

**ERwin - средство разработки
структуры базы данных (БД)**

**Model Mart - система управления
моделями для групповой разработки
при создании приложений для
архитектуры "клиент-сервер",
хранилищ данных, Web**

ERwin — графический инструмент для моделирования баз данных, создания и поддержки баз, витрин (data marts) и хранилищ данных, а также моделей ресурсов данных предприятия. Модели ERwin визуализируют структуры данных для облегчения организации и управления данными, упрощения сложных взаимосвязей данных, а также технологий создания баз данных и среды развертывания.

ERwin — это не просто средство проектирования, но и инструмент разработки, способный автоматически создавать таблицы и генерировать тысячи строк текста хранимых процедур и триггеров для всех популярных СУБД.

ERwin поддерживает две методологии визуального моделирования данных:

1) IDEF1X (Integration Definition for Information Modeling — интегрированное описание информационных моделей). IDEF1X — высокоструктурированная методология моделирования;

2) IE (Information Engineering — информационная инженерия).

Основные компоненты диаграммы Erwin - это сущности, атрибуты и связи.

Каждая сущность является множеством подобных индивидуальных объектов, называемых экземплярами. Каждый экземпляр индивидуален и должен отличаться от всех остальных экземпляров. Атрибут выражает определенное свойство объекта. С точки зрения БД (физическая модель) сущности соответствует таблица, экземпляру сущности - строка в таблице, а атрибуту - колонка таблицы.

Построение модели данных предполагает определение сущностей и атрибутов, т. е. необходимо определить, какая информация будет храниться в конкретной сущности или атрибуте. Сущность можно определить как объект, событие или концепцию, информация о которых должна сохраняться. Сущности должны иметь наименование с четким смысловым значением, именоваться существительным в единственном числе, не носить "технических" наименований и быть достаточно важными для того, чтобы их моделировать. Именование сущности в единственном числе облегчает в дальнейшем чтение модели. Фактически имя сущности дается по имени ее экземпляра. Примером может быть сущность *Заказчик*

Связь является логическим соотношением между сущностями. Каждая связь должна именоваться глаголом или глагольной фразой (Relationship Verb Phrases).

Имя связи выражает некоторое ограничение или бизнес-правило и облегчает чтение диаграммы, например:

Каждый КЛИЕНТ *<размещает>* ЗАКАЗы;

Каждый ЗАКАЗ *<выполняется>* СОТРУДНИКОМ.

Связь показывает, какие именно заказы разместил клиент и какой именно сотрудник выполняет заказ.



ERwin – средство разработки структуры базы данных (БД)

Процесс построения информационной модели состоит из следующих шагов:

- определение сущностей;
- определение зависимостей между сущностями;
- задание первичных и альтернативных ключей;
- определение атрибутов сущностей;
- приведение модели к требуемому уровню нормальной формы;
- переход к физическому описанию модели: назначение соответствий имя сущности - имя таблицы, атрибут сущности - атрибут таблицы;
- задание триггеров, процедур и ограничений;
- генерация базы данных.

Отображение логического и физического уровня модели данных в ERwin

В ERwin существуют два уровня представления и моделирования - логический и физический.

Логический уровень означает прямое отображение фактов из реальной жизни. Например, люди, столы, отделы, собаки и компьютеры являются реальными объектами. Они именуется на естественном языке, с любыми разделителями слов (пробелы, запятые и т.д.). На логическом уровне не рассматривается использование конкретной СУБД, не определяются типы данных (например, целое или вещественное число) и не определяются индексы для таблиц.

