CS 1555

Lecture 15

**Integrity Constraints in SQL (continued)**

Mutating trigger

- Recursive call of triggers is not permitted

- Table read in a trigger, it cannot be updated

- INTO: the tuple assignment operator in PL/SQL (programming language for SQL)

- Ex: SELECT MAX(ID) + 1 INTO NEW.ID

- Don’t want to call it while you’re inserting and change the values (race conditions)

- Change trigger from AFTER INSERT to BEFORE INSERT

Final note on integrity constraints

- Assertions and checks support declarative approach of supporting ICs

- Triggers combine declarative and procedural approach of implementing ICs

**Transactions in SQL**

Queries and transactions

- Queries: requests to DBMS to retrieve data

- Updates: insert, delete, modify existing data

- Transaction: logical unit of work

Chicken and the egg problem

- Two tables that reference each other’s primary key (creating dependencies)

- Do we know commands that could create these tables?

- ALTER

- But how can we insert values into either table?

- Need to treat two inserts into both tables as one logical unit of work

Standard SQL transactions

- Set transaction [transaction characteristics];

- Does not start a transaction

- Can be invoked between transactions to set the next transaction to be activated

- Transaction characteristics: READ WRITE | READ ONLY

SQL transactions in Postgres

- START TRANSACTION [READ WRITE | READ ONLY]

- BEGIN [TRANSACTION] [READ WRITE | READ ONLY]

- Should be unnecessary according to SQL standard

- COMMIT: unless START TRANSACTION is issued, it is implicitly added

- Cannot effectively have multi-statement transaction without issuing a start transaction

Transaction atomicity (atomic block)

- “All or nothing” can be achieved with begin-end block

Modes of constraint enforcement

- NOT DEFERRABLE or IMMEDIATE (default)

- Evaluation is performed at input time

- DEFERRED

- Constraints are not evaluated until commit time

- DEFERRABLE

- Can be changed within a transaction to be deferred using SET CONSTRAINTS

- PK can be DEFERRED/DEFERRABLE only if it’s not referenced by another table in Postgres

Changing constraint evaluation mode

- Permitted only for deferrable constraints

- Setting constraint validation mode during a transaction

Specifying transaction atomicity

- Errors at commit time: only when deferred constraints are violated

- No constraint violation of an insert detected at commit time 🡪 whole transaction is committed

SQL savepoints

- In long transactions, isolates set of “safe” operations from set of “risky” operations

- Operations

- SAVEPOINT NAME <name>

- RELEASE SAVEPOINT <name>

- ROLLBACK [WORK] [TO SAVEPOINT name]

ANSI SQL2 isolation levels

- READ UNCOMMITTED

- READ COMMIT

- REPEATABLE READ

- SERIALIZABLE