

Администрирование баз данных

Администрирование БД заключается в предоставлении пользователям соответствующих прав использования возможностей работы с базой данных; обеспечении целостности данных, а также создании многопользовательских приложений.

Администратор базы данных может использовать табличные пространства для:

- управления распределением памяти для объектов базы данных;
- установления квот памяти для пользователей базы данных;
- управления доступностью данных, включая режимы (состояния) online или offline;
- копирования и восстановления данных;
- распределения данных по устройствам для повышения производительности.

Администрирование базой данных – это её инсталляция (установка), управление доступом к ней, обеспечение целостности базы данных и др.

Формирование базы данных на компьютере заключается в подготовке одного или нескольких файлов данных, которые будут установлены и использованы в виде единой база данных. **База данных** создается один раз, независимо от того, сколько файлов данных она имеет, и сколько обращений к ней будет производиться. Процедуру создания базы данных можно использовать, если стереть информацию в существующей базе данных. При этом будет создана новая база данных с тем же именем и той же физической структурой.

Создание базы данных включает следующие операции:

- создание новых файлов данных, или стирание данных, хранившихся в предыдущих файлах данных;
- создание структур, требующихся для доступа и работы с базой данных;
- инициализацию управляющих файлов и журнала для базы данных.

Создаваемый **журнал** обычно содержит подробную информацию о загрузке, включая:

- имена входящих в базу данных файлов;
- входные данные и связанные с ними определения таблиц;
- ошибки и результаты работы БД;
- итоговую статистику.

Прежде чем **создавать базу данных** необходимо:

- спланировать таблицы и индексы, а также оценить необходимое им пространство;
- спланировать защиту базы данных, включая конфигурацию её журналов с учётом занимаемого ими пространства и стратегию резервного копирования.

Администратор базы данных также осуществляет запуск и останов базы данных.

Выделяют три основные категории пользователей баз данных:

- разработчики,
- администраторы баз данных,
- пользователи.

Обязанности разработчика

В обязанности разработчика входит:

- проектирование и разработка базы данных;
- проектирование структуры базы данных в соответствии с установленными требованиями;
- оценка требований необходимой памяти;
- формулирование модификаций структуры базы данных;
- передача этой информации администратору базы данных (инструкция администратору), а также создание иных инструктивных документов (инструкции оператору, пользователю и т.п.);
- установка мер по защите базы данных в процессе её разработки и эксплуатации и др.

Обязанности администратора БД

Администрирование базами данных осуществляется лицом, управляющим этой системой. Такое лицо называется **администратором** базы данных.

Если база данных большая, эти обязанности могут выполнять несколько человек (группа) – администраторов БД.

В **обязанности администратора** могут входить:

- инсталляция БД и обновление версий СУБД;
- распределение дисковой памяти и планирование требований системы к памяти;
- создание структур памяти в базе данных (табличных пространств);
- создание первичных объектов (таблиц, индексов);
- модификация структуры базы данных;
- отслеживание и оптимизация производительности базы данных;
- соблюдение лицензионных соглашений;
- зачисление пользователей и поддержание защиты системы;

- управление и отслеживание доступа пользователей к базе данных;
- планирование резервного копирования и восстановления;
- поддержание архивных данных на устройствах хранения информации;
- осуществление резервного копирования и восстановления.

В некоторых случаях база данных должна также иметь одного или нескольких **сотрудников службы безопасности**, которые, главным образом, отвечают за регистрацию новых пользователей, управление и отслеживание доступа пользователей к базе данных, и защиту базы данных.

Пользователи базы данных и схемы

Каждая база данных включает список имен пользователей. Чтобы получить доступ к базе данных, пользователь должен соединиться с базой данных, предоставив своё имя, которое включает собственно **имя (логин) и пароль**. Такая структура предназначена для предотвращения несанкционированного доступа к БД.

При первоначальном создании (включении) пользователя в БД ему можно определить некоторое табличное пространство (например, заданное по умолчанию), в котором будут создаваться объекты пользователя. Пользователю может быть определён объём памяти, который он может использовать в табличном пространстве.

Другой способ создания пользователя БД подразумевает предоставление пользователю определённой роли.

Для удаления (исключения) пользователя из базы данных удаляют его учётную запись из словаря базы данных. Если пользователь владел некоторыми объектами базы данных, то их можно последовательно удалить, или автоматически уничтожить все объекты, связанные с учётной записью пользователя.

Обеспечение целостности базы данных

Важнейшим вопросом, возникающим при проектировании и эксплуатации систем обработки данных, является **обеспечение целостности данных**. Нарушение целостности может быть связано не только с ошибками оператора при вводе данных, но и со сбоями в работе системы. Отказ в работе системы может возникнуть, когда одни операции были выполнены, а другие, связанные с ними определенными ограничениями операции, не успели выполняться. Например, удалена запись о сотруднике, а не удалены относящиеся к ней записи о его детях и т. п. Нарушение целостности возникает при вводе или корректировке данных, а также при выполнении реляционных операций (проекция, соединение).

