SQL language: other definitions

Transactions

DBMG

Transactions

- Introduction
- Transactions in SQL
- Properties of transactions

Transactions

Introduction

Example of application

- Banking operations
  - cash withdrawal from a current account using a cash card
  - depositing cash on a current account

Cash withdrawal

- Operations
  - specify the amount required
  - check availability
  - memorize transaction
  - update balance
  - enable withdrawal of the amount required
- All the operations have to be carried out correctly, otherwise the cash cannot be withdrawn

Cash withdrawal

- *What happens if a co-signatory of a joint account makes another cash withdrawal?*
- *What happens if there is a malfunction?*

Cash deposit

- Operations
  - check the amount paid in
  - memorize the transaction
  - update the balance

- All the operations have to be carried out correctly, otherwise the cash cannot be deposited

Cash deposit

- *What happens if another person pays cash into the same account?*
- *What happens if there is a malfunction?*

Example: banking operations

- The bank database is a multi-user environment
  - various operators can work simultaneously on the same portion of data
- The correct management of the information requires
  - mechanisms for the management of *simultaneous access* to the database
  - Mechanisms for the *recovery* of the correct state of the database in the case of malfunction

Transactions

- It is necessary when several users can simultaneously access the data
- It provides efficient mechanisms for
  - managing competing access to data
  - recovery after a malfunction

Transactions

- A transaction is a sequence of operations that
  - represents an elementary unit of work
  - can end in success or failure
    - in the case of success, the result of the operations has to be permanent in the database
    - in the case of failure, the database has to return to the original state before the transaction was initiated

Transactional system

- A system that makes a mechanism available for the definition and execution of transactions is called a *transactional system*
- The DBMS contain architecture blocks that offer transaction management services

Transactions in SQL

Transaction

- A transaction is
  - a logical unit of work, which cannot be broken down any further
  - a sequence of operations (SQL instructions) to change data, which takes the database from a consistent state to another consistent state
    - it is not necessary to conserve the consistency of the intermediate states

Beginning a transaction

- To define the beginning of a transaction, the SQL language uses the instruction
    - START TRANSACTION
- Usually the instruction to begin a transaction is omitted
    - the beginning is implicit for
        - the first SQL instruction of the programme that accesses the database
        - the first SQL instruction following the instruction ending the previous transaction

Ending a transaction

- The SQL language has instructions for defining the end of a transaction
  - Transaction successful
    - COMMIT [WORK]
    - the action associated with the instruction is called *commit*
  - Transaction failed
    - ROLLBACK [WORK]
    - the action associated with the instruction is called *abort*

Commit

- Action executed when a transaction ends with success
- The database is in a new (final) correct state
- The changes to the data executed by the transaction become
  - permanent
  - visibile to other users

Rollback

- Action executed when a transaction ends because of an error
  - for example, an error in application
- All the operations modifying the data executed during the transaction are “annulled”
- The database returns to the state prior to the beginning of the transaction
  - the data is once more visible to the other users

Example

- Transfer the sum of 100
  - from current account number IT92X0108201004300000322229
  - to current account number IT32L0201601002410000278976

```
START TRANSACTION;
UPDATE Account
    SET Balance= Balance + 100
    WHERE IBAN='IT92X0108201004300000322229';
UPDATE Account
    SET Balance = Balance - 100
    WHERE IBAN= 'IT32L0201601002410000278976';
COMMIT;
```

Transactions

Transaction properties

Transaction properties

- The principal properties of transactions are
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- They are summarized by the English acronym *ACID*

Atomicity

- A transaction is an indivisible unit (atom) of work
  - all the operations contained in the transaction have to be executed
  - or none of the operations contained in the transaction have to be executed
    - the transaction has no effect on the database
- The database cannot remain in an intermediate state arrived at during the processing of a transaction

Consistency

- The execution of a transaction has to take the database
  - from an initial state of consistency (correct)
  - to a final state of consistency
- Correctness is verified by integrity constraints defined on the database
- When there is a violation of the integrity constraint the system intervenes
  - to annul the transaction
  - or to modify the state of the database by eliminating the violation of the constraint

Isolation

- The execution of a transaction is independent from the simultaneous execution of other transactions
- The effects of a transaction are not visible by other transactions until the transaction is terminated
  - the visibility of unstable intermediate states is avoided
    - an intermediate state can be annulled by a subsequent rollback
    - in the case of rollback, it is necessary to rollback the other transactions that have observed the intermediate state (domino effect)

Durability

- The effect of a transaction that has executed a commit is memorized permanently
  - the changes to the data carried out by a transaction ending successfully are permanent after a commit
- It guarantees the reliability of the operations of data modification
  - the DBMS provides mechanisms for recovery to the correct state of the database after a malfunction has occurred