

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Тема_. ПРОЕКТИРОВАНИЕ БАЗ
ДАННЫХ

Требования к БД

Развитие технологий БД определяют следующие факторы:

- рост информационных потребностей пользователей;
- требования эффективного доступа к информации;
- новые виды машинной памяти и увеличение ее емкости;
- новые средства в сфере телекоммуникаций и др.

ОСНОВНЫЕ ТРЕБОВАНИЯ К БД:

1. высокое быстродействие (малое время отклика на запрос). **Время отклика** - промежуток времени от момента запроса к БД до фактического получения данных.

Существует термин **время доступа** - промежуток времени между выдачей команды записи (считывания) и фактическим получением данных.

Под **доступом** понимают операции поиска, чтения данных, операции записи, удаления и модификации данных часто называют **операциями обновления**.

Требования к БД

2. простота обновления данных.

Это требование и предыдущее противоречивы: повышение быстродействия требует упрощения структуры БД, что затрудняет процедуру обновления данных, увеличивает их избыточность.

3. независимость данных - возможность изменения логической и физической структуры БД без изменения представлений пользователей.

Независимость предполагает инвариантность к характеру хранения данных, ПО и техническим средствам. Она обеспечивает минимальные изменения структуры БД при изменениях стратегии доступа к данным и структуры самих исходных данных.

Требования к БД

Независимость данных достигается (см. далее) «смещением» всех изменений на этапы концептуального и логического проектирования с минимальными изменениями на этапе физического проектирования.

4. Совместное использование данных пользователями.
5. Безопасность данных - защита данных от нарушения секретности (преднамеренного или непреднамеренного), искажения или разрушения.

Безопасность данных включает их целостность и защиту.

Целостность данных - устойчивость хранимых данных к разрушению и уничтожению, связанных с неисправностями технических средств, системными ошибками и ошибочными действиями пользователей.

Требования к БД

Целостность данных предполагает:

- **отсутствие неточно введенных данных или двух одинаковых записей об одном и том же факте;**
- **защиту от ошибок при обновлении БД;**
- **невозможность удаления (или каскадное удаление) связанных данных разных таблиц;**
- **неискажение данных при работе в многопользовательском режиме и в распределенных БД;**
- **сохранность данных при сбоях (восстановление данных).**

Требования к БД

Для обеспечения целостности БД накладывают ограничения целостности в виде некоторых условий, которым должны удовлетворять хранимые в базе данные (например, диапазон значений атрибутов и др.):

целостности сущностей - любой кортеж (запись) любого отношения (таблицы) отличим от любого другого кортежа этого отношения;

целостности по ссылкам - для каждого значения внешнего ключа, появляющегося в ссылающемся отношении, в отношении, на которое ведется ссылка, должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным;

Целостность обеспечивается триггерами целостности - специальными приложениями-программами, работающими при определенных условиях.

Требования к БД

Защита данных от несанкционированного доступа предполагает ограничение доступа к конфиденциальным данным и может достигаться:

- 1. введением системы паролей;**
- 2. получением разрешений от администратора БД (АБД);**
- 3. запретом от администратора БД на доступ к данным;**
- 4. формированием видов - таблиц, производных от исходных и предназначенных конкретным пользователям.**

Процедуры 2-4 легко выполняются в рамках языка SQL .

Требования к БД

6. Стандартизация построения и эксплуатации БД (СУБД).

Стандартизация обеспечивает преемственность поколений СУБД, упрощает взаимодействие БД одного поколения СУБД с одинаковыми и различными моделями данных. Стандартизация (ANSI/SPARC) осуществлена в значительной степени в части интерфейса пользователя СУБД и языка SQL. Это позволило успешно решить задачу взаимодействия различных реляционных СУБД как с помощью языка SQL, так и с применением приложения Open DataBase Connection (ODBC).

- 7. Минимальная избыточность – любой элемент данных должен храниться в единственном экземпляре.
- 8. Многократное использование данных.
- 9. Однократный ввод данных.
- 10. Адекватность отображения данных соответствующей предметной области.
- 11. Дружелюбный интерфейс пользователя.

