

ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

Основные задачи проектирования баз данных

- Обеспечение хранения в БД всей необходимой информации.
- Обеспечение возможности получения данных по всем необходимым запросам.
- Сокращение избыточности и дублирования данных.
- Обеспечение целостности базы данных.

ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДААННЫХ

Проектирование базы данных осуществляется в три этапа:

- Концептуальное (инфологическое) проектирование;
- Логическое (дatalogическое) проектирование;
- физическое проектирование.

КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Концептуальное (инфологическое) проектирование — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER-диаграммам.

Чаще всего концептуальная модель базы данных включает в себя:

- описание информационных объектов или понятий предметной области и связей между ними.
- описание ограничений целостности, т.е. требований к допустимым значениям данных и к связям между ними.

КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Цель инфологического моделирования (концептуального проектирования) - обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком.

Одной и наиболее популярных семантических моделей данных на этапе инфологического проектирования является «Сущность-Связь»(Entity-Relationship – ER - модель). Модель была предложена Ченом (Chen) в 1976 г. Моделирование предметной области базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов. В связи с наглядностью представления концептуальных схем баз данных ER – модели получили широкое распространение в CASE – системах (Computer Aided Software Engineering – программные средства, поддерживающие процессы автоматизированного проектирования баз данных, создания и сопровождения ПО и баз данных, генерацию кода, тестирование, документирование и управление проектом).

КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Цель этапа концептуального проектирования – создание концептуальной модели данных исходя из представлений пользователей о предметной области. Для ее достижения выполняется ряд последовательных процедур.

1. Определение сущностей и их документирование. Для идентификации сущностей определяются объекты, которые существуют независимо от других. Такие объекты являются сущностями. Каждой сущности присваивается осмысленное имя, понятное пользователям. Имена и описания сущностей заносятся в словарь данных. Если возможно, то устанавливается ожидаемое количество экземпляров каждой сущности.
2. Определение связей между сущностями и их документирование. Определяются только те связи между сущностями, которые необходимы для удовлетворения требований к проекту базы данных. Устанавливается тип каждой из них. Выявляется класс принадлежности сущностей. Связям присваиваются осмысленные имена, выраженные глаголами. Развернутое описание каждой связи с указанием ее типа и класса принадлежности сущностей, участвующих в связи, заносится в словарь данных.
3. Создание ER-модели предметной области. Для представления сущностей и связей между ними используются ER-диаграммы. На их основе создается единый наглядный образ моделируемой предметной области – ER-модель предметной области.

КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

4. Определение атрибутов и их документирование. Выявляются все атрибуты, описывающие сущности созданной ER-модели. Каждому атрибуту присваивается осмысленное имя, понятное пользователям. О каждом атрибуте в словарь данных помещаются следующие сведения:

- имя атрибута и его описание;
- тип и размерность значений;
- значение, принимаемое для атрибута по умолчанию (если такое имеется);
- может ли атрибут иметь Null-значения;
- является ли атрибут составным, и если это так, то из каких простых атрибутов он состоит. Например, атрибут "Ф.И.О. клиента" может состоять из простых атрибутов "Фамилия", "Имя", "Отчество", а может быть простым, содержащим единые значения, как-то "Сидорский Евгений Михайлович". Если пользователь не нуждается в доступе к отдельным элементам "Ф.И.О.", то атрибут представляется как простой;
- является ли атрибут расчетным, и если это так, то как вычисляются его значения.

5. Определение значений атрибутов и их документирование. Для каждого атрибута сущности, участвующей в ER-модели, определяется набор допустимых значений и ему присваивается имя. Например, атрибут "Тип счета" может иметь только значения "депозитный", "текущий", "до востребования", "карт-счет". Обновляются записи словаря данных, относящиеся к атрибутам, – в них заносятся имена наборов значений атрибутов.

6. Определение первичных ключей для сущностей и их документирование. На этом шаге руководствуются определением первичного ключа – как атрибута или набора атрибутов сущности, позволяющего уникальным образом идентифицировать ее экземпляры. Сведения о первичных ключах помещаются в словарь данных.