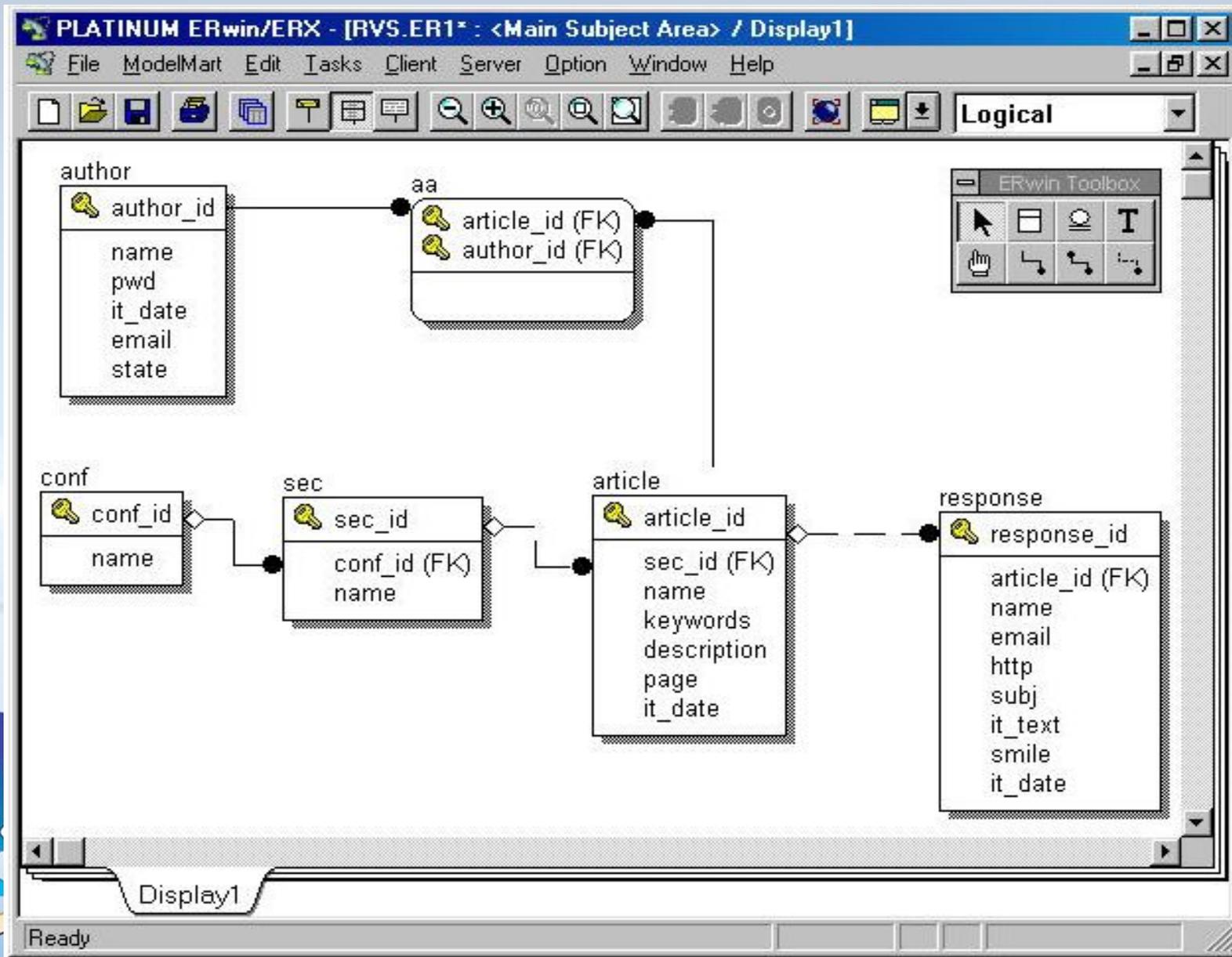
Физический уровень — это по существу отображение системного каталога, который зависит от конкретной реализации СУБД.

ERwin предоставляет возможности создавать и управлять этими двумя различными уровнями представления одной диаграммы (модели).

В логической модели отображаются сущности, атрибуты и отношения между сущностями. Для работы с MySQL в ERWin нам требуется получить только красивую картинку структуры базы данных.

В физической модели можно выбрать конкретную СУБД и в зависимости от этой СУБД настроить типы атрибутов.





Entity tool



Инструмент для создания сущностей.

Select Tool



С помощью инструмента выбора (Select tool) можно перемещать сущности вместе с атрибутами. Имеется также инструмент для перемещения атрибутов - Attribute Manipulation Tool.



Attribute Manipulation Tool

Select Tool



Для создания отношений, в которых определено, как соотносятся записи одной таблице по отношению к другой используется инструмент - identifying relationship.



Для создания отношений, в которых не определено, как соотносятся записи используется - non-identifying relationship.

Attribute Manipulation Tool

Компоненты диаграммы ERwin и основные виды представлений диаграммы

Диаграмма ERwin строится из трех основных блоков - сущностей, атрибутов и связей.

