Язык SQL: транзакции, представления и права доступа

Виноградова М.В.

Базы данных

МГТУ им. Н.Э. Баумана (ИУ5)

SQL – Structured Query Lanquage

- 1986 – ANSI – SQL/86 (ISO в 1987)
- 1989 - SQL/86 – запросы и схемы
- 1992 - SQL/92 – схемы, транзакции, соединения, авторизация
- 1995 - SQL/CLI – динамический SQL, ODBC
- 1996 - SQL/PSM – хранимые процедуры
- 1999 – SQL:1999 (SQL3) – объектное расширение, UDT
- 2003 – SQL:2003 – OLAP, XML
- 2006 – XQuery

Основные разделы SQL

- Создание/изменение схемы БД – DDL
- Запросы к данным (CRUD) – DML, DQL
- Ограничения целостности и триггеры
- Транзакции
- Представления БД
- Управление доступом
- Структуры физического уровня

Основные объекты БД

- Таблицы (table)
- Представления (view)
- Триггеры (trigger)
- Ограничения целостности (constraint)
- Хранимые процедуры и функции (procedure, function)
- Правила сравнения (collation)
- Домены (domain)
- Последовательности (sequence)

Транзакции

- Набор (одна или более) операций над БД, который выполняется атомарным образом.
- Выполняются либо все операции транзакции, либо ни одна из них.

Требования ACID

- **Atomicity** — Атомарность
  - Фиксация изменений в БД после успешного завершения всех операций (все или никто).
- **Consistency** — Согласованность
  - До и после транзакции БД находится в согласованном состоянии
- **Isolation** — Изолированность
  - Выполнение параллельных транзакции независимо друг от друга (как последовательных)
- **Durability** — Долговечность
  - Результат транзакции сохранен и необратим

SQL операторы транзакции

begin transaction [Имя]

- начать транзакцию

Commit transaction

- Зафиксировать транзакцию

Rollback transaction

- откатить транзакцию

Save transaction Имя_точки

- Создать точку сохранения

Пример простой транзакции с фиксацией

begin transaction

insert into Person values(….)

insert into Workers select from Person …

update Person set ….

commit transaction

Пример простой транзакции с откатом по условию

begin transaction
insert into Person values(….)
insert into Workers select from Person …
update Person set ….
If not exists(select …)
    rollback transaction
else
    commit transaction

Пример точки сохранения

begin transaction t1
    insert into Person values(….)
save transaction point1
    insert into Person values(….)
save transaction point2
    insert into Workers values(….)
    insert into Workers values(….)
rollback transaction point2
    update Person set …..
commit transaction t1

Свойства транзакции

- Фиксация транзакции только если все ограничения целостности БД выполняются.
- Использование журнала транзакций для учета действий при выполнении транзакции.
- Сбой (программный или аппаратный) или нарушение целостности БД при любой операции ведет к откату всей транзакции.
- Если транзакция не указана, то она содержит одну SQL команду.

Уровни изоляции транзакций

- **READ UNCOMMITTED**
  - доступ к «мусору» других транзакций
- **READ COMMITTED**
  - видит только фиксированные данные. Но видит результаты других транзакций
- **REPEATABLE READ**
  - Прочтенные данные всегда видны. М.б. фантомы.
- **SERIALIZABLE**
  - блокировка считанных данных от изменений извне

Определение изоляции транзакций

- SET TRANSACTION ISOLATION LEVEL **READ UNCOMMITTED**

- задает уровень изоляции данной транзакции (что она видит)
- не влияет на работу других транзакций (что они видят)

Особенные транзакции

- Вложенные транзакции
  - Откат внешней ведет в откату внутренних
  - Откат внутренней ведет в откату внешних
  - Фиксация всех только при завершении самой внешней
  - Счетчик транзакций
- Распределенная транзакция
  - Содержит распределенный запрос

Представления (view)

- Объекты БД
- Виртуальные таблицы
- Используются наравне с обычными таблицами
- Не содержат данных
- Хранятся в форме откомпилированного запроса
- При обращении выполняется соответствующий запрос

DDL для представлений

Создание

CREATE VIEW Имя [(поля,...)]
AS SELECT …

Изменение (при сохранении полей)

ALTER VIEW Имя [(поля,...)]
AS SELECT …

Удаление

DROP VIEW Имя

Пример представления

create view Worker **as**

select fio,age

from Person where work is not null;