7. Обсуждение концептуальной модели данных с конечными пользователями. Концептуальная модель данных представляется ER-моделью с сопроводительной документацией, содержащей описание разработанной модели данных. Если будут обнаружены несоответствия предметной области, то в модель вносятся изменения до тех пор, пока пользователи не подтвердят, что предложенная им модель адекватно отображает их личные представления.

МОДЕЛЬ «СУЩНОСТЬ-СВЯЗЬ»

Модель «сущность-связь» (англ. “Entity-Relationship model”), или ER-модель, предложенная П. Ченом[1] в 1976 г., является наиболее известным представителем класса семантических (концептуальных, инфологических) моделей предметной области. ER-модель обычно представляется в графической форме, с использованием оригинальной нотации П. Чена, называемой ER-диаграмма, либо с использованием других графических нотаций (Crow's Foot, Information Engineering и др.).

Основные преимущества ER-моделей:

- наглядность;
- модели позволяют проектировать базы данных с большим количеством объектов и атрибутов;
- ER-модели реализованы во многих системах автоматизированного проектирования баз данных (например, ERWin).

Основные элементы ER-моделей:

- объекты (сущности);
- атрибуты объектов;
- связи между объектами.

Сущность — объект предметной области, имеющий атрибуты.

Связь между сущностями характеризуется:

- типом связи (1:1, 1:N, N:M);
- классом принадлежности. Класс может быть обязательным и необязательным. Если каждый экземпляр сущности участвует в связи, то класс принадлежности — обязательный, иначе — необязательный.

МОДЕЛЬ «СУЩНОСТЬ-СВЯЗЬ»



ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

- Логическое (дatalogическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных дatalogическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.
- Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.
- На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Цель этапа логического проектирования – преобразование концептуальной модели на основе выбранной модели данных в логическую модель, не зависимую от особенностей используемой в дальнейшем СУБД для физической реализации базы данных. Для ее достижения выполняются следующие процедуры.

1. Выбор модели данных. Чаще всего выбирается реляционная модель данных в связи с наглядностью табличного представления данных и удобства работы с ними.
2. Определение набора таблиц исходя из ER-модели и их документирование. Для каждой сущности ER-модели создается таблица. Имя сущности – имя таблицы. Осуществляется формирование структуры таблиц на основании изложенных в параграфе 1.4 правил. Устанавливаются связи между таблицами посредством механизма первичных и внешних ключей. Структуры таблиц и установленные связи между ними документируются.
3. Нормализация таблиц. Для правильного выполнения нормализации проектировщик должен глубоко изучить семантику и особенности использования данных. На этом шаге он проверяет корректность структуры таблиц, созданных на предыдущем шаге, посредством применения к ним процедуры нормализации. Эта процедура была описана в параграфе 1.5. Она заключается в приведении каждой из таблиц, по крайней мере, к 3НФ. В результате нормализации получается очень гибкий проект базы данных, позволяющий легко вносить в нее нужные расширения.

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

4. Проверка логической модели данных на предмет возможности выполнения всех транзакций, предусмотренных пользователями. Транзакция – это набор действий, выполняемых отдельным пользователем или прикладной программой с целью изменения содержимого базы данных. Так, примером транзакции в проекте БАНК может быть передача права распоряжаться счетами некоторого клиента другому клиенту. В этом случае в базу данных потребуется внести сразу несколько изменений. Если во время выполнения транзакции произойдет сбой в работе компьютера, то база данных окажется в противоречивом состоянии, так как некоторые изменения уже будут внесены, а остальные еще нет. Поэтому все частичные изменения должны быть отменены для возвращения базы данных в прежнее непротиворечивое состояние.