Решение проблемы целостности заключается в обеспечении в любой момент времени правильности данных в базе данных, для этого используют правила проверки достоверности данных гарантирующие, что недействительные данные не попадут в таблицы БД. **Целостность данных** обеспечивается набором специальных предложений, называемых **ограничениями целостности**.

Ограничение целостности

Ограничения – это правила, применимые во время или после создания таблиц, распространяемые на то, как эти таблицы могут заполняться.

Ограничения целостности представляют собой утверждения о допустимых значениях отдельных информационных единиц и связях между ними.

Ограничение целостности устанавливает правила на уровне БД, определяя набор проверок для таблиц системы. Эти проверки автоматически выполняются всякий раз, когда вызывается оператор вставки, модификации или удаления данных в таблице. Если какие-либо ограничения нарушены, операторы отменяются.

Ограничения целостности определяются в большинстве случаев особенностями предметной области, хотя могут отражать и чисто информационные характеристики. Например, если используются цифровые коды для кодирования какой-либо номенклатуры, то ограничения на тип используемых символов для соответствующего атрибута в БД определяются не спецификой предметной области, а выражается в том, что возраст работающего должен быть не менее 16 лет (специфика предметной области).

Ограничения целостности могут относиться к разным информационным объектам атрибутам (полям), кортежам (строкам, записям), отношениям (таблицам, файлам), связям между файлами и т. п. Для простоты изложения будем далее пользоваться в основном терминологией реляционных СУБД. Однако все сказанное ниже относится и к СУБД других классов.

Для полей чаще всего используются следующие **виды ограничений**.

1. Тип и формат поля.
2. Задание диапазона значений. Обычно используется для числовых полей. Различают открытые и закрытые диапазоны. Первые фиксируют значение только одной из границ (верхней или нижней), а вторые – обеих границ. Не только значения диапазона, но и его тип зависят от особенностей предметной области. Так, например, существовал диапазон заработной платы, ограничивающий нижний и верхний её пределы. Ныне снято ограничение по верхнему пределу.
3. Признак непустого поля. Характеризует недопустимость пустого значения поля в БД. Например, поля со сведениями о сотрудниках («фамилия», «имя», «отчество», «оклад») обязательно должны иметь какое-то значение, а в поле «ученая степень» может отсутствовать значение.

Специфическим ограничением на значение поля является признак его уникальности. Это ограничение проверяет допустимость значения данного поля, но при этом просматривается вся таблица (файл). Признак уникальности поля тесно связан с понятием ключа, но уже последнего, так как ключ может быть составным. Уникальное поле является вероятным ключом данного отношения. При наличии нескольких вероятных ключей один из них должен быть выбран в качестве первичного ключа. Это поле не должно иметь пустое значение. Не все СУБД поддерживают концепцию ключа и требуют определять его при описании базы данных. Рассматриваемое ограничение обычно возникает при отображении объектов в базе. Уникальное поле при этом является идентификатором данного объекта, поэтому оно часто называется ограничением целостности объекта.

Рассмотренные выше ограничения определяли проверки значения поля вне зависимости от того, вводится ли значение впервые или корректируются имеющиеся в базе данных значения.

Ограничения, используемые только при проверке допустимости корректировки, называют ограничениями перехода. Например, в базе данных имеется поле «возраст сотрудника». При корректировке значение этого поля может только увеличиваться. Если в БД хранится поле «год рождения», то на корректировку этого поля вообще должен быть наложен запрет. Другой пример ограничения перехода – корректировка поля «семейное положение». В нём значение «вдовец» может быть исправлено только на значение «женат», а значение «холост» не должно быть исправлено на значение «разведен». При попытке произвести некорректное исправление должно выдаваться диагностирующее сообщение.

Когда речь идёт об ограничениях целостности, относящихся к кортежу, то имеется в виду ограничение на значение всей строки, рассматриваемой как единое целое, или ограничения на соотношения значений отдельных полей в пределах одной строки. Естественным ограничением является требование уникальности каждой строки таблицы.

По определению в реляционном отношении не может быть одинаковых кортежей. Однако не все реляционные СУБД обеспечивают автоматическое соблюдение этого ограничения. Пример ограничения на соотношения полей внутри одного кортежа: значение поля «стаж» не должно превышать «возраст» – 16.

Пример ограничений, относящихся ко всей таблице. Допустим, что фонд заработной платы формируется исходя из величины средней заработной платы одного сотрудника, равной 5000 руб. В качестве ограничения целостности таблицы можно задать выражение, где среднее значение поля «оклад» не должно быть больше 5000.