Основные этапы проектирования БД

Проектирование – процесс создания описаний новой системы, которая способна функционировать.

В процессе проектирования БД выделяют три этапа:

- **концептуальное проектирование** – создается концептуальная модель БД
- **логическое проектирование** – создается логическая модель БД для выбранной СУБД
- **физическое проектирование** – создаются файлы БД на машинном носителе.

Концептуальное проектирование

Процесс создания информационной (концептуальной) модели начинается с определения концептуальных требований ряда пользователей.

Концептуальные требования могут определяться и для некоторых задач (приложений), которые в ближайшее время реализовывать не планируется. Это может повысить трудоемкость работы, однако поможет наиболее полно учесть все нюансы функциональности, требуемой для разрабатываемой системы, и снизит вероятность переделки в дальнейшем. Требования отдельных пользователей должны быть представлены в едином «обобщенном представлении». Последнее называют **концептуальной моделью**.

Концептуальная модель...

...представляет объекты и их взаимосвязи без указания способов их физического хранения

Объект – это абстракция множества предметов реального мира, обладающих одинаковыми характеристиками и законами поведения.

Класс – это множество предметов реального мира, связанных общностью структуры и поведением. Объекты объединяются в классы по общим характеристикам.

Концептуальная модель – это модель предметной области. При проектировании концептуальной модели все усилия должны быть направлены на структуризацию данных и выявление взаимосвязей между ними без рассмотрения особенностей реализации и вопросов эффективности обработки.

Концептуальная модель

Проектирование концептуальной модели основано на анализе решаемых предприятием задач по обработке данных.

Концептуальная модель включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области и выявляемых в результате анализа данных. Имеются в виду данные, используемые как в уже разработанных прикладных программах, так и в тех, которые только будут реализованы.

Пример

На предприятии выделены первоначальные объекты:

Заявки- поступающие от магазинов на определённый период.

Договора- заключаются с поставщиками на вид товара.

Поставщики- организации или физические лица, с которыми заключаются договора на поставку товара.

Заказчики- магазины, предприятия и организации, подающие заказ на приобретение того или иного товара.

Счета- ведутся на этапе заключения договоров с поставщиками и заказчиками.

Накладные- создаются на основании получения заказа от заказчика, для отгрузки.

Справки- получение/выдача различных справок как заказчику, так и поставщику.

Товар- присутствует на основании заявки и договора с поставщиком.

Пример

Определяются взаимосвязи объектов. Взаимосвязь выражает отображение или связь между двумя множествами данных.

Различают взаимосвязи типа «**один к одному**», «**один ко многим**» и «**многие ко многим**».

Например, **заказчик** производит заказ на покупку товара впервые, осуществляется первичная регистрация его данных и сведений о сделанном заказе. Если **заказчик** производит заказ повторно, осуществляется регистрация только данного заказа.

Вне зависимости от числа заказов, заказчик имеет уникальный идентификационный номер (уникальный ключ заказа).

Информация о каждом заказчике включает: **наименование заказчика, адрес, телефон, факс, фамилию, имя, отчество, признак юридического лица и примечание.**

Свойствами объекта **Заказчик** являются «**уникальный ключ заказчика**», «**наименование заказчика**».

Пример

Следующий объект - **Товар**. Этот объект имеет свойства «уникальный ключ товара», «наименование товара».

Третий объект — **Поставщик**. Его свойствами являются «уникальный ключ поставщика», «наименование поставщика».

Допустим, в определенный момент времени один **заказчик** может сделать только один заказ. В этом случае между объектами **Заказчик** и **Товар** устанавливается взаимосвязь «**один к одному**» (между двумя типами объектов).

В определенный момент времени один **заказчик** может стать обладателем нескольких товаров, при этом несколько заказчиков не могут являться обладателями одного товара. Тогда между объектами **Заказчик** и **Товар** устанавливается взаимосвязь «**один ко многим**» (между двумя типами объектов).

