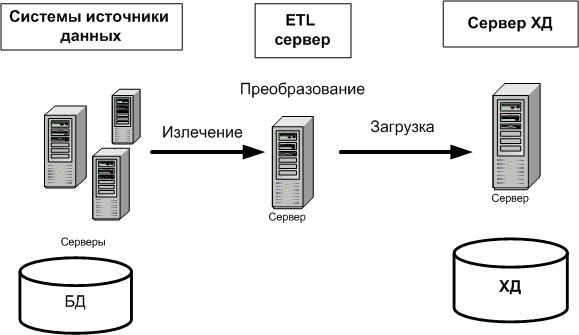
**Подходы к реализации ETL-процесса**

Существует несколько подходов к реализации процесса *ETL*. Общепринятый подход состоит в *извлечении данных* из систем источников, размещении их в промежуточной области дисковой памяти (*Data* Staging *Area*), выполнении в этой промежуточной области процедур *преобразования* и *очистки данных*, а затем *загрузки данных* в ХД, как показано на [рис. 15.2](https://intuit.ru/studies/courses/599/455/lecture/10175?page=1#image.15.2).



**Рис. 15.2.**Реализация ETL-процесса с использованием промежуточной области

*Размещение* извлеченных данных в промежуточной области означает *запись* данных в *БД* или файлы дисковой подсистемы. Альтернативным подходом к реализации процесса *ETL* является выполнение преобразований в оперативной памяти *ETL*-сервера и непосредственную загрузку в ХД, как показано на [рис. 15.3](https://intuit.ru/studies/courses/599/455/lecture/10175?page=1#image.15.3).



**Рис. 15.3.**Реализация ETL-процесса без использования промежуточной области

*Преобразование данных* в оперативной памяти выполняется быстрее, чем при размещении их предварительно на диске. Однако применение такого подхода лимитируется размером порции загружаемых данных. Если размер порции загружаемых данных достаточно большой, то необходимо использовать промежуточную область.

Иногда применяется еще один подход к реализации процесса *ETL*, когда *преобразование данных* выполняется на сервере ХД, в процессе их *загрузки*. Использование такого подхода определяется вычислительными возможностями сервера ХД. Обычно такой подход применяется для *MPP* серверов ХД.

В зависимости от того, *кто* извлекает данные из систем источников, реализация *ETL*-процесса может быть выполнена следующими способами.

1. *ETL*-сервер периодически подключается к системам, источникам данных, опрашивает их, извлекает результаты выполнения запросов и размещает их у себя для дальнейшей обработки.
2. Триггеры систем источников данных отслеживают изменения в данных и размещают измененные данные в отдельных таблицах, которые затем экспортируются на *ETL*-сервер.
3. Специально разработанное приложение в системах источниках данных периодически опрашивает их и экспортирует данные на *ETL*-сервер.
4. Используются log-журналы БД систем источников, которые содержат все транзакции изменения данных. Измененные данные извлекаются из log-журналов и сохраняются на сервере системы источника данных для последующего импорта в *ETL*-сервер.

В зависимости от того, *где* выполняется процесс *извлечения данных* из систем источников, реализация *ETL*-процесса может быть выполнена следующими способами.

1. *ETL*-процесс выполняется на выделенном *ETL*-сервере, который располагается между системами источниками данных и сервером ХД. В этом случае процесс *ETL* не использует вычислительных ресурсов сервера ХД и серверов систем источников данных.
2. *ETL*-процесс выполняется на сервере ХД. В этом случае сервер ХД должен иметь достаточное дисковое пространство для выполнения *ETL*-процесса, использование ресурсов сервера не должно сильно влиять на производительность запросов пользователей к ХД.
3. *ETL*-процесс выполняется на серверах систем источников данных для ХД. В этом случае изменения в данных сразу же отражаются в ХД. Такой подход используется при разработке ХД реального времени.

Таким образом, при проектировании процесса *ETL* проектировщик ХД должен на основе анализа требований к функционированию ХД совместно с руководителем ИТ-проекта выбрать программно-аппаратное решение для реализации *ETL*-процесса, а именно – точно определить, *где* и *каким способом* будет выполняться *ETL*-процесс. На это решение может сильно повлиять *бюджет проекта*. Например, может быть недостаточно финансовых средств, чтобы реализовать процесс *ETL* на выделенном сервере.

**Разработка ETL-процесса**

Как правило, при конструировании процесса *ETL* для ХД придерживаются следующей последовательности действий.

* *Планирование ETL-процесса*, которое включает в себя разработку диаграммы потоков данных от систем-источников, определение преобразований, метода генерации ключей и последовательности операций для каждой таблицы назначения.
* Конструирование процесса заполнения таблиц измерений, которое включает в себя разработку и верификацию процесса заполнения статических таблиц измерений, разработку и верификацию механизмов изменения для каждой таблицы измерений.
* Конструирование процесса заполнения таблиц фактов, которое включает в себя разработку и верификацию процесса первоначального заполнения и периодического дополнения таблиц фактов, построение агрегатов и разработку процедур автоматизации процесса *ETL*.

**Планирование ETL-процесса**

Процесс *преобразования данных* играет весьма важную роль в достижении успеха реализации проекта ХД, поэтому он должен быть хорошо спланирован. Разработка плана носит интерактивный характер.