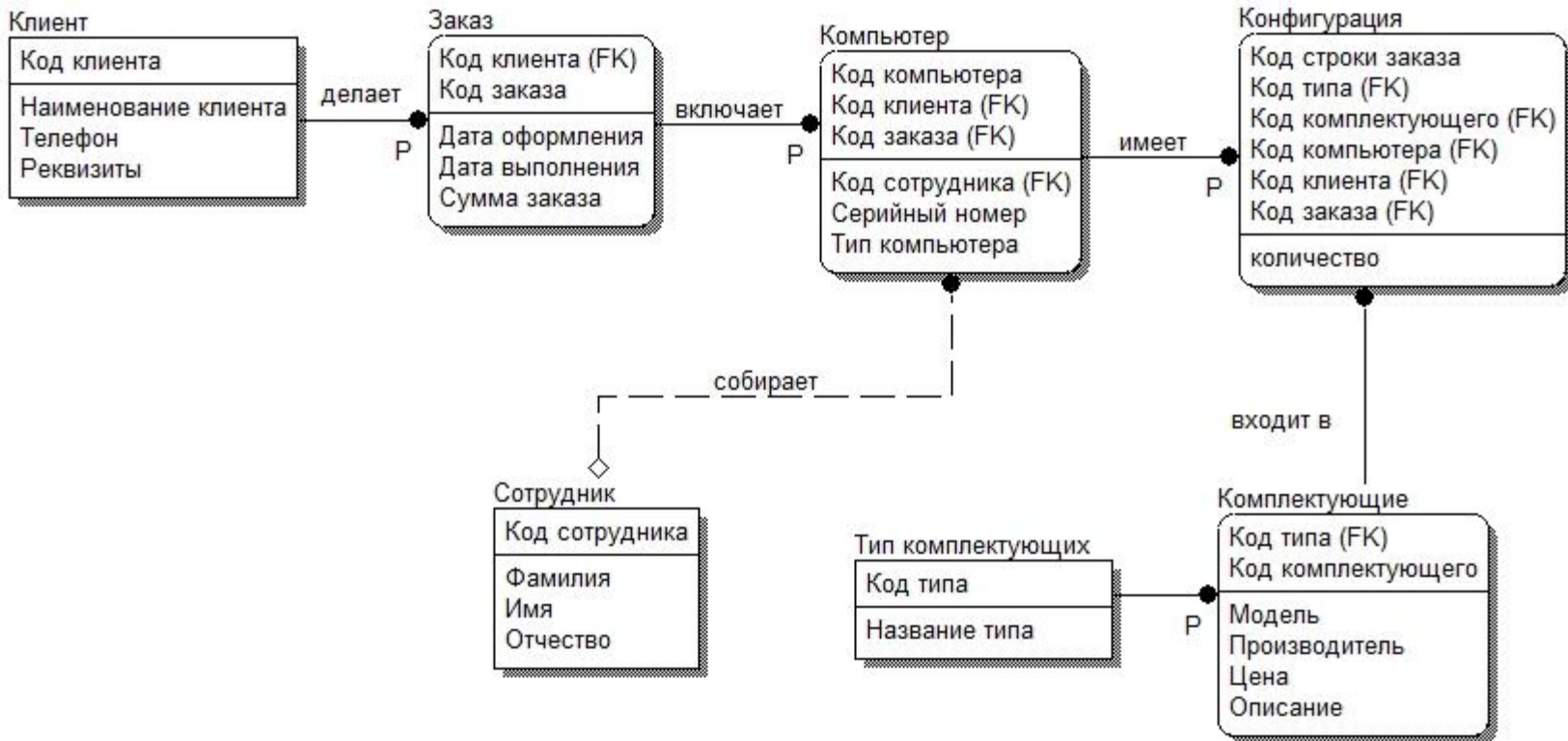
Перечень транзакций определяется действиями пользователей в предметной области. Используя ER-модель, словарь данных и установленные связи между первичными и внешними ключами, производится попытка выполнить все необходимые операции доступа к данным вручную. Если какую-либо операцию выполнить вручную не удастся, то составленная логическая модель данных является неадекватной и содержит ошибки, которые надо устранить. Возможно, они связаны с пропуском в модели сущности, связи или атрибута.