Пример

Взаимосвязь «**один ко многим**» можно обозначить с помощью одинарной стрелки в направлении к «одному» и **двойной стрелки** в направлении ко «**многим**». В этом случае одной записи данных первого объекта (его часто называют **родительским** или **основным**) будет соответствовать несколько записей второго объекта (**дочернего** или **подчиненного**).

Взаимосвязь «**один ко многим**» очень распространена при разработке реляционных баз данных. В приведенном примере в качестве основного можно представить объект **Заказчик**, в котором хранятся сведения обо всех заказчиках. При обращении к записи для определенного заказчика нам доступен список всех покупок, которые он сделал, и сведения о которых хранятся в объекте **Товар**.

Пример. Определение объектов

Выделим следующие объекты:

- | | | |
|-----------------|--------------------|---------------------|
| 1. ТОВАР - (Т); | 2. ЗАКАЗЧИК - (З); | 3. ПОСТАВЩИК - (П); |
| 4. СЧЕТА - (С); | 5. ДОГОВОР - (Д); | 6. НАКЛАДНЫЕ - (Н). |

Для объектов определим свойства, которые будем хранить в БД. Учитывать, что при переходе от логической к физической модели данных может произойти усечение числа объектов.

Как правило, значительное число необходимых данных, может быть подсчитано при выводе информации. В связи с изменением алгоритмов расчета или исходных величин, некоторые расчетные показатели приходится записывать в БД, чтобы гарантированно обеспечить фиксацию их значений.

Выбор показателей, которые обязательно следует хранить в БД, достаточно сложен. Требуется изучение работы предприятия и анализа концептуальной модели.

Концептуальная модель БД представляет собой совокупность объектов (с указанием их характеристик), которые должны быть помещены в БД, и связей между ними.

Пример.

Задание первичных и альтернативных ключей, определение свойств объектов

Концептуальная модель переносится затем в модель данных, совместимую с выбранной СУБД. Возможно, что отраженные в концептуальной модели взаимосвязи между объектами окажутся впоследствии нереализуемыми средствами выбранной СУБД. Это потребует изменения концептуальной модели.

Версия концептуальной модели, которая может быть обеспечена конкретной СУБД, называется **логической моделью**.

Логическая модель данных ...

...отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения. **Логическая** модель может быть реляционной, иерархической или сетевой.

В реляционной модели все объекты, составляющие концептуальную модель, помещаются в одну или несколько таблиц (между которыми должны быть установлены связи).

Размещение данных в таблицы производится так, чтобы каждая таблица была нормализована, т.е. соответствовала по крайней мере третьей нормальной форме.

Нормализация - процесс реорганизации данных путем ликвидации повторяющихся групп и иных противоречий в хранении данных с целью приведения таблиц к виду, позволяющему осуществлять непротиворечивое и корректное редактирование данных.

Теория нормализации основана на концепции нормальных форм. Приведение модели к требуемому уровню нормальной формы является основой построения реляционной БД

Нормализация

В процессе нормализации элементы данных группируются в таблицы, представляющие объекты и их взаимосвязи.

Теория нормализации основана на том, что определенный набор таблиц обладает лучшими свойствами при включении, модификации и удалении данных, чем все остальные наборы таблиц, с помощью которых можно представить те же данные.

Введение нормализации отношений при разработке информационной модели обеспечивает минимальный объем физической, то есть записанной на каком-либо носителе БД и ее максимальное быстродействие, что напрямую отражается на качестве функционирования информационной системы. Нормализация информационной модели выполняется в несколько этапов.

Нормализация(1НФ)

Данные, представленные в виде двумерной таблицы, являются первой нормальной формой реляционной модели.

Первый этап нормализации - образование двумерной таблицы, содержащей все необходимые свойства (поля) информационной модели, и в выделении ключевых свойств (полей).

Таблица находится в 1НФ, если значения всех полей **атомарны**, а записи – **уникальны**. Полученная внушительная таблица будет содержать разнородную информацию. Будут наблюдаться аномалии включения, обновления и удаления данных, так как при выполнении этих действий нам придется уделить внимание данным (вводить или заботиться о том, чтобы они не были стерты), которые не имеют к текущим действиям никакого отношения. ...