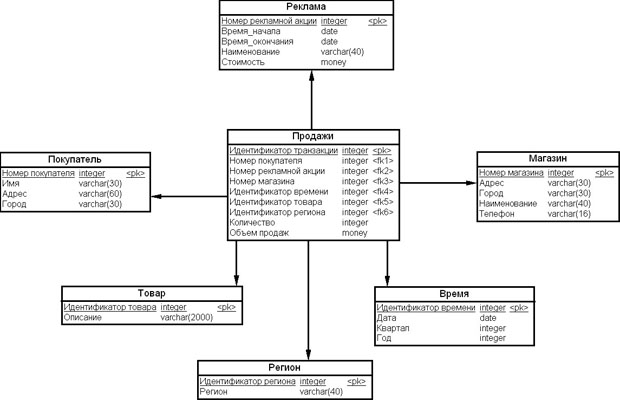
Сначала создается обобщенный план, в котором отражается перечень систем –источников данных и указываются планируемые целевые области данных (данных, которые будут размещаться в ХД). Источник целевых данных определяется на основе сформулированных бизнес-требований к ХД. Как правило, источники данных существенно различаются: от БД и текстовых файлов до SMS-сообщений. Это обстоятельство может значительно усложнить задачу *преобразования данных*.

Назначение таких высокоуровневых описаний источников дает, с одной стороны, разработчикам представление и о создаваемой системе, и о существующих источниках данных, а с другой, руководству организации, — понимание сложности, связанной с процессами *преобразования данных*.

К составлению обобщенного плана лучше всего приступать, когда разработана *многомерная модель* ХД. Тогда для каждой таблицы многомерной схемы можно определить таблицы – источники данных.

Рассмотрим пример многомерной схемы ХД системы поддержки принятия решений на [рис. 15.4](https://intuit.ru/studies/courses/599/455/lecture/10175?page=1#image.15.4).

Связи между таблицами многомерной схемы ХД и таблицами – источниками данных можно указать с помощью стрелочек, как на диаграмме на [рис. 15.5](https://intuit.ru/studies/courses/599/455/lecture/10175?page=1#image.15.5).

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_04.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_04.jpg)  
**Рис. 15.4.**Многомерная схема ХД системы поддержки принятия решений

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_05.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_05.jpg)  
**Рис. 15.5.**Планирование систем источников для ETL-процесса

Каждую стрелочку на диаграмме следует пронумеровать и сопроводить комментарием, который должен служить напоминанием разработчикам о необходимости следить за целостностью ссылочных данных или других определенных бизнес-правилами особенностях обработки каждой таблицы источника.

На этой стадии планирования необходимо зафиксировать все обнаруженные расхождения в определениях данных и схемах кодирования.

Детальное *планирование ETL-процесса* во многом зависит от использования выбранных *ETL*-инструментов. К настоящему времени разработано достаточно много таких инструментов как компаниями производителями комплексных решений в области ХД (IBM, Oracle, MicroSoft), так и сторонними производителями программного обеспечения (Sunopsis). Поэтому задача выбора подходящих *ETL*-инструментов должна быть решена до того, как приступать к детальному планированию.

Программное обеспечение этого класса предназначено для *извлечения*, приведения к общему формату, *преобразованию*, *очистки* и *загрузки данных* в хранилище. Существуют два подхода к написанию *ETL*-процедур: 1) их можно написать вручную; 2) можно воспользоваться специализированными средствами *ETL*.

Каждый из подходов имеет ряд преимуществ и недостатков, поэтому выбор того или иного метода реализации процедур *ETL* определяется требованиями к подсистеме *загрузки данных* в каждом конкретном случае. Выделим наиболее важные достоинства каждого из способов написания *ETL*-процедур.

Написание вручную:

* возможность использования широко распространенных парадигм программирования, например, объектно-ориентированного программирования;
* возможность применения многих существующих методик и программных средств, позволяющих автоматизировать процесс тестирования разрабатываемых процедур *загрузки данных* ;
* доступность человеческих ресурсов;
* возможность построения наиболее производительного решения с использованием при программировании всех преимуществ систем управления базами данных (СУБД), задействованных в проекте;
* возможность построения наиболее гибкого решения.

Применение *ETL*-инструментов:

* упрощение процесса разработки, и, главное, процесса поддержания и модификации процедур *ETL*;
* ускорение процесса разработки системы, возможность использования готовых наработок, поставляемых вместе со средствами *ETL*;
* возможность использования встроенных систем управления метаданными, позволяющих синхронизовать метаданные между СУБД, средством *ETL*, а также инструментами визуализации данных;
* возможность автоматической документации написанных процедур;
* многие средства *ETL* предоставляют собой средства увеличения производительности подсистемы *загрузки данных*, которые включают в себя возможность распараллеливания вычислений на различных узлах системы, использование хеширования и многие другие.

