Database Systems II B: DBMS-Implementation

Chapter 1: Introduction

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2021/22

http://www.informatik.uni-halle.de/~brass/dbi21/

Objectives

After completing this chapter, you should be able to:

- explain some functions of a DBMS.
- give an overview of the architecture of a DBMS.

  Name some important components.
- enumerate tasks of the database administrator.

Contents

Database Services (1)

- The main task of a database management system (DBMS) is to manage persistently stored data.

  Persistent means that the data needs to be remembered longer than a single program execution, it even has to survive a shutdown of the OS. Today this basically means it must be stored on disk.

- The data is shared between many users and many application programs.
- The DBMS acts as a server:
  - It receives requests (queries and updates) over the network
  - and sends responses (answers, results) back.
- The DBMS is usually a set of background processes.

  Like a web server, but much more complex.

Database Services (2)

- Most of the queries and updates are executed by application programs.

  Ad-hoc queries are relatively seldom in practice. In a heavily loaded system, the DBA will forbid that ad-hoc queries are entered during standard working hours.

- Therefore, most of the queries/updates are known in advance, including estimates for their frequency (e.g. 5 per min).

  "Load profile": List of DB commands with frequencies. Of course, the queries/updates are not completely known. They usually contain parameters in place of constants. (In JDBC, parameters are marked with "?", in embedded SQL they are written `:X` with a program variable `X`.) Values for the parameters are given at runtime when the command is executed.

Database Services (3)

- The data is an important asset of the company. Hardware/software failures should not lead to a loss or corruption of data.

  Transactions are an important concept to protect the data in the presence of failures (see below).
  The DBMS offers the tools for protecting the data, but it must be correctly configured by a knowledgeable DBA. One must also check regularly whether the automatic backup processes really work.

- Some systems must run 7 days a week, 24 hours a day (very high availability requirements).

- Many users are (usually) working with a DBMS at the same time. This requires synchronization of concurrent accesses, e.g., via locks.

Database Services (4)

- Not every user is allowed to do everything:
  The DBMS must manage access rights for each user.

  It might also have to identify (authenticate) the users, if this service is not taken from the operating system.

- Other database services are, e.g.:
  - Integrity enforcement,
  - views,
  - stored procedures, triggers,
  - system catalog (Data Dictionary).

Contents

Physical Data Independence (1)

- Application programs depend only on the logical structure of the data (e.g. tables and columns).
- Queries and updates in SQL do not depend on the way the data is stored, e.g.
  - how the tables are distributed over disks,
  - which access paths / indexes exist,

    An index over attribute A of relation R is a data structure that permits efficient access to the tuples of R with a given value for A.
  - how free-space management parameters are set.

    These determine where on the disk a new row is stored.
    E.g. it might be possible that rows are stored clustered by an attribute (rows with the same value are stored near to each other).

Physical Data Independence (2)

- SQL is a declarative query language:
  - An SQL query specifies only what information is sought,
  - but does not prescribe any particular method how to compute this information.
- Declarative languages often allow simpler/shorter formulations.

  The user does not have to think about efficient execution.

  Declarative languages require a powerful optimizer, but they also make it possible, because the language does not prescribe a specific execution sequence.
- Physical storage details are abstracted away on the SQL level.

Physical Data Independence (3)

- The DBMS automatically translates the given SQL query into a query evaluation plan (QEP) which is then executed to compute the result of the query.

  The translation is done by the query optimizer. A QEP is a program for the execution engine in the DBMS. QEPs is also called access plans or execution plans. They are similar to relational algebra expressions.

- The physical parameters (indexes etc.) do influence
  - the performance of query evaluation and
  - the required disk space.

- If an index is used for query evaluation, it is explicitly mentioned in the QEP.

  Thus, QEPs are on a somewhat lower implementation level than relational algebra expressions. Also duplicate elimination and sorting are operators in QEPs.

Physical Data Independence (4)

- The task of physical DB design is to find good settings for the physical parameters, e.g. to select indexes.
- The physical design needs to be modified from time to time because
  - the size of the database objects changes,
  - the invocation frequency of programs changes,
  - new applications are developed.
- Because of physical data independence, application programs are not affected by this modification.

Performance Tuning

- The goal of performance tuning is to meet given performance requirements. Techniques are:
  - Modifying the physical design.
  - Changing parameters of the DBMS or the OS.

    E.g. sizes of certain main-memory areas (buffer cache).
  - Extending the given hardware.

    Buying e.g. more main memory or additional disks.
  - Changing the application programs.
  - Changing logical design (e.g. denormalization).
  - Changing business rules (requirements).

Contents

Transactions (1)

- A transaction is a sequence of DB commands, particularly updates, which the DBMS treats as a unit.

  E.g. a transfer of 50 dollars from account 1 to account 2 consists of
  (1) checking the current balance/credit limit of account 1,
  (2) decreasing the balance of 1 by 50 (debit),
  (3) increasing the balance of 2 by 50 (credit).