Выбор между логическим и физическим уровнем отображения осуществляется через линейку инструментов или меню. Внутри каждого из этих уровней есть следующие режимы отображения:

- Режим "сущности" - внутри прямоугольников отображается имя сущности (для логической модели) или имя таблицы (для физического представления модели).

- Режим "определение сущности" служит для презентации диаграммы другим людям.

- Режим "атрибуты". При переходе от предметной области к модели требуется вводить информацию о том, что составляет сущность. Этот режим является основным при проектировании на логическом и физическом уровнях.

- Режим "первичные ключи" - внутри прямоугольников - сущностей показывается только атрибуты/колонки, составляющие первичный ключ.

- Режим "пиктограммы". Для презентационных целей каждой таблице может быть поставлена в соответствие пиктограмма (bitmap).

- Режим "показ глагольной фразы". На дугах связей показываются глагольные фразы, связывающие сущности (для логического уровня) или имена внешних ключей (для физического уровня).



Инструменты для создания модели в ERwin

Основные инструменты создания модели доступны как из меню, так и через окно инструментов. С их помощью создаются независимые и зависимые сущности, идентифицирующие и неидентифицирующие связи, полные и неполные категории, неспецифические связи и текстовые элементы.

Нажатием мыши над сущностью производится вход в один из многочисленных редакторов ERwin:

- редакторы, связанные с сущностью в целом (определение сущности, дополнительная информация, триггеры, индексы, характеристики таблицы, хранимые процедуры, связанные с таблицей);

- редакторы атрибутов (определение атрибутов, колонки таблицы в физическом представлении модели, репозитарий средства 4GL, например, расширенные атрибуты в PowerBuilder).

Идентификация сущностей. Сущности в ERwin

Сущность представляет собой множество реальных или абстрактных объектов, например, люди, места, события, факты, которые имеют общие характеристики.

Сущность - это логическое понятие.

Сущности соответствует таблица в реальной СУБД.

В ERwin сущность визуально представляет три основных вида информации:

- атрибуты, составляющие первичный ключ;
- неключевые атрибуты;
- тип сущности (независимая/зависимая).

Первичный ключ - это атрибут или набор атрибутов, уникально идентифицирующий экземпляр сущности.

Для каждого первичного ключа ERwin создает при генерации структуры БД уникальный индекс.

Связи (relationships) в ERwin

Связь - это функциональная зависимость между двумя сущностями (в частности, возможна связь сущности с самой собой). Например, важно знать фамилию сотрудника, и не менее важно знать, в каком отделе он работает. Таким образом, между сущностями "отдел" и "сотрудник" существует связь "состоит из" (отдел состоит из сотрудников).

Связь - это понятие логического уровня, которому соответствует внешний ключ на физическом уровне.

В ERwin связи представлены пятью основными элементами информации:

- тип связи (идентифицирующая, неидентифицирующая, полная/неполная категория, неспецифическая связь);
- родительская сущность;
- дочерняя (зависимая) сущность;
- мощность связи (cardinality);
- допустимость пустых (null) значений.

Связь называется идентифицирующей, если экземпляр дочерней сущности идентифицируется через ее связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в первичный ключ дочерней сущности. Дочерняя сущность при идентифицирующей связи всегда является зависимой.

Связь называется неидентифицирующей, если экземпляр дочерней сущности идентифицируется иначе, чем через связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в состав неключевых атрибутов дочерней сущности.

Идентифицирующая связь изображается сплошной линией; неидентифицирующая - пунктирной линией. Линии заканчиваются точкой со стороны дочерней сущности.

При определении связи происходит миграция атрибутов первичного ключа родительской сущности в соответствующую область атрибутов дочерней сущности. Поэтому такие атрибуты не вводятся вручную.

Атрибуты первичного ключа родительской сущности по умолчанию мигрируют со своими именами.

ERwin позволяет ввести для них роли, т.е. новые имена, под которыми мигрирующие атрибуты будут представлены в дочерней сущности. В случае неоднократной миграции атрибута такое переименование необходимо. Например, сущность "посредническая сделка" имеет атрибут "код предприятия-продавца" и "код предприятия-покупателя". В данном случае первичный ключ сущности "предприятие" ("код предприятия") имеет две роли в дочерней сущности.