Следует обратить внимание на выбор технологии для реализации процедур *ETL*, в случае если одной из систем-источников данных выступает ERP-система. Системы данного класса являются наиболее сложными, так как обладают очень запутанной моделью данных и зачастую содержат десятки тысяч таблиц. Для реализации процедур *загрузки данных* из ERP-систем в команду разработчиков должен быть включен специалист, хорошо знакомый с данной системой-источником, так как анализ подобного рода систем с нуля занимает слишком длительное время. Кроме того, большинство поставщиков средств *ETL* предоставляют коннекторы ко многим ERP-системам, позволяющим импортировать метаданные ERP-систем и работать с ними на более высоком уровне. Наличие коннекторов к ERP-системам предоставляет специализированным средствам *ETL* большое преимущество над написанием вручную процедур *загрузки данных*, в случае если в качестве источника данных выступает ERP-система.

После выполнения предварительного планирования приступают к детальному планированию.

Детализированные планы *преобразования данных* составляются для всех таблиц, участвующих в процессе *преобразования*.

В настоящей лекции мы не даем рекомендаций по формированию детального плана, поскольку в каждом конкретном случае детальное планирование выполняется руководителем проекта создания ХД и включает в себя учет различных факторов, связанных со спецификой предметной области ХД. Так, например, для банковских ХД, возможно, будет необходимо выполнить проверку корректности бухгалтерской базы по бизнесправилам (провести в процессе *ETL* подсчет остатков и оборотов по отдельным лицевым и/или балансовым счетам).

**Конструирование процесса заполнения таблиц измерений**

Для таблиц измерений ХД, которые не будут изменяться со временем, в разработке процесса *ETL* первой основной задачей является выбор первичного ключа таблицы. Выбор ключа осуществляется проектировщиком ХД на основе анализа источников данных.

Второй основной задачей является проверка наличия в измерении отношений "один к одному" и "один ко многим". Как правило, для такой проверки используется сортировка.

Затем следует рассмотреть изменяющиеся измерения, определить тип изменений и описать процедуры работы с такими измерениями.

Загрузка таблиц измерений выполняется либо путем перезаписи таблицы измерения (для небольших по объему таблиц), либо загружаются только изменения в данных таблиц измерений.

**Конструирование процесса заполнения таблиц фактов**

При конструировании процесса заполнения таблиц фактов проектировщик решает следующие основные задачи:

* проанализировать построенные таблицы фактов;
* рассмотреть процесс загрузки таблиц фактов;
* рассмотреть и проанализировать построенные агрегаты;
* рассмотреть процесс загрузки агрегатов.

Процесс загрузки таблиц фактов бывает двух типов: первоначальная загрузка и периодическая загрузка изменений.

Проблемы, связанные с первоначальной загрузкой, состоят в том, что с большой долей вероятности вы не получите корректных исторических данных из-за больших объемов данных. Поэтому важно оценить, какой тип загрузки для какой таблицы фактов подходит наилучшим образом.

Отметим, что в процессе *ETL* таблицы измерения должны обновляться до обновления таблиц фактов, поскольку релевантные строки таблиц измерений должны быть размещены для загружаемых новых фактов.

**Классификация типов источников данных**

Источники данных для ХД можно разбить на следующие основные типы.

* Не корпоративные источники данных:
  + моментальные снимки -- полная копия источника;
  + специфические источники данных - унаследованные системы;
  + журналы транзакций БД источников -- переписываются только изменения;
  + наборы данных, получаемые в результате запроса SQL.
* Корпоративные источники данных:
  + реплицируемые источники данных, работающие по механизмам публикации или подписки;
  + источники, работающие по вызову (Call back sources), когда в случае изменений вызываются процедуры *ETL*-процесса;
  + источники, работающие по факту изменений данных, когда *ETL*-процесс запускается, как только данные в источнике изменились, например, по *механизму триггеров* БД.

Отметим, что стратегия *извлечения данных* в значительной степени зависит от типа источника данных.

**Элементы ETL-процесса**

**Извлечение данных**

Целью процесса *извлечения данных* является быстрое *извлечение* релевантных данных из источников данных.

Процесс *извлечения данных* из источников данных можно разбить на следующие основные типы:

* *извлечение данных* при помощи приложений, основанных на выполнении SQL-команд. Эти приложения функционируют совместно с другими приложениями систем источников данных;
* *извлечение данных* при помощи встроенных в СУБД механизмов импорта/экспорта данных. Использование таких механизмов, как правило, обеспечивает более быстрое *извлечение данных*, чем с помощью команд SQL;
* *извлечение данных* с помощью специально разработанных приложений.

Процесс *извлечения данных* может выполняться ежедневно, еженедельно или, в редких случаях, ежемесячно. Отметим, что существует целый класс систем бизнес-аналитики, которые требуют *извлечения данных* в режиме реального времени: например, системы, анализирующие биржевые операции (каждую секунду), или системы в области телекоммуникаций.

Процесс *извлечения данных* может выполняться либо в среде оперативных систем обработки данных (источников), либо в среде функционирования ХД.

**Преобразование данных**

Процесс *преобразования данных* источников включает в себя следующие основные действия.

* Преобразование типов данных:
  + *преобразования*, связанные с кодировкой данных, например, *EBCDIC* -> ASCII / UniCode;
  + преобразование строковых данных;
  + *преобразование форматов данных* для представления даты или времени.
* Преобразования, связанные с нормализацией или *денормализацией* схемы данных:
  + преобразование *денормализации* схемы с целью увеличения производительности выполнения запросов к ХД;
  + нормализация схемы ХД с целью обеспечения простоты SQL-запросов.
* Преобразования ключей, связанные с обеспечением соответствия бизнес-ключей суррогатным ключам ХД.
* Преобразования, связанные с обеспечением *качества данных* в ХД.