- Transactions have the ACID-Properties:
  Atomicity, Consistency, Isolation, Durability.

- The successful end of a transaction is marked with the SQL command `COMMIT`.

  One can explicitly declare the unsuccessful end with the command `ROLLBACK`. Implicitly this happens when the program crashes.

Transactions (2)

Atomicity:

- DBMS guarantee that each transaction is either executed in total, or not at all.

  If the transaction cannot be executed until the end (e.g. because of a power failure or system crash), the DB state before the transaction has begun will be restored when the DBMS is started the next time.

- As long as the transaction has not been declared as complete all changes can be undone (`ROLLBACK`).

- I.e. the DBMS needs to log changes / remember old versions until the transaction is finished.

  This is the purpose of the rollback segments or undo tablespace in Oracle.

Transactions (3)

Durability:

- When a DBMS acknowledges the `COMMIT`, it guarantees that the changes are durable.
- The changes are stored on disk — they are not lost even if there is a power failure one second later.

  In operating systems, one often cannot be sure whether the data is on disk or still in a buffer. In a typical DBMS (like Oracle), the changes are not immediately written to the right place on disk, but to sequential file, the redo log (for efficiency reasons).

- Larger DBMS have powerful backup and recovery mechanisms: Even if a disk fails, no data is lost.

  With OS utilities, typically only one backup per day is created.

Transactions (4)

Atomicity and Durability Together:

- There is one point in time that lies between
  - the user telling the system that the transaction is complete (`COMMIT`), and
  - the system telling the user that this command was successfully processed,

  when all changes become effective.

  If the system crashes before this point, the database state is not changed.
  If the system crashes after this point, the database state contains all changes that the transaction has executed. In a typical DBMS, this is the point when the entry for the `COMMIT` is completely contained in the redo log.

Transactions (5)

Isolation:

- Different processes can access the database concurrently. Without control, this could have strange effects including lost updates.

  DBMS try to create the impression that every transaction runs in isolation, i.e. has exclusive access to the complete DB. The theoretical goal of "Serializability" is not quite reached in practice, some support from the programmer is needed.

- Usually, a DBMS manages locks on database objects (tables, rows, table entries) for this purpose.

  Different types of locks (e.g. shared, exclusive) are used. For the most part, lock management is done automatically by the DBMS. Locks are typically kept until the end of the transaction.

Transactions (6)

Consistency:

- User and system can be sure that the current state is the result of a sequence of completely executed transactions.

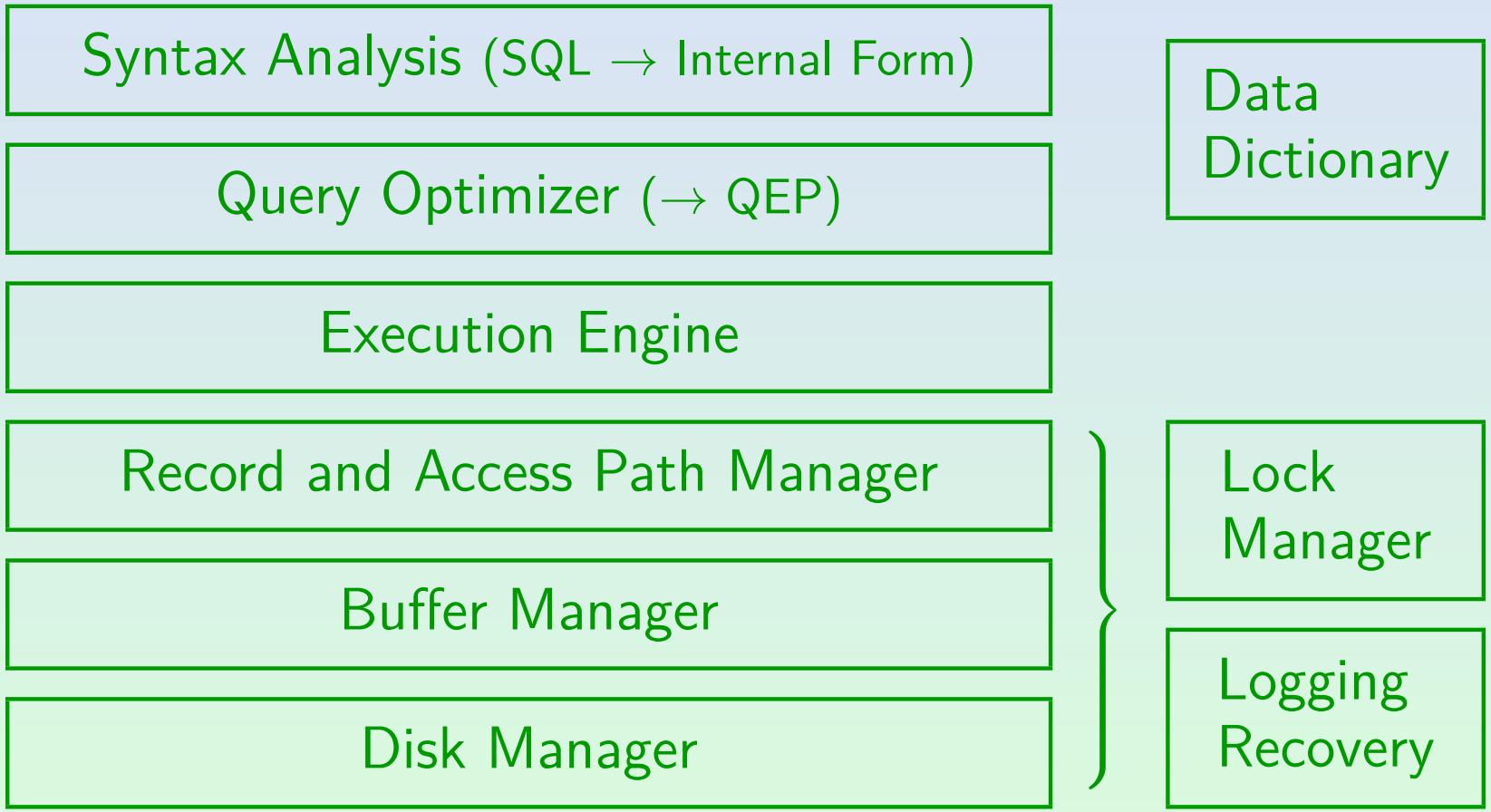
  E.g., it is not possible that a tuple was entered into a table, but not into the index on this table because of a power failure in between.