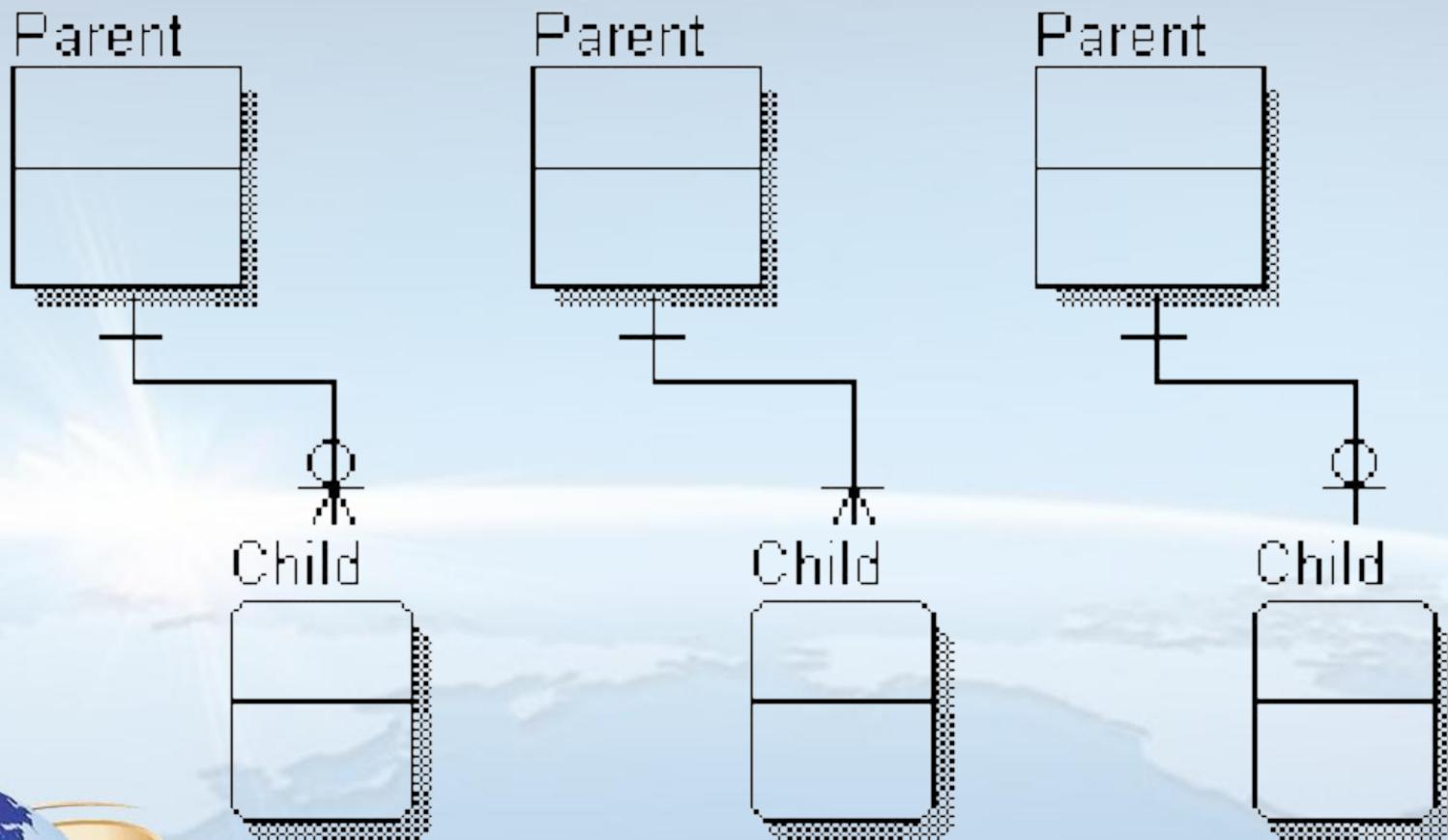
На физическом уровне имя роли - это имя колонки внешнего ключа в дочерней таблице.

Мощность связи представляет собой отношение количества экземпляров родительской сущности к соответствующему количеству экземпляров дочерней сущности. Для любой связи, кроме неспецифической, эта связь записывается как 1:n.

ERwin в соответствии с методологией IDEF1X предоставляет 4 варианта для n, которые изображаются дополнительным символом у дочерней сущности: ноль, один или больше (по умолчанию); ноль или один; равно N, где N - конкретное число.