Как правило, данные источников не обладают необходимым уровнем *качества данных*. Заметим, что данные в ХД должны быть:

* точными – данные должны содержать правильные количественные значения метрик или давать объяснения, почему невозможно такие значения иметь;
* полными – пользователи ХД должны знать, что имеют доступ ко всем релевантным данным;
* согласованными – никакие противоречия в данных не допускаются: агрегаты должны точно соответствовать подробным данным;
* уникальными – одни и те же объекты предметной области должны иметь одинаковые наименования и идентифицироваться в ХД одинаковыми ключами;
* актуальными – пользователи ХД должны знать, с какой частотой данные обновляются (т.е. на какую дату данные действительны).

Для обеспечения качества данные при *преобразовании* подвергаются процедуре *очистки*. Процедура *очистки данных* необходима, поскольку системы бизнес-аналитики не работают с несогласованными и неточными данными, иначе бизнес-анализ становится бессмысленным.

Процедура *очистки данных* включает в себя согласование форматов данных, кодирование данных, исключение ненужных атрибутов (например, комментариев), замещение кодов значениями (например, почтового индекса наименованием населенного пункта), комбинирование данных из различных источников под общим ключом (например, собрать все данные о покупателях), обнаружение одинаково поименованных атрибутов, которые содержат различные по смыслу значения.

*Очистку данных* можно разделить на следующие типы:

* конвертация и нормализация данных (приведение к одинаковому кодированию текста, форматам даты и т. д.);
* стандартизация написания имен, представления адресов, устранение дубликатов;
* стандартизация наименований таблиц, индексов и т.д.;
* очистка, основанная на бизнес-правилах предметной области.

Процедуры *очистки* также включают в себя создание меток статуса фактов в таблицах измерения (нормальный, ненормальный, невозможный, выходящий за границы, анализируемый или нет и т.д.), распознавание случайных и зашумленных значений (замещение их NULL-значением или оценкой), унификация использования NULL-значений, маркирование фактов с изменившимся статусом (например, покупатель аннулировал контракт), агрегирование фактов и т.п.

**Загрузка данных**

Основная цель процесса *загрузки данных* состоит в быстрой *загрузке данных* в ХД. Отметим некоторые особенности выполнения процесса *загрузки данных* в ХД.

Во-первых, *загрузка данных*, основанная на использовании команд обновления SQL, является медленной. Каждая команда SQL выполняется СУБД по определенному плану выполнения, и ее обработка включает выполнение нескольких фаз. Поэтому *загрузка* с помощью встроенных в СУБД средств импорта/экспорта является предпочтительной.

Во-вторых, индексы таблиц загружаются медленно. Во многих случаях целесообразно удалить индекс и построить его заново.

В-третьих, следует максимально использовать параллелизм при *загрузке данных*. Измерения могут производиться одновременно с фактами и *секциями таблиц*. Аналогично факты и *секции таблиц* могут загружаться одновременно с измерениями.

Следует заметить, что при *загрузке данных* должна быть гарантирована ссылочная целостность данных, а агрегаты должны быть построены и загружены одновременно с подробными данными.

Настройка производительности *загрузки данных* в ХД выполняется администратором ХД с помощью набора процедур, предусмотренных используемой СУБД.

Таким образом, мы рассмотрели в общих чертах основные *элементы ETL-процесса*. Теперь разберем, как проектировать *ETL*-процессы.

**Проектирование процессов преобразования данных с помощью CASE-инструментов**

В компьютерных системах организации *информация* хранится в различных *БД* систем оперативной обработки данных, корпоративных приложениях и ХД. Для использования этой информации необходима ее систематизация. Систематизация информации приводит к решению задач комбинирования информации из различных источников, преобразованию ее к единому формату для производства корпоративной отчетности и *интеллектуального анализа данных*.

Для решения поставленной выше задачи используются две технологии – **технология ETL-процессов и технология процессов интеграции данных**, или технология *EII*- (*Enterprise* *Information* Integration) процессов.

Как уже указывалось в этой лекции, *ETL* является процессом в складировании данных, который выполняет три функции в одной программной среде: *чтение данных* из определенных источников ( *извлечение* ), *преобразование данных* в соответствии с принятыми соглашениями (преобразование) и *запись* преобразованных данных в источник назначения ( *загрузка* ).

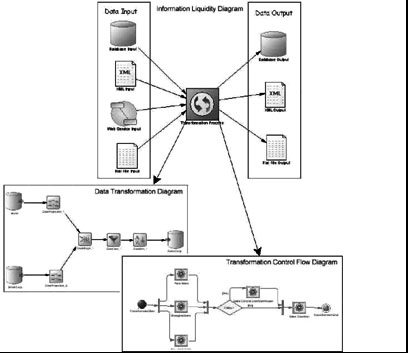
*EII* комбинирует необработанные данные посредством управления и организации запросов к различным системам без предварительной агрегации контента. Полученные данные не помещаются, как правило, на хранение в какие-либо *БД* или ХД.

