

Introduction

The DBMS (Database Management System) is a software package created with two purposes:

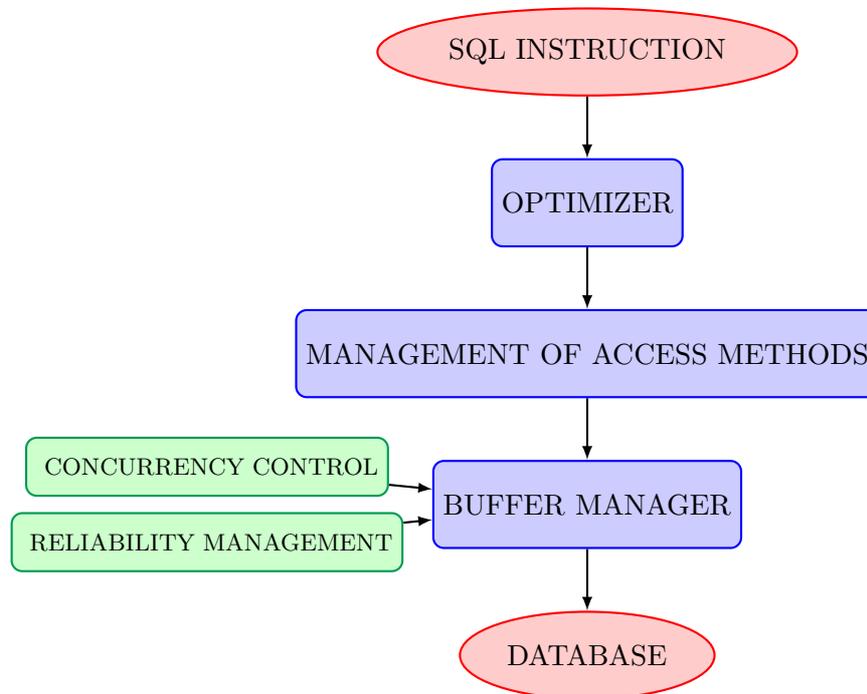
- . store information on a database;
- . manage the database.

One of the most important property is the *data independence property*: the user does not know where files are physically located but he operates with DBMS. In this way he can write a query and obtain results: it is possible only because DBMS is written in logical mode.

Architecture

The DBMS architecture is made of the following modules:

- . optimizer;
- . management of access methods;
- . buffer manager:
 - . concurrency control;
 - . reliability management;
- . database, system catalogue:
 - . index files;
 - . data files.

Scheme**Optimizer**

This module guarantees data independence property and its function is to select the best strategy to access data in order to answer queries; substantially it defines the order to execute operations.

Access Method Manager

This module has to perform the physical access to data and to use the strategy chosen by the optimizer.

Buffer Manager

Its functions are:

- . manage the data transfer from the secondary memory to the main memory;
- . manage the main memory.

Concurrency Problem

This module avoids that concurrent processes access to data in the same time; in particular, it manages the *consistency problem* (especially doing writing operations).

Reliability Manager

This module guarantees:

- . independent execution of transactions;
- . correctness of database if there are system crashes.

Transactions

A transaction is a logical unit of work, for example a read table. Transactions are characterized by:

- . correctness;
- . reliability;
- . isolation.

A transaction can start:

- . in implicit mode;
- . if is the first SQL instruction:
 - . at the beginning of a program;
 - . after the end of a transaction.

A transaction can end:

- . correctly (COMMITTED);
- . with errors (ROLLBACK):
 - . suicide if rollback is required by the transaction;
 - . murderer if rollback is required by the system.

Properties

Transactions must satisfy four properties:

- . atomicity;
- . consistency;
- . isolation;
- . durability.

Atomicity

A single transaction is the smallest part: it can not be divided more. In order to guarantee this property there are two operators:

- . *redo*: failure after committed transaction (data is not written into the secondary memory);
- . *undo*: used in case of rollback transactions.

Consistency

This property guarantees *integrity constraints*, so when there are errors (primary or foreign keys wrong, domain constraints) the system can:

- . rollback the transaction;
- . correct himself the violation.

Isolation

Each execution of a transaction is processed independently of another execution of another transaction. It is guaranteed by the Concurrency Control block.

Durability

Guaranteed by the Reliability Manager operates in order to maintain the effect of a committed transaction in presence of failures.