Тема 26. Защита баз данных

Под защитой баз данных понимается сохранение целостности и обеспечение секретности данных.

Сохранение целостности – непрерывное выполнение всех ограничений целостности, включающих как функциональные зависимости, так и семантические ограничения.

Обеспечение секретности означает ограничение прав доступа пользователей, которым разрешается работать лишь с частью БД, выполнять ограниченный круг операций и т.д.

Целостность

Целостность в реляционной модели имеет четыре аспекта:

- Структурная целостность,
- Языковая целостность,
- Ссылочная целостность,
- Семантическая целостность.

Структурная целостность – требование к СУБД работать только с однородными структурами данных типа «реляционное» отношение. Реляционное отношение должно удовлетворять всем накладываемым на него в реляционной модели требованиям: отсутствие дубликатов, обязательность ключа, неупорядоченность кортежей и т.д. Сюда же относятся и проблемы неопределенных (NULL) значений.

Языковая целостность – требование к языкам описания и манипулирования данными, чтобы они были не ниже стандарта SQL. Низкоуровневые средства манипулирования данными не должны быть доступны.

Ссылочная целостность означает непрерывное выполнение одного из принципов взаимосвязи между кортежами связанных отношений:

- Кортежи подчиненного отношения удаляются при удалении связанного с ними кортежа главного отношения;

- Кортежи подчиненного отношения модифицируются при удалении связанного с ними кортежа главного отношения, при этом внешний ключ получает значение NULL.

Ссылочная целостность обеспечивает поддержку непротиворечивого состояния БД в процессе модификации данных.

Семантическая целостность понимается как ограничения на содержание баз данных.

Целостность может быть обеспечена двумя путями: декларативным и процедурным.

Декларативные ограничения (бизнес-правила) включают:

- Ограничения на значения атрибута (значение по умолчанию, неопределенное значение, произвольное условие);
- Ограничения доменов (тип данных, бизнес-правила и т.д.);
- Ограничения отношений (ограничения значений комбинации атрибутов и др.);
- Ограничения связи отношений (обязательность связи, каскадное удаление и изменение данных, ограничения типа связи).

Процедурное поддержание целостности обеспечивается использованием хранимых процедур и триггеров.

Задание ограничений целостности в SQL

Оператор CREATE TABLE

CREATE TABLE имя_таблицы ({ описание_элемента_табл | ограничение_табл},..)

Описание_элемента_таблицы:

Имя_столбца тип_данных [DEFAULT {literal | USER | NULL}]

[NOT NULL [UNIQUE] | FOREIGN KEY REFERENCES имя_табл

(имя_первичного_ключа_табл) | CHECK (условие_допустимости)]…

Определение столбца содержит имя столбца, его тип и два необязательных раздела: значение по умолчанию и дополнительные ограничения.

Значения по умолчанию (DEFAULT): literal – константа, USER – имя пользователя, NULL – неопределенное значение.

Дополнительные ограничения:

NOT NULL – запрет неопределенных значений;

UNIQUE – требование уникальности ключа;

FOREIGN KEY – значение столбца берется из главной связанной таблицы (внешний ключ, связь типа M:1);

CHECK – условие допустимости значения.

Пример.

CREATE TABLE Заказы

(Фамилия CHAR (20) NOT NULL FOREIGN KEY REFERENCES Покупатели (Фамилия),

Товар CHAR (20) NOT NULL CHECK (Товар<>’СПИЧКИ’),

Количество NUMERIC NOT NULL,

PRIMERY KEY (Фамилия, Товар))

Ограничения можно именовать, используя конструкцию CONSTRAINT имя_ограничения, где имя ограничения записывается в форме тип_имя-отношения. Тип ограничения: PK – первичный ключ, FK внешний ключ, CK - CHECK, U – UNIQUE, DF – DEFAULT.

Пример.

CONSTRAINT PK_Заказы PRIMARY KEY (Фамилия, Товар).

Оператор ALTER TABLE

ALTER TABLE имя_таблицы

{ADD определение_столбца |

ALTER имя _столбца {SET DEFAULT значение | DROP DEFAULT} |

DROP имя_столбца {CASCADE | RESTRICT} |

ADD{опред_перв_ключа | опр_внешн_ключа | условие_уникальности | условие_проверки }|

DROP CONSTRAINT имя_условия {CASCADE | RESTRICT}}

Одним оператором ALTER TABLE можно провести только одно изменение: добавить новый столбец, изменить умолчание, удалить столбец, добавить или удалить первичный ключ, внешний ключ, условие уникальности, условие проверки.

Ограничения целостности в QBE

Система QBE позволяет задавать ключевые атрибуты отношений. При этом поддерживается функциональная зависимость всякого неключевого поля от множества совместно взятых ключевых полей. Такая проверка целостности производится при каждом включении или модификации кортежа отношения. Операции, нарушающие зависимости, не выполняются.

В QBE для каждого отношения поддерживается таблица ограничений, которая записывается в виде специальных строк таблицы отношения.

Пример.

Покупатели	Фамилия	Адрес	Баланс
CONSTR(I.,U.)			>= -50

Конструкция CONSTR указывает, что строка является декларацией ограничений. Ограничению подвергаются операции INSERT и UPDATE: новые и модифицируемые записи должны иметь баланс >= -50.

Пример.

Заказы	Фамилия	Товар	Количество
CONSTR(I.)		-товар	

Поставщики	Название	Адрес	Товар	Цена
			_товар	

Ограничение запрещает включать товары, для которых нет поставщиков.