Нормализация(2НФ)

Отношение задано во 2НФ, если оно является отношением в 1НФ и каждое свойство, не являющееся первичным свойством в этом отношении, полностью зависит от любого возможного ключа этого отношения.

Если все возможные ключи отношения содержат по одному свойству, то это отношение задано во 2НФ, так как в этом случае все свойства, не являющиеся первичными, полностью зависят от возможных ключей.

Если ключи состоят более чем из одного свойства, отношение, заданное в 1НФ, может не быть отношением во 2НФ. Приведение отношений ко 2НФ заключается в обеспечении полной функциональной зависимости всех свойств от ключа за счет разбиения таблицы на несколько, в которых все имеющиеся свойства будут иметь полную функциональную зависимость от ключа этой таблицы.

В процессе приведения модели ко 2НФ в основном исключаются аномалии дублирования данных.

Нормализация(3НФ)

Отношение задано в 3НФ, если оно задано во 2НФ и каждое свойство этого отношения, не являющееся первичным, не транзитивно зависит от каждого возможного ключа этого отношения.

Транзитивная зависимость выявляет дублирование данных в одном отношении. Если А, В и С - три свойства одного отношения и С зависит от В, а В от А, то говорят, что С транзитивно зависит от А. Преобразование в 3НФ происходит за счет разделения исходного отношения на два.

Пользователям выделяются подмножества логической модели, называемые внешними моделями, отражающие их представления о предметной области.

Нормализация(3НФ)

Внешняя модель соответствует представлениям, которые пользователи получают на основе логической модели, в то время как концептуальные требования отражают представления, которые пользователи первоначально желали иметь и которые легли в основу разработки концептуальной модели.

Логическая модель представляет собой совокупность нормализованных таблиц, содержащих данные, со связями между ними.

Физическая модель

На этапе физического проектирования логическая модель отображается в физическую память (жесткий магнитный диск, оптический диск, магнитная лента или какой-либо другой носитель информации).

На этом этапе решают, на каком устройстве будет храниться БД, как она будет храниться (в виде одного или нескольких файлов), выбираются необходимые средства резервирования и восстановления базы данных.

Физическая модель представляет собой размещенную в памяти компьютера БД.

Физическая модель, определяющая размещение данных, методы доступа и технику индексирования, называется **внутренней моделью системы**.

Внешние модели ...

... никак не связаны с типом физической памяти, где будут храниться данные, и с методами доступа к этим данным.

Это - первый уровень независимости данных.

Если концептуальная модель способна учитывать расширение требований к системе в будущем, то вносимые в нее изменения не должны оказывать влияния на существующие внешние модели. Это — второй уровень независимости данных.

Построение логической модели обусловлено требованиями используемой СУБД. Поэтому при замене СУБД она также может измениться.

Внешние модели ...

Все актуальные требования предметной области и адекватные им «скрытые» требования на стадии проектирования должны найти свое отражение в концептуальной модели.

Конечно, нельзя предусмотреть все возможные варианты использования и изменения БД. Но в большинстве предметных областей такие основные данные, как объекты и их взаимосвязи, относительно стабильны. Меняются только информационные требования, то есть способы использования данных для получения информации.

Степень независимости данных определяется тщательностью проектирования БД. Всесторонний анализ объектов предметной области и их взаимосвязей минимизирует влияние изменения требований к данным в одной программе на другие программы. В этом и состоит всеобъемлющая независимость данных.

Внешние модели ...

Основное различие между указанными выше тремя типами моделей данных (концептуальной, логической и физической) состоит в способах представления взаимосвязей между объектами.

При проектировании БД требуется различать взаимосвязи между объектами, между свойствами одного объекта и между свойствами различных объектов.

В процессе проектирования объекты преобразуются в отношения, свойства в поля таблиц, методы – в процедуры, формы и т.д. (что и было произведено). Правильно проведенный объектно-ориентированный анализ позволяет значительно облегчить работу.