*CASE*-инструменты предоставляют графическую среду для моделирования и документирования *ETL*- и *EII*-процессов. В *CASE* PowerDesigner для этого служит модель движения данных (*Information* Liquidity *Model*, или *ILM*).

*ILM* предоставляет проектировщику ХД средства моделирования и анализа *преобразования данных* как на общем уровне рассмотрения, так и на детальном уровне рассмотрения.

* ***Диаграмма движения данных*** (Information liquidity diagram) - это диаграмма высокого уровня, которая позволяет моделировать *преобразование данных* посредством определения:
  + **входных источников данных** (БД, XML-документов, бизнес-процессов, неструктурированных файлов);
  + **процесса преобразования** (Transformation process), позиции, где преобразования выполняются. Процесс преобразования включает в себя набор диаграмм преобразования (transformation diagrams), которые описывают *преобразования данных* на более детальном уровне;
  + **выходных источников данных** (БД, XML-документов, неструктурированных файлов).
* ***Диаграмма преобразования данных*** (*Data transformation* diagram) - это диаграмма низкого уровня, которая позволяет моделировать конкретные задачи преобразования посредством определения того, как данные извлекаются из входного источника данных, как преобразуются и как загружаются в выходной источник данных.
* **Диаграмма управления преобразованием** (Transformation *control flow* diagram) - это диаграмма низкого уровня модели, которая позволяет определить последовательность выполнения серии задач преобразования.

На [рис. 15.6](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.6) приведен пример того, как могут быть связаны входные и выходные источники данных на высоком уровне и как преобразования моделируются на диаграммах более низкого уровня.



**Рис. 15.6.**Диаграммы преобразования данных

**Моделирование движения данных (Information Liquidity Modeling)**

Движение данных — один из главных вопросов для любой организации. Моделирование движения данных позволяет определить в модели, где данные генерируются, где загружаются в систему и как они преобразуются в процессе их использования.

PowerDesigner **Information Liquidity Model (ILM) является графическим инструментом моделирования движения данных, который позволяет построить модели**:

* репликации данных (Data replication) – данные реплицируются из БД источника в БД назначения посредством механизма репликации, такого как *Replication Server*;
* *преобразования данных* (*Data transformation*) – данные из различных источников объединяются для того, чтобы быть извлеченными, преобразованными и загруженными в другой источник посредством процесса *ETL* (Extract Transform and Load) и *EII* (Enterprise Information Integration).

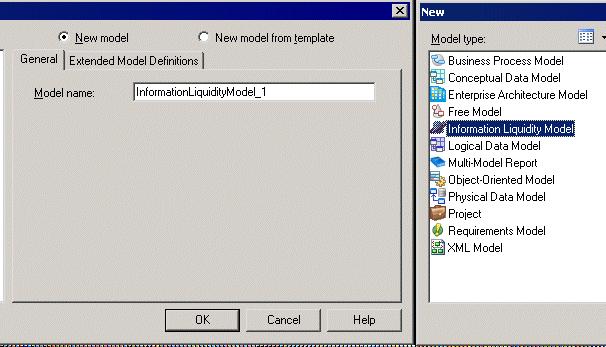
Теперь построим модель движения данных (*ILM*), содержащую *диаграмму движения данных* (Information Liquidity Diagrams), диаграмму управления преобразованием (Transformation *Control Flow* Diagrams) и *диаграммы преобразования данных* (*Data Transformation* Diagrams), и создадим тем самым модель *ETL*-процесса.

**Создание ILM-модели**

Для того чтобы создать *ILM*-модель, необходимо выполнить следующую последовательность действий.

1. Выбрать пункт меню **File -> New**

В результате на экране будет показано диалоговое окно, как на [рис. 15.7](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.7).



**Рис. 15.7.**Создание ILM-модели

1. В качестве типа модели выберите из списка моделей **Information Liquidity Model**.
2. Установите поле "Новая модель" ( **New Model** ) для создания новой стандартной модели.
3. В текстовом поле "Имя модели" ( **Model Name** ) наберите имя модели.
4. Щелкните правой кнопкой мыши на кнопке **OK**.

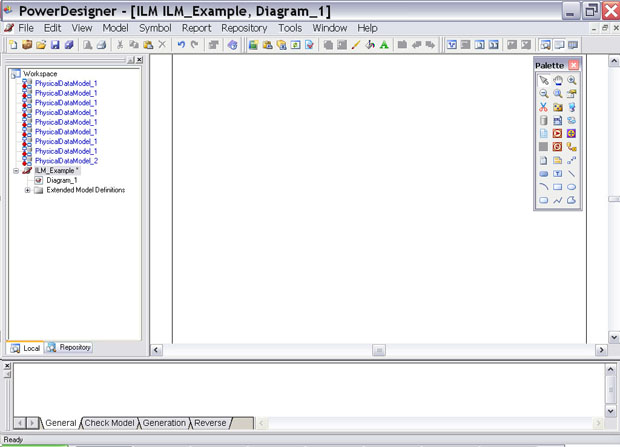
Таким образом, *ILM*-модель создана. Теперь ее нужно наполнить диаграммами.