- The user must ensure that each transaction, if applied fully and in isolation to a consistent state, will produce a consistent state.

- Modern DBMS offer some support for this:
  Declarative constraints, triggers, stored procedures.

Contents

DBMS Architecture: Example



Contents

DB Administrator Tasks (1)

- The DBA ensures that the DBMS keeps running:
  - Monitors available disk space, installs new disks.

    If disks fill up, the system will come to a standstill.
  - Ensures that backup copies are made.
    Does the recovery after disk failures etc.
  - Kills sessions of users who locked an important object and then went to lunch.
  - Monitors performance, ensures that users do not monopolize the system.

    He/she administers quotas (disk, CPU) and usage rules.

DB Administrator Tasks (2)

- The DBA ensures security and confidentiality of the data (some companies have a special security administrator):
  - The DBA creates new user accounts.

    And deletes or locks unused accounts.
  - The DBA manages access rights to DB objects.
  - The DBA might do auditing (who did what) and analyzes the collected data for suspicious events.
- The DBA sometimes needs powerful privileges for the OS (operating system). He/she can damage everything.

DB Administrator Tasks (3)

- The DBA tries to ensure data correctness.
- The DBA does performance tuning.

  Sometimes, external experts are asked to do this.
- The DBA should be an expert for
  - the DBMS software,

    Oracle 8 had $\geq$ 95 volumes (= 1.70 m) of documentation.
  - the database schema (or schemas).
- In addition, he/she should know the users of the system.

  If the company has thousands of employees, distributed over many countries in the world, is is not realistic. Then more information is needed in a user database and automatic checks should be applied to make sure that there is no suspicious behaviour.

DB Administrator Tasks (4)

- The DBA installs new versions of the DBMS software.
- The DBA is the technical contact point for the DBMS vendor.

  E.g., for support, information about security holes.
- The DBA is responsible for keeping the terms of the software licence agreement.
- In all new developments of application programs for this database the DBA has a say.

References

- Ramez Elmasri, Shamkant B. Navathe: Fundamentals of Database Systems, 3rd Ed. Chapter 17: "Database System Architectures and the System Catalog", Chapter 10: "Examples of Relational Database Management Systems: Oracle and Microsoft Access"
- Raghu Ramakrishnan, Johannes Gehrke: Database Management Systems, 2nd Ed. McGraw-Hill, 2000, ISBN 0-07-232206-3, 906 pages.
- H. Garcia-Molina, J. D. Ullman, J. Widom: Database System Implementation. Prentice Hall, ISBN 0130402648, 672 pages, ca. $60.00
- Jason S. Couchman: Oracle8i Certified Professional: DBA Certification Exam Guide with CDROM. Osborne/ORACLE Press, ISBN 0-07-213060-1, ca. 1257 pages, ca. $99.99.
- Jim Gray, Andreas Reuter: Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers, 1993, ISBN 1-55860-190-2, 1070 pages, ca. $84.95
- Oracle 8i Concepts, Release 2 (8.1.6), Oracle Corporation, 1999, Part No. A76965-01.
- Oracle 8i Administrator's Guide, Release 2 (8.1.6), Oracle Corporation, 1999, Part No. A76956-01.