Обозначения мощности связи в нотации IE



Графическое редактирование модели

Все объекты модели ERwin могут редактироваться средствами, принятыми в Windows - группировка, копирование, удаление, перемещение, использование системного буфера. Установка цветов и шрифтов осуществляется в удобных диалогах.

Компоненты модели, представленные текстом (имена сущностей, атрибутов, текстовые элементы) могут редактироваться непосредственно на экране.



Альтернативные ключи

Альтернативный ключ - это атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр сущности. Например, для сущности служащий (идентификатор служащего, фамилия, имя, отчество) группа атрибутов "фамилия", "имя", "отчество" может являться альтернативным ключом (в предположении, что на предприятии не работают полные тезки).

Для альтернативного ключа, как и для первичного, ERwin автоматически создает индексы при генерации БД.



Инвертированные индексы

Атрибуты, составляющие альтернативный ключ, однозначно (уникально) идентифицируют экземпляры сущности. В ERwin можно также составлять группы атрибутов, которые не идентифицируют уникально экземпляры сущности, но часто используются для доступа к данным. Для каждой такой группы атрибутов ERwin создает неуникальные индексы.

Одни и те же атрибуты сущности могут входить в несколько различных групп ключей.



Унификация атрибутов

Зависимая сущность может наследовать один и тот же внешний ключ от более чем одной родительской сущности, или от одной и той же родительской сущности через несколько связей. Если не введены различные роли для такого множественного наследования, ERwin считает, что в зависимой сущности атрибуты внешнего ключа появляются только один раз.

Унификация - это объединение двух или более групп атрибутов внешних ключей в один внешний ключ (группу атрибутов), в предположении, что значения одноименных атрибутов в дочерней сущности всегда одинаковы.



Реализация ссылочной целостности с помощью ERwin

Ссылочная целостность - это обеспечение требования, чтобы значения внешнего ключа экземпляра дочерней сущности соответствовали значениям первичного ключа в родительской сущности.

Ссылочная целостность может контролироваться при всех операциях, изменяющих данные (INSERT/UPDATE/DELETE). Средства контроля ссылочной целостности в ERwin включают автоматическую генерацию триггеров и использование механизмов декларативной ссылочной целостности (для тех СУБД, которые поддерживают данные механизмы).



Для каждой связи на логическом уровне могут быть заданы требования по обработке операций INSERT/UPDATE/DELETE для родительской и дочерней сущности.

ERwin представляет следующие варианты обработки этих событий:

- отсутствие проверки;
- проверка допустимости;
- запрет операции;
- каскадное выполнение операции (DELETE/UPDATE);
- установка пустого (NULL-значения) или заданного значения по умолчанию.



Хранение информации в модели ERwin

Обычно модели ERwin сохраняются на диск в виде файла. Имеется возможность хранить модель в целевой СУБД. Для этого с помощью самого ERwin в целевой СУБД создается метабаза ERwin. В этой базе данных сохраняется информация модели. В частном случае базой данных могут быть и dBase-файлы, с которыми ERwin работает через ODBC.



Model Mart - система управления моделями для групповой разработки при создании приложений для архитектуры "клиент-сервер", хранилищ данных, Web

Model Mart - система управления моделями для групповой разработки при создании приложений для архитектуры "клиент-сервер", хранилищ данных, Web. Обеспечивает многопользовательский доступ к моделям, созданным с помощью ERwin и VPwin. Модели хранятся на центральном сервере и доступны для всех участников группы проектирования, при этом обеспечивается возможность коллективного создания сложных и объемных моделей.

Основные особенности ModelMart:

- поддерживается несколько версий модели, участники коллектива проектировщиков имеют доступ к самой последней версии модели и могут видеть результаты сделанных изменений;
- используются гибкие средства разграничения доступа;
- пользователь может получить список различий между двумя версиями модели и, при необходимости, вернуться к предыдущей версии;
- производится анализ влияния вносимых в модель изменений;
- поддерживается возможность слияния нескольких моделей в одну;
- обеспечивается гибкая блокировка доступа к моделям;
- специальное средство интеллектуального разрешения конфликтов автоматически идентифицирует любые конфликты, возникающие при одновременной модификации одной и той же модели несколькими пользователями.