Создадим диаграмму преобразования. Диаграмма преобразования является диаграммой высокого уровня для документирования *ETL*- и *EII*-процессов. На ней показываются входные и выходные источники данных, вовлекаемые в процесс преобразования. Источники связываются процессом преобразования, которое определяется на последующих *диаграммах преобразования данных*.

Предположим, в наше ХД Giant Corp данные поступают из следующих источников: двух оперативных БД системы организации Small Corp и Acme, модели бизнес-процессов Small Corp Role и XML-модели Acme Role.

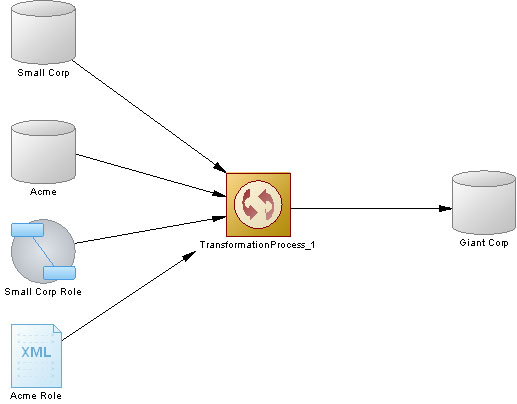
С помощью палитры инструментов ([рис. 15.8](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.8)) разметим объекты модели.

|  |  |
| --- | --- |
| **Действие** | **Пиктограмма** |
| три источника данных типа БД |  |
| один источник данных типа "XML-документ" |  |
| один источник данных типа "Бизнес-процесс" |  |
| установим "Процесс преобразования" |  |

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_08.jpg)

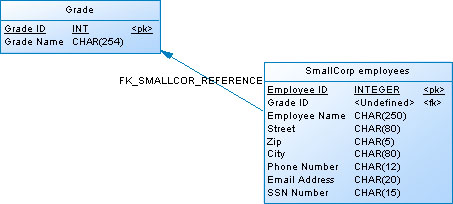
[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_08.jpg)  
**Рис. 15.8.**Палитра инструментов

Установим связи между ними с помощью стрелок и таким образом получим диаграмму преобразования, как показано на [рис. 15.9](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.9).

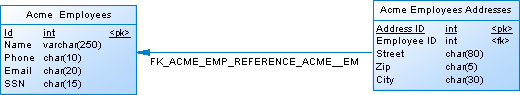


**Рис. 15.9.**Диаграмма преобразования

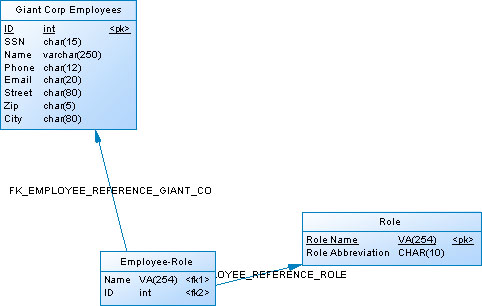
Теперь необходимо связать объекты диаграммы с описанием источников данных. Пусть для нашего примера БД Small Corp имеет схему, как показано на [рис. 15.10](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.10), БД Small Corp — схему, как на [рис. 15.11](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.11), ХД Giant Corp — как на [рис. 15.12](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.12), Acme Role XML-документ имеет модель, как показано на [рис. 15.13](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.13).



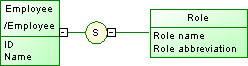
**Рис. 15.10.**



**Рис. 15.11.**

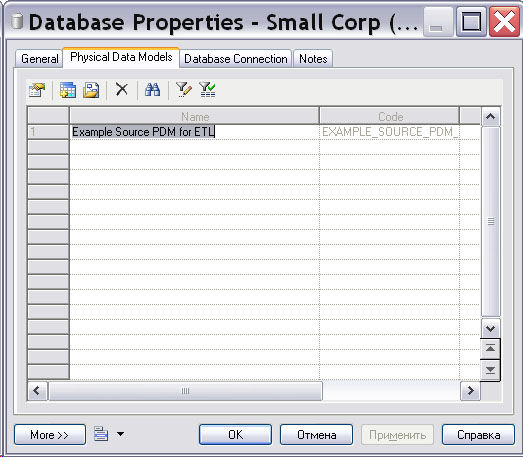


**Рис. 15.12.**



**Рис. 15.13.**

Покажем на примере БД Small Corp, как связать объект диаграммы с описанием источника данных. Дважды щелкнем левой кнопкой на пиктограмме объекта, на панели " **Database property sheet** " откроем вкладку " **Physical Data Models** ". Щелкнем кнопкой мыши на пиктограмме " **Add Physical Data Models** ", откроется список моделей. Выберем из списка соответствующую модель и щелкнем кнопкой мыши на **ОК** (см. [рис. 15.14](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.14)). Связь установлена.



**Рис. 15.14.**Установление связи объекта диаграммы ILM-модели со схемой БД

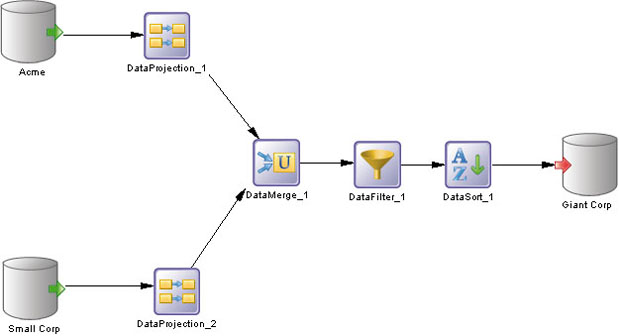
**Моделирование процесса преобразования данных**

**Диаграмма преобразования данных**

Объект *ILM*-диаграммы "Процесс преобразования" определяет список задач *преобразования данных* и поток управления преобразованием.