Приведём **примеры ограничения целостности, которыми проверяют соотношения между записями одного файла:**

- 1) нельзя быть родителем и ребенком одного и того же человека;
- 2) год рождения родителя должен быть меньше, чем год рождения ребенка.

Первый из приведенных примеров является частным случаем более общего ограничения на отсутствие циклов. К аналогичным ограничениям относятся ограничения на наличие циклов при определении состава изделия (узел не может входить сам в себя), при описании организационной структуры и в других случаях.

Если СУБД не позволяет контролировать подобные ограничения целостности, то надо написать **универсальную программу** (создать процедуру), позволяющую это делать. Такого рода проверки нужны достаточно часто. Рассмотренные ограничения затрагивали информационные единицы в пределах одной таблицы. Ограничения могут затрагивать несколько взаимосвязанных таблиц. Например, ограничения целостности связи.

Такое ограничение выражается в том, что значение атрибута, отражающего связи между объектами и являющегося внешним ключом отношения, обязательно должно совпадать с одним из значений атрибута, являющегося ключом отношения, описывающего соответствующий объект. Например, если в базе данных существует файл, отражающий связь между преподавателями и дисциплинами, которые каждый из них может преподавать, то код преподавателя в этом файле должен соответствовать одному из кодов в файле «Преподаватели», а код дисциплины – значению соответствующего поля в файле «Дисциплины».

Видом ограничения является **запрет на обновление**. Он может относиться как к отдельному полю, так и к записи или целой таблице. В некоторых СУБД существует **запрет на корректировку ключевого поля**.

Запрет на обновление может быть связан с технологией обработки данных или со спецификой предметной области. Например, если описывается объект «Личность», то его атрибуты дата и место рождения являются постоянными и не могут меняться. Для соответствующих полей в БД рекомендуется задавать запрет на обновление. При этом запрет на обновление может относиться как к отдельному полю, так и ко всей записи и или файлу.

По моменту контроля за соблюдением ограничения целостности различают **одномоментные** и **отложенные ограничения целостности**. **Отложенные ограничения целостности** могут не соблюдаться в процессе выполнения какой-то группы операций, но обязаны быть соблюдены по завершению выполнения этой группы операций. С понятием отложенного ограничения целостности тесно связано **понятие транзакции**. Примером отложенных ограничений целостности могут служить действия при выполнении бухгалтерских проводок.

Реализация контроля за соблюдением ограничений целостности может потребовать больших затрат. Целесообразно задавать условия, приводящие к нарушению целостности, а данные проверять только в случаях, приводящих к нарушению ограничения. Например, имеется ограничение: заработная плата руководителя не может быть ниже зарплаты подчиненных. В этом случае увеличение заработной платы руководителя не может нарушить целостность и проверку БД проводить не требуется.

Ограничения целостности, присущие той или иной предметной области, должны быть выявлены при обследовании и зафиксированы в инфологической модели. Для того чтобы обеспечить целостность при выполнении реляционных операций, необходимо, чтобы соблюдались определенные правила при их выполнении, а также чтобы файлы были правильно спроектированы.

Большинство СУБД не контролирует семантическую целостность при выполнении реляционных операций, и, как следствие, такой контроль возлагается на администратора БД или конечного пользователя.

При выполнении арифметических операций нарушение целостности может возникнуть при «рассогласовании» единиц измерения.

Для обеспечения защиты базы данных, используют различные ограничения целостности данных, журнализацию, репликацию и резервное копирование.

Резервное копирование базы данных

С целью обеспечения достоверности и постоянной работоспособности БД периодически вручную или автоматически осуществляется копирование базы данных. Основными причинами, побуждающими выполнение процедур копирования, являются различные структурные изменения базы данных:

- создание или удаление табличного пространства;
- добавление или переименование (перемещение) файла данных в существующем табличном пространстве;
- добавление, переименование (перемещение) или удаление журнала повторения и др.

При этом резервное копирование может осуществляться непосредственно перед изменениями и после них.

Выделяют два основных вида резервного копирования:

1. **Непротиворечивое (холодное) резервное копирование**, когда копии создаются, в случае закрытой для пользователей БД (close). Копия базы данных, созданной в автономном режиме, содержит: все файлы данных, журналы повторов и управляющие файлы. После остановки БД, все файлы базы копируются на один из backup дисков. По окончании копирования осуществляется перезагрузка базы данных.
2. **Резервное (горячее) копирование в оперативном режиме**, к примеру, когда БД всё время находится в оперативном режиме и доступна пользователям.

Первоначальный **режим архивирования** базы данных устанавливаются во время её создания. После создания базы данных решают, необходимо ли изменить первоначальный режим архивирования.

Автоматическое архивирование журнала в любой момент можно выключить. В этом случае придётся вручную периодически архивировать заполняемые группы журнала.