5. Определение требований поддержки целостности данных и их документирование. Эти требования представляют собой ограничения, которые вводятся с целью предотвратить помещение в базу данных противоречивых данных. На этом шаге вопросы целостности данных освещаются безотносительно к конкретным аспектам ее реализации. Должны быть рассмотрены следующие типы ограничений:

- обязательные данные. Выясняется, есть ли атрибуты, которые не могут иметь Null-значений;
- ограничения для значений атрибутов. Определяются допустимые значения для атрибутов;
- целостность сущностей. Она достигается, если первичный ключ сущности не содержит Null-значений;
- ссылочная целостность. Она понимается так, что значение внешнего ключа должно обязательно присутствовать в первичном ключе одной из строк таблицы для родительской сущности;
- ограничения, накладываемые бизнес-правилами. Например, в случае с проектом БАНК может быть принято правило, запрещающее клиенту распоряжаться, скажем, более чем тремя счетами.
- Сведения обо всех установленных ограничениях целостности данных помещаются в словарь данных.

6. Создание окончательного варианта логической модели данных и обсуждение его с пользователями. На этом шаге детализируется окончательный вариант ER-модели, представляющей логическую модель

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ



ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

- Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т.п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т.д.

ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Цель этапа физического проектирования – *описание конкретной реализации базы данных, размещаемой во внешней памяти компьютера*. Это описание структуры хранения данных и эффективных методов доступа к данным базы. При логическом проектировании отвечают на вопрос – *что надо сделать*, а при физическом – *выбирается способ, как это сделать*. Процедуры физического проектирования следующие.

1. Проектирование таблиц базы данных средствами выбранной СУБД.

Осуществляется выбор реляционной СУБД, которая будет использоваться для создания базы данных, размещаемой на машинных носителях. Глубоко изучаются ее функциональные возможности по проектированию таблиц. Затем выполняется проектирование таблиц и схемы их связи в среде СУБД. Подготовленный проект базы данных описывается в сопровождаемой документации.

2. Реализация бизнес-правил в среде выбранной СУБД.

Обновление информации в таблицах может быть ограничено бизнес-правилами. Способ их реализации зависит от выбранной СУБД. Одни системы для реализации требований предметной области предлагают больше возможностей, другие – меньше. В некоторых системах вообще отсутствует поддержка реализации бизнес-правил. В таком случае разрабатываются приложения для реализации их ограничений.

Все решения, принятые в связи с реализацией бизнес-правил предметной области, подробно описываются в сопроводительной документации. всесторонне взвешенными.

ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

3. Проектирование физической организации базы данных. На этом шаге выбирается наилучшая файловая организация для таблиц. Выявляются транзакции, которые будут выполняться в проектируемой базе данных, и выделяются наиболее важные из них. Анализируется *пропускная способность транзакций* – количество транзакций, которые могут быть обработаны за заданный интервал времени, и *время ответа* – промежуток времени, необходимый для выполнения одной транзакции. Стремятся к повышению пропускной способности транзакций и уменьшению времени ответа. На основании указанных показателей принимаются решения об оптимизации производительности базы данных путем определения индексов в таблицах, ускоряющих выборку данных из базы, или снижения требований к уровню нормализации таблиц. Проводится оценка дискового объема памяти, необходимого для размещения создаваемой базы данных. Стремятся к его минимизации.

Принятые решения по изложенным вопросам документируются.

4. Разработка стратегии защиты базы данных. База данных представляет собой ценный корпоративный ресурс, и организации ее защиты уделяется большое внимание. Для этого проектировщики должны иметь полное и ясное представление обо всех средствах защиты, предоставляемых выбранной СУБД.

5. Организация мониторинга функционирования базы данных и ее настройка. После создания физического проекта базы данных организуется непрерывное слежение за ее функционированием. Полученные сведения об уровне производительности базы данных используются для ее настройки. Для этого привлекаются и средства выбранной СУБД.

Решения о внесении любых изменений в функционирующую базу данных должны быть обдуманными и всесторонне взвешенными.

ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