ModelMart построен по правилам архитектуры "клиент-сервер": устанавливается на сервере в среде СУБД, в качестве которой в настоящее время могут выступать Sybase, MS SQL Server, Oracle , Informix.

В качестве клиента могут выступать ERwin/ERX for ModelMart, ERwin/Navigator for ModelMart версий 2.6 и выше, и VPwin версий 2.01 и выше.

Модели в **ModelMart** хранятся в определяемых пользователем библиотеках. Все модели, хранящиеся в одной библиотеке, имеют в совместном владении некоторое множество объектов: это домены, триггеры, шаблоны для них, правила проверки и физические свойства. При создании в этой библиотеке новой модели, она автоматически получает доступ к этим разделяемым объектам. При внесении изменений в такой объект, эти изменения отображаются во всех моделях библиотеки, обеспечивая непротиворечивость проекта.

ModelMart поддерживает одновременную работу с моделями данных нескольких пользователей. Перед началом работы с моделью ModelMart сохраняет ее исходное состояние, которое и будет затем использоваться для сравнения изменений, внесенных конкретным пользователем, с изменениями, сделанными другими.

При одновременном доступе к модели, компонент "Интеллектуальное разрешение конфликтов" автоматически идентифицирует любые конфликты, которые могут возникнуть при попытках одновременно модифицировать модель.

Поддерживается возможность хранить несколько версий одной и той же модели.



ModelMart включает специальную утилиту - **ModelMart Synchronizer**, позволяющую проводить синхронизацию моделей процессов (**BPwin**) и данных (**ERwin**), хранящихся в библиотеках ModelMart.

Разграничение доступа осуществляется на основе специальных профилей доступа. Каждому пользователю могут быть назначены predetermined, встроенные в ModelMart, и специальные профили, настроенные для конкретной ситуации. Изменения, которые может внести конкретный пользователь, могут быть ограничены уровнем библиотеки, диаграммы или предметной области в пределах диаграммы. Это дает возможность строго разграничить ответственность членов коллектива.



Проблемы создания Хранилищ данных

В 90-е годы западными корпорациями накоплен значительный опыт создания и внедрения Хранилищ данных (DWH).

Этот опыт показал, что в случае успешного внедрения, проекты, связанные с DWH окупаются и приносят прибыль. Однако процент неудачных проектов очень велик. По некоторым оценкам он составляет 60–80 процентов. При внедрении Хранилищ возникает множество организационных, методических и технических трудностей, преодоление которых часто занимает месяцы, а иногда и годы. В результате превышает бюджет, иссякает терпение персонала и руководства.

Одна из причин неудач состоит в том, что Хранилище данных, также как в семидесятые годы БД, рассматривалось как готовый продукт, а не как средство разработки. Поэтому не принималось в расчет, что для проектирования и разработки Хранилища — БД, запросов, интерфейсов, правил извлечения, очистки и загрузки данных — с использованием универсальных инструментов необходимо десятки человеко-лет труда профессиональных проектировщиков и программистов. Вторая причина — каждая организация, внедряющая DWH, становилась первопроходцем в создании DWH в своей отрасли, и методом проб и ошибок искала верные методические решения.

Универсальные инструменты для разработки DWH

Хранилища данных строятся на одной из трех платформ, или их совокупности:

- Реляционные СУБД (DB2, MS SQL, Oracle и т.д.),
- Специальные платформы (Sybase IQ, RedBrick, Teradata и т.д.),
- OLAP-серверы (Hyperion Essbase, IBM OLAP Server, MS Analysis Services, Oracle Express и т.д.).

Классическая архитектура DWH состоит из следующих элементов: реляционная, многомерная, или гибридная БД, средства извлечения, очистки и загрузки данных, средства визуализации данных и генерации отчетов (OLAP-клиенты). Реляционная БД строится по архитектуре «звезда», в которой с одной таблицей фактов связаны несколько таблиц измерений (справочников), или снежинка, отличающаяся наличием иерархических справочников.

Универсальные инструменты покрывают все аспекты проектирования, создания и эксплуатации классических DWH на любых платформах и для любых предметных областей. В них входят:

- CASE-системы, предназначенные для проектирования специфических реляционных схем DWH — «звезда» и «снежинка».

- Системы для управления метаданными.

- Системы для извлечения, очистки и загрузки данных.

- Системы для выполнения запросов, визуализации данных и генерации отчетов.



Спасибо за внимание!