Пример.

Покупатели	Фамилия	Адрес	Баланс
COND	Петров		
CONSTR(COND)			>= -50

Проверка ограничения производится только для Петрова.

Обеспечение секретности

Наиболее важными аспектами секретности являются защита от нежелательной модификации или разрушения БД и защита от несанкционированного чтения. Для обеспечения секретности БД и других систем используются следующие общие механизмы:

1. Идентификация пользователя. Различным пользователям предоставляются разные права по отношению к базам данных или их частям. Поэтому необходимо уметь идентифицировать пользователей. Идентификация обычно производится с помощью паролей, известных только системе и конкретному лицу. Пароли так же нуждаются в защите, как и данные.
2. Физическая защита.
3. Поддержка и передача прав. Система должна поддерживать перечень прав, предоставленных каждому пользователю для каждой защищенной части БД. Одним из таких прав может быть право передачи своих прав другому лицу.

В системах баз данных используются два специальных механизма защиты: представления (подсхемы) и языки запросов, как средства определения прав.

Подсхема позволяет переопределять концептуальную БД, поддерживает логическую независимость данных, служит удобным средством защиты. Доступ пользователя к БД возможен только в пределах подсхемы. Это обеспечивает автоматическую защиту частей БД, не попавших в подсхему. Кроме этого, имеется возможность ограничения прав доступа в рамках

подсхемы: разные категории пользователей могут иметь разные права на выполнение операций чтения, добавления, удаления и модификации.

Языки запросов могут быть использованы для определения привилегий доступа почти так же, как и при определении ограничений целостности.

Секретность в SQL

Стандарт SQL определяет два оператора GRANT и REVOKE для предоставления и отмены привилегий.

Предоставление привилегий

GRANT {список_действий | ALL PRIVILEGES}

ON имя_объекта TO { имя_пользователя | PUBLIC}

[WITH GRANT OPTION],

где список_действий – действия из набора: SELECT, INSERT, DELETE, UPDATE (для таблиц);

ALL PRIVILEGES – все действия;

имя_объекта – имя таблицы, представления, хранимой процедуры, триггера;

PUBLIC – права предоставляются всем пользователям;

WITH GRANT OPTION - передаются не только права на действия, но и право на передачу прав другим пользователям.

Пример.

С базой данных работают три пользователя с именами User1, User2, User3. Пользователь User1 создал объект Table1 и может передать права на работу с ним другим пользователям. Предполагается, что пользователь User2 будет только просматривать таблицу, а User2 может вводить новые записи. В этом случае права можно распределить следующим образом:

GRANT SELECT ON Table1 TO User2;

GRANT INSERT ON Table1 TO User3;

Оператор отмены привилегий

REVOKE { список_операций | ALL PRIVILEGES} ON имя_объекта

FROM {список_пользователей | PUBLIC } {CASCADE | RESTRICT}.

Вариант CASCADE - отмена привилегий не только упомянутых пользователей, но и всех пользователей, получивших привилегии от данного пользователя; RESTRICT – отмена привилегий только упомянутых пользователей.

Секретность в QBE

В QBE признаются права на включение (I.), удаление (D.), обновление (U.) и чтение (P.). Чтобы предоставить пользователю некоторые права на отношение R, владелец этого отношения записывает специальный кортеж в таблицу для R: AUTR(список_прав).имя. Здесь имя является либо именем человека, либо переменной, представляющей произвольное лицо. Если список прав опущен, то предоставляются все четыре права. Если имя опущено, то права предоставляются всем пользователям.

В строке ограничений в колонках атрибутов записываются переменные, константы, выражения. Переменная указывает, что право применяется к данному столбцу. Константа означает, что право распространяется только на кортежи с указанным значением атрибута. Пробел – доступ к атрибуту запрещен. Выражение представляет некоторое условие.

Примеры.

Заказы	Фамилия	Товар	Количество
AUTR(P.).Петров	_n	_i	_q

Петров имеет право читать любую информацию из отношения Заказы.

Заказы	Фамилия	Товар	Количество
AUTR().Петров	_n	_i	_q

Петров может выполнять любые операции над отношением Заказы.

Покупатели	Фамилия	Адрес	Баланс
AUTR(P.)._Петров	_n		>0

Все покупатели могут читать положительный баланс и фамилию, но не адрес.

Покупатели	Фамилия	Адрес	Баланс

AUTR(P.)._Петров	_Петров		_b

Покупатели могут читать только собственные балансы.

Секретность статистических баз данных

Базы данных, позволяющие получать совокупную (агрегированную) информацию о больших подмножествах некоторого множества объектов, называются статистическими. Примерами могут служить базы данных переписи населения, местных органов власти, налоговых служб, медицинские и др. Помимо обычных проблем обеспечения секретности в статистических БД существует проблема запрета доступа к данным, касающихся конкретных лиц. Другими словами, разрешается доступ к совокупной информации большой категории лиц, и запрещается выдача сведений о конкретных лицах.

Проблема возникает вследствие того, что данные о конкретной личности могут быть получены косвенным путем посредством нескольких совокупных запросов. Например, запрос 1 относится к некоторой группе лиц, а запрос 2 к той же группе плюс конкретное лицо. Сравнивая ответы на два запроса, нетрудно сделать вывод о данных интересующего нас лица.

Можно показать, что раскрытие индивидуальных данных максимально осложняется, если разрешается делать запросы только для больших групп лиц, и размер пересечения любых двух запросов должен быть мал. Это соображения используются при реализации защиты статистических баз данных.