Лекция #8

Базы данных

Михаил Моисеев

Механизм транзакций

Транзакции

Транзакция - группа логически связанных операторов SQL, может выполниться только полностью

- **START TRANSACTION** (неявный запуск транзакции)
- **COMMIT**
- **ROLLBACK**

```
START TRANSACTION;
OPERATOR 1;
OPERATOR 2;
…
COMMIT;
```

ACID

- **Atomicity** – гарантирует, что никакая транзакция не будет зафиксирована в системе частично
- **Consistency** – согласованность с точки зрения логики функционирования системы (частично обеспечивается ограничениями целостности). ИС должна находится в согласованном состоянии до начала транзакции и после ее завершения
- **Isolation** – при выполнении транзакции другие транзакции не должны видеть данные в промежуточном состоянии
- **Durability** – изменения, сделанные успешно завершённой транзакцией, должны сохраниться в ИС независимо от сбоев и отказов

Транзакции #2

Проблемы контроля согласованности при возникновении ошибок

- Частичное выполнение действия состоящего из нескольких операторов

Перевод денег с одного счета на другой:

- Снятие денег со счета 1.
- Зачисление денег на счет 2.

Транзакции #3

- Проблемы контроля согласованности, атомарности и изолированности при многопользовательском доступе
  - Потерянные изменения – данные меняются сначала одним, а потом другим пользователем
  - Чтение незафиксированных изменений – данные меняются одним пользователем, читаются другим пользователем, а потом первый пользователь восстанавливает исходные данные
  - Неповторяемое чтение и фантомные записи – при повторной выборке данных получается другой результат

Транзакции #4

Сериализуемость транзакций - параллельное выполнение заданного множества транзакций будет верным, если при его выполнении будет получен такой же результат, как и при последовательном выполнении тех же транзакций

Сериализуемость обеспечивается с помощью механизма **блокировок** и определения **способов взаимодействия** транзакций

Последовательное выполнение транзакций – блокировка на уровне всей БД

Транзакции #5

Блокировка приводит к остановке транзакции

Типы блокировок

- Для чтению **S**hared-блокировка
- Для записи e**X**clusive-блокировка

	S	X
S	+	-
X	-	-

Уровни блокировок

- Уровень БД
- **Уровень таблиц**
- **Уровень записей**
- Уровень полей

Двухфазный механизм выполнения транзакций

- Накопление(захват) блокировок
- Выполнение операций и освобождение блокировок

Транзакции #6

SET TRANSACTION
[**READ WRITE** | READ ONLY]
[**WAIT** | NO WAIT]
[[ISOLATION LEVEL]
{**SNAPSHOT** [TABLE STABILITY] |
READ COMMITTED
[[NO] RECORD_VERSION]}]
[RESERVING <reserving_clause>];

SET TRANSACTION READ WRITE WAIT SNAPSHOT;

Транзакции #7

Настройки параметров транзакции

- Уровень изоляции транзакции
  - DIRTY READ – чтение незафиксированных изменений (не поддерживается)
  - READ COMMITTED – чтение всех изменений своей транзакции и зафиксированных изменений других транзакций
  - SNAPSHOT (CONCURRENCY) – чтение всех изменений своей транзакции, изменения сделанные в других транзакциях недоступны
  - SNAPSHOT TABLE STABILITY (CONSISTENCY) – запрещает вносить изменения другим транзакциям

Транзакции #8

Настройки параметров транзакции:

- Режим обращения
  - READ WRITE
  - READ ONLY
- Режим обработки конфликтов:
  - WAIT
  - NO WAIT
- Наличие неподтвержденных версий (только READ COMMITED)
  - RECORD VERSION – использует последнюю подтвержденную версию записи
  - NO RECORD VERSION – требует чтобы все версии записи были подтверждены

Транзакции #9

Резервирование таблиц – защита от действий других транзакций.

RESERVING table [, table …]
[**FOR** [**SHARED** | **PROTECTED**] {**READ** | **WRITE**}]

PROTECTED READ другие транзакции не могут читать строки, с которыми работает эта транзакция;
PROTECTED WRITE другие транзакции не могут изменять строки, но могут их читать;
SHARED READ разрешение чтения другим транзакциям;
SHARED WRITE разрешение изменения и чтения другим транзакциям;

RESERVING students FOR PROTECTED WRITE;

Примеры настройки транзакций

- Транзакция для чтения
  - READ ONLY
  - READ_COMMITED
  - WAIT
  - REC_VERSION
- Транзакция для добавления/изменения/удаления
  - READ_COMMITED
  - NO WAIT / WAIT
  - REC_VERSION / NO REC_VERSION
- Транзакция для длительных операций чтения
  - SNAPSHOT
  - NOWAIT
- Транзакция для операций изменения большого количества данных
  - SNAPSHOT TABLE STABILITY
  - NOWAIT

DEADLOCK

Конфликты блокировок транзакций, возникают при попытке двух транзакций установить несовместимые блокировки на один объект.