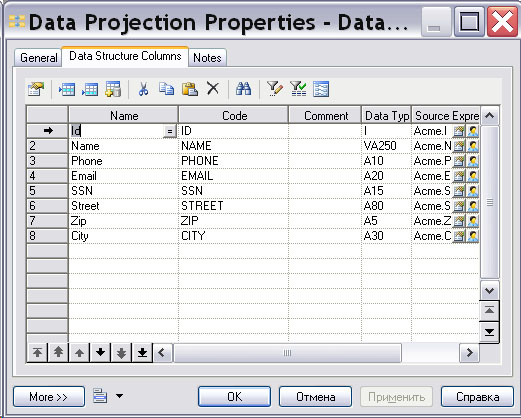
Построим *диаграмму преобразования данных*. Задача *преобразования данных* включает одну или несколько *диаграмм преобразования данных*, которые позволяют моделировать процессы *извлечения данных* из источников данных, *преобразования данных* и *загрузки данных* в источник назначения. Данные поступают из источников данных (*data inputs*), преобразуются посредством выполнения определенных манипуляций над ними и загружаются в источник назначения (data outputs). Последовательность этих процессов связывается вместе посредством потока данных (data flows), а сами данные представляются посредством колонок структуры данных, которые используются в каждом из перечисленных процессов.

На [рис. 15.15](https://intuit.ru/studies/courses/599/455/lecture/10175?page=3#image.15.15), на *диаграмме преобразования данных*, показано, что данные извлекаются из БД Acme и Small Corp, затем объединяются в БД DataMerge, фильтруются фильтром DataFilter, сортируются процедурой DataSort и загружаются в БД Giant Corp.

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_15.jpg)

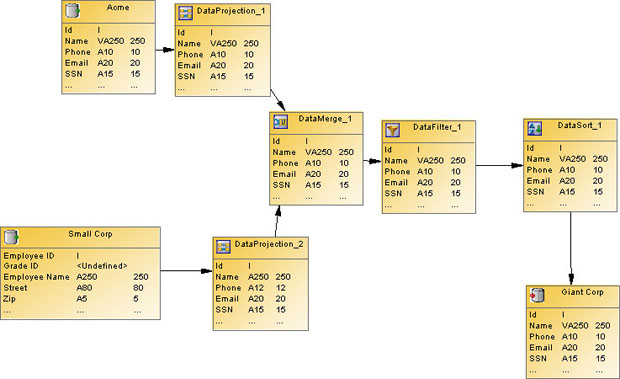
[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_15.jpg)  
**Рис. 15.15.**Диаграмма преобразования данных

Для каждого объекта нужно определить перечень колонок, который будет задействован в процессе преобразования на этом объекте. Эти действия выполняются на диалоговом окне " **Data Projection Properties** "" как показано на [рис. 15.16](https://intuit.ru/studies/courses/599/455/lecture/10175?page=3#image.15.16).



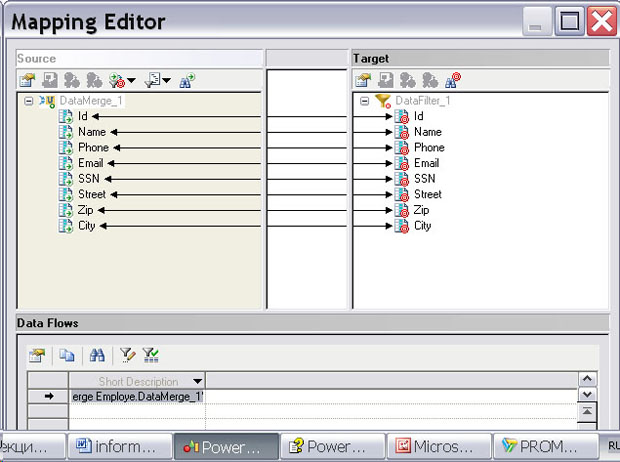
**Рис. 15.16.**Определение структуры данных объекта

Изображенная на [рис. 15.15](https://intuit.ru/studies/courses/599/455/lecture/10175?page=3#image.15.15) *диаграмма преобразования данных* может быть развернута таким образом, чтобы видеть структуры данных на каждом элементе диаграммы, как показано на [рис. 15.17](https://intuit.ru/studies/courses/599/455/lecture/10175?page=3#image.15.17).

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_17.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_17.jpg)  
**Рис. 15.17.**Диаграмма преобразования данных со структурами данных

Поток данных (стрелочки на диаграмме) передает данные между объектами *диаграммы преобразования данных*. Для определения потока данных может быть использован редактор отображения (Mapping Editor), как показано для потока данных между **DataMerge** и **DataFilter** на [рис. 15.18](https://intuit.ru/studies/courses/599/455/lecture/10175?page=3#image.15.18).

