DATA BASES 2

Federico Mainetti Gambera

25 settembre 2020

Indice

1 introduction

Introduction slides:
`../other/professor'sslides/0-Intro.pdf`

1.1 References to Relational Algebra and SQL

Relational Algebra

`../other/RelationalAlgebraAndSQL/04-RelationalAlgebra.pdf`

`../other/RelationalAlgebraAndSQL/05-RelationalAlgebra2.pdf`

SQL

`../other/RelationalAlgebraAndSQL/08-SQL1.pdf`

`../other/RelationalAlgebraAndSQL/09-SQL2.pdf`

`../other/RelationalAlgebraAndSQL/10-SQL3.pdf`

1.2 Books

All books can be found at:
`../other/books`

2 Transactional Systems

`../other/professor'sslides/1_Transactions_NEW.pdf`

A **transaction** is an elementary, atomic, unit of work performed by an application.
Each transaction is conceptually encapsulated within two commands:

- **begin of transaction** (bot);

- **end of transaction** (eot).

Within a transaction, **one** of the commands below is executed (**exactly once**) to signal the end of the transaction:

- **commit**;

- **rollback**.

Transactional System (OLTP) is a system that supports the execution of transactions on behalf of concurrent applications.

A transaction is a unit of work enjoying the **ACID** properties:

- **Atomicity** (abort-rollback-restart, commit protocols): either all the operations in the transaction are executed or none is executed;

- **Consistency** (integrity checking of the DBMS): a transaction must satisfy the DB integrity constraints, if the initial state is consistent then the final state is also consistent, but this is not necessarily true for the intermediate states of the transaction;

- **Isolation** (concurrency control): the concurrent execution of a number of transaction must produce the same result as the execution of the same transactions in a sequence;

- **Durability** (recovery management): the effect of a transaction that has successfully committed will last "forever".

Where are the ACID properties managed in the DBMS architecture?
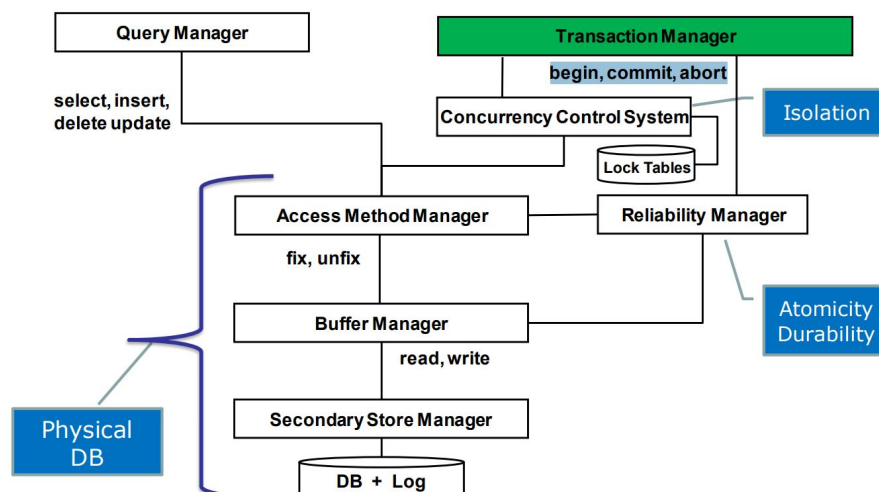
3 Concurrency Control

../other/professor'sslides/2-ConcurrencyControl.pdf