Модификация одной записи двумя транзакциями

Взаимные блокировки (DEADLOCK) возникают при попытке двух транзакций установить конфликтующие блокировки на два объекта одновременно

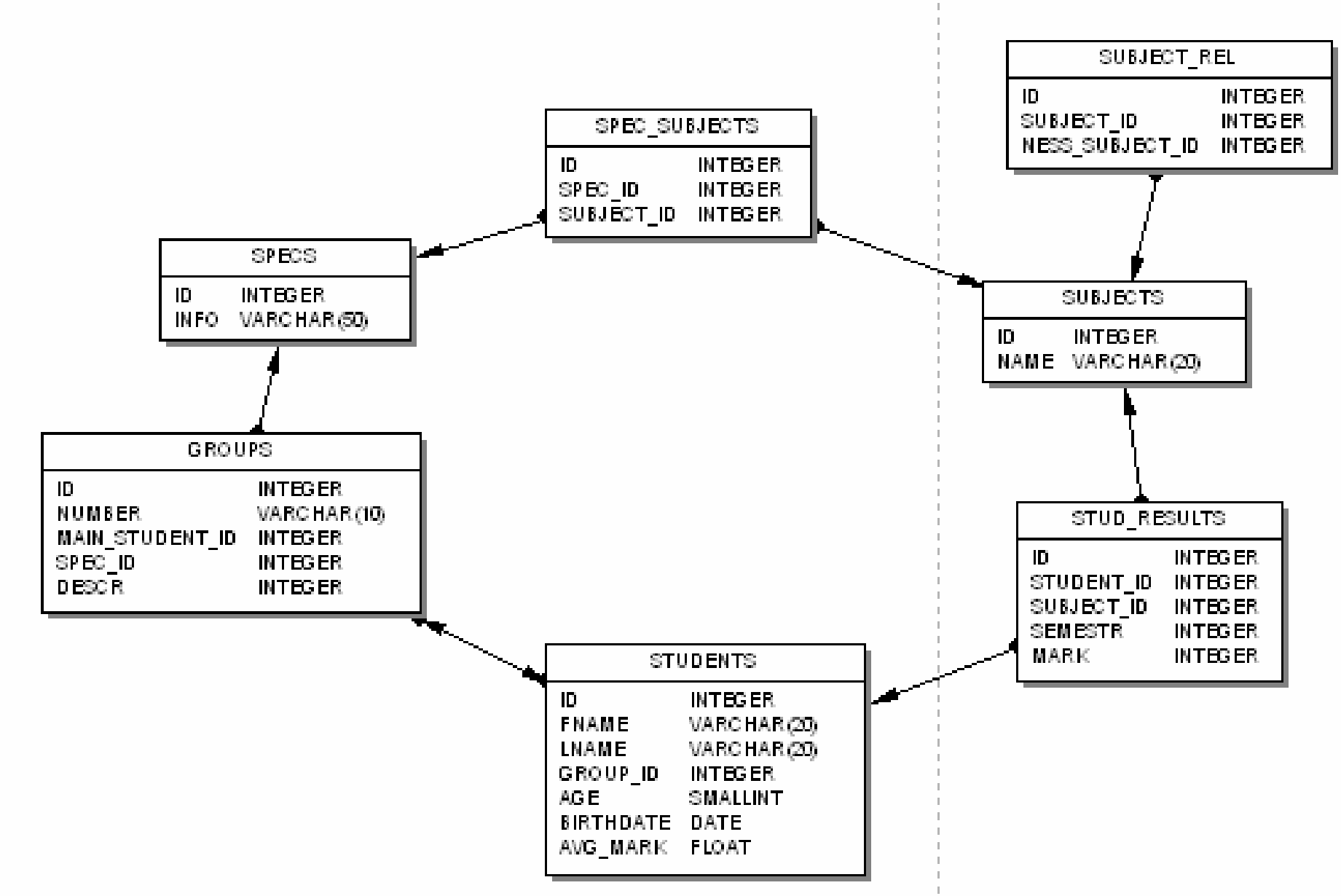
T1 : блокировка X для записи A; блокировка X для записи B
T2 : блокировка X для записи B; блокировка X для записи A

Разрешение DEADLOCK – сервер прекращает одну транзакцию

Вопросы

- Что такое транзакция ?
- Какие проблемы помогают решать транзакции ?
- Каков механизм выполнения транзакций?
- Перечислите основные параметры настройки транзакции ?
- В каких случаях следует применять CONCURRENCY транзакции ?
- В каких случаях следует применять CONSISTANCY транзакции ?
- В каких случаях следует применять READ COMMITED транзакции ?
- Чем отличается NO REC VERSION от REC VERSION ?
- Для чего используется параметр WAIT / NO WAIT ?
- Что такое DEADLOCK, в каких случаях он возникает ?
- Как разрешаются DEADLOCK`и ?

Схема БД


SUBJECT_REL
ID INTEGER
SUBJECT_ID INTEGER
NESS_SUBJECT_ID INTEGER
SPEC_SUBJECTS
ID INTEGER
SPEC_ID INTEGER
SUBJECT_ID INTEGER
SPECS
ID INTEGER
INFO VARCHAR(50)
SUBJECTS
ID INTEGER
NAME VARCHAR(20)
GROUPS
ID INTEGER
NUMBER VARCHAR(10)
MAIN_STUDENT_ID INTEGER
SPEC_ID INTEGER
DESCR INTEGER
STUD_RESULTS
ID INTEGER
STUDENT_ID INTEGER
SUBJECT_ID INTEGER
SEMESTR INTEGER
MARK INTEGER
STUDENTS
ID INTEGER
FNAME VARCHAR(20)
LNAME VARCHAR(20)
GROUP_ID INTEGER
AGE SMALLINT
BIRTHDATE DATE
AVG_MARK FLOAT

Контрольная работа #3

- Написать ХП
  - Рассчитывающую средний балл студента по 20 наилучшим оценкам, для всех студентов и 10 наилучшим оценкам если студент является старостой
- Написать триггеры
  - Проверяющий, что в группе куда добавляют нового студента не более 20 человек, иначе создать новую группу и зачислить в нее добавляемого студента
  - Рассчитывающий средний балл для каждого студента на основе написанной процедуры