➔ Worker(fio, age)

Select * from Worker where age>25;

Пример представления с переименованием полей

```
create view Worker(name,vosr) as
   select fio, age
   from Person where work is not null;
```

➔ Worker(name, vosr)

```
Select * from Worker where vosr>25;
```

Использование представлений

- Определение
  - На основе любых запросов на выборку
  - На основе любого количество таблиц или представлений из разных БД
- Использование
  - Наравне в таблицами в любой части select – запросов (from, exists, …)
  - Вместо подзапросов в сложных запросах
  - В любых операциях при соединении таблиц/представлений
  - Назначать на представление индексы и триггеры

Пример представления на основе нескольких таблиц

```
create view Worker as
        select Person.*, Org.*
                from Person join Org on
                Person.work=Org.title;

Select * from Worker join City
   on Worker.city = City.name
   where Worker.post = ‘dir’  and City.capital = true;
```

Обновляемое представление

- Допускает выполнение команд insert, update, delete
- Основано на одной таблице или обновляемом представлении
- Содержит все обязательные и ключевые поля
- Запрос не содержит операторы по преобразованию строк, например, группировки, агрегирования и т.д.

Управление доступом к БД

- Пользователи
  - Идентификаторы авторизации (имена)
  - Получают привилегии и роли
- Роли
  - Именованные наборы привилегий
  - Назначаются пользователям и другим ролям
- Привилегии доступа
  - На объекты БД
  - На выполнение действий

Привилегии доступа

- **SELECT, INSERT, UPDATE, DELETE**
  - для таблиц и представлений (и полей)
- **REFERENCE**
  - Назначений ограничений (на поля)
- **USAGE**
  - Использование/обращение к объекту БД
- **TRIGGER**
  - Создание триггера
- **EXECUTE**
  - Выполнение процедуры или функции
- **ALTER**
  - Создание, изменение и удаление объектов в области
- **ALL PRIVILEGES**
  - Все действия над объектом

Создание пользователя

CREATE LOGIN Имя [WITH опции]

CREATE LOGIN nick WITH password='123'

Опции: пароль, БД, тип авторизации, политики безопасности и т.д.

Авторизация

- **CONNECT TO** сервер **AS** имя **AUTHORIZATION** логин

- подключиться к серверу под именем

- **CURRENT_USER** - имя текущего пользователя

- **SETUSER** логин - изменить пользователя

(*EXECUTE AS USER* = ‘имя')

Назначение привилегий

GRANT список_привилегий
ON объект_БД
TO список_пользователей
[WITH GRANT OPTION]

WITH GRANT OPTION – право назначения привилегий другим

Примеры назначения привилегий

- Grant SELECT, INSERT ON Person to nick, ann with grant option;
- Grant SELECT(fio,age) ON Person to tom;
- Grant ALL Privileges ON Person to jack with grant option;

Аннулирование привилегий

REVOKE список_привилегий
ON объект_БД
FROM список_пользователей
[**CASCADE**]

- Удаление привилегий в соответствии с диаграммой назначений
- CASCADE - отмена прав по каскаду

Аннулирование права на передачу привилегий

REVOKE
GRANT OPTION FOR
список_привилегий
ON объект_БД
FROM список_пользователей
[**CASCADE**]

- Сами привилегии сохраняются.
- CASCADE - назначенные третьим лицам привилегии будут аннулированы.

Роли

- Создание роли в БД

  CREATE ROLE имя;

- Назначение привилегий роли

  GRANT …. TO имя;

- Назначение роли

  GRANT имя **TO** логин, роль;

Пример ролей

CREATE ROLE student;

CREATE ROLE aspirant;

GRANT … ON … TO student;

GRANT student TO ivanov, petrov, aspirant;

GRANT … ON … TO aspirant;

GRANT aspirant TO minina, popov;

REVOKE aspirant FROM popov;

Уровни привилегий

- Учетная запись на сервере (вход)
- Учетная запись в БД (пользователь)

- Привилегии доступа к БД
- Привилегии доступа к схеме БД
- Привилегии доступа к объекту схемы (таблица и тд)
- Привилегии доступа к полям объекта

Владелец объекта

- Тот, кто создал объект
- Имеет все привилегии над объектом
- Может быть изменен
- По умолчанию используется схема владельца