[](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_18.jpg)

[увеличить изображение](https://intuit.ru/EDI/20_07_20_2/1595197216-9970/tutorial/632/objects/15/files/10_18.jpg)  
**Рис. 15.18.**Редактор отображения (Mapping Editor)

**Диаграмма управления потоком преобразования данных**

*Диаграммы управления потоком преобразования данных* предназначены для моделирования последовательных или параллельных задач *преобразования данных*. *Диаграмма управления потоком* содержит одну или несколько *диаграмм управления потоком преобразования*, моделирующих порядок, в котором последовательность задач *преобразования данных* выполняется. Последовательность задач включает в себя инициализацию задачи (starts), выполнение задачи (task executions), решение (decisions), синхронизацию (synchronizations) и завершение задачи. Связанные вместе, они составляют поток управления.

На [рис. 15.19](https://intuit.ru/studies/courses/599/455/lecture/10175?page=3#image.15.19) показана *диаграмма управления потоком преобразования данных* для нашего примера.

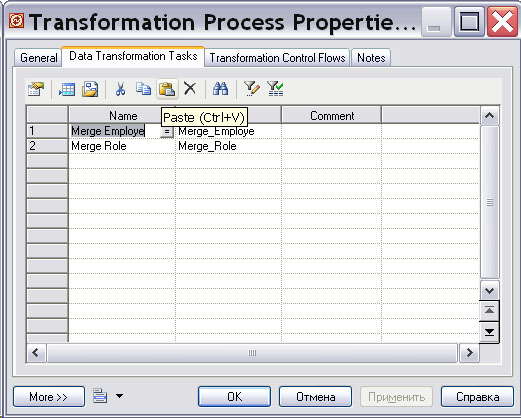
Диаграмма управления потоком преобразования данных

**Рис. 15.19.**Диаграмма управления потоком преобразования данных

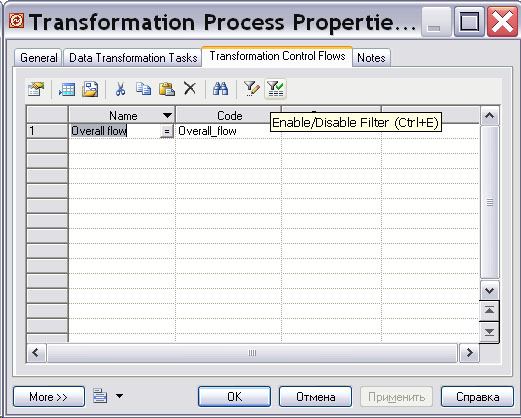
Пиктограмма инициализации задачи преобразования ( **TransformationStart\_1** ) инициирует выполнение последовательности задач *преобразования данных* на диаграмме управления потоком. Далее последовательно выполняются задачи "Объединить сотрудников" ( **Merge Employe** ) и "Объединить роли" ( **Merge Role** ). Решений никаких не принимается, синхронизация не выполняется. Процесс заканчивается выполнением задачи завершения процесса преобразования ( **TransformationEnd\_1** ).

Все построенные диаграммы связаны между собой на диаграмме высокого уровня *ILM*-модели (см. [рис. 15.9](https://intuit.ru/studies/courses/599/455/lecture/10175?page=2#image.15.9)) в объекте "Процесс преобразования" ( **TransformationProcess\_1** ). На [рис. 15.20](https://intuit.ru/studies/courses/599/455/lecture/10175?page=3#image.15.20) и [10.21](https://intuit.ru/studies/courses/599/455/lecture/10175?page=3#image.15.21) соответственно изображены вкладки диалогового окна свойств этого объекта, на которых показана привязка задач преобразования и потока управления.

Таким образом, проектировщик ХД данных может моделировать *ETL*-процессы для разрабатываемого хранилища данных с помощью CASE-инструментов.



**Рис. 15.20.**Задачи преобразования в объекте "Процесс преобразования" ILM-модели



**Рис. 15.21.**Поток управления преобразованием данных в объекте "Процесс преобразования" ILM-модели

Заметим, что при изложении материала последнего раздела мы опустили многочисленные детали определения объектов и их свойств при проектировании модели *ETL*-процесса. Нашей задачей в данном случае является иллюстрация возможностей CASE-инструментов. Отметим также, что ведущие компании, производители комплексных решений для создания ХД, такие как IBM, Oracle, MicroSoft и ряд других, поставляют встроенные средства для проектирования *ETL*-процессов. Поэтому проектировщику ХД будет необходимо ознакомиться с возможностями таких инструментов после выбора несущей СУБД.

**Резюме**

Таким образом, разработка *ETL*-процесса включает в себя следующие основные стадии:

* *планирование ETL-процесса* ;
* конструирование процесса заполнения таблиц измерений;
* конструирование процесса заполнения таблиц фактов;
* *извлечение данных* ;
* *преобразование* и *очистка данных* ;
* *загрузка данных*.

При проектировании процессов *преобразования данных* проектировщик ХД должен решить следующие задачи:

* проанализировать требования к данным ХД;
* проанализировать и описать источники данных для ХД;
* создать модель *преобразования данных* высокого уровня;
* определить и подробно описать каждую задачу *преобразования данных* ;