DBMS And Database Server Software

Asked 10 years, 1 month ago Modified 10 years, 1 month ago

Viewed 1k times

1

OK.. I have done research on this but the more I look into this the more I get confused. So thought of posting it here.

- DBMS provides database server functionality
- DBMS has a server component and a client component
- DBMS = database server software
- DBMS operates on a server/client architecture

These are the four concepts I come across when I tried to understand the difference between the **database server software and DBMS**. Could some one here clarify which is the right concept please?

mysql database rdbms database-server

Share

Improve this question

Follow

asked Nov 13, 2014 at 0:06

higgs_boson

119 ● 1 ● 6

1 Answer

Sorted by: Highest score (default)

4

✓

Taken from here:

> A **database** is an organized collection of data. The data are typically organized to model aspects of reality in a way that supports processes requiring information. For example, modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.
>
> **Database management systems** (***DBMS*** s) are computer software applications that interact with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases.

So, in short, a DBMS is the piece of software responsible for the storage and management of the data stored in the database, and it interacts with the user, other pieces of software and the database itself.

Notice that not all the DBMSs have a client-server architecture, and there's not a specific model for the data stored in the database. The most used database model in this days is the relational model (many popular DBMSs are Relational DBMSs, or *RDBMS*s), but there are other models (object-relational, document-oriented, etcetera)

Examples of DBMSs which follow a client-server architecture:

- MySQL
- PostgreSQL
- SQL Server
- MongoDB (a NoSQL DBMS)

Examples of DBMSs which do not follow a client-server architecture ("embeded"):

- SQLite (Android apps use this a lot)
- Apache Derby (can also be used as a client-server DBMS)
- HSQLDB (can also be used as a client-server DBMS)
- MS Access (it can be a client, but implementing it as a database server ~~is a real pain in the neck and a source of sorrow and tears~~ is kinda hard and not very effective for big projects)

So, taking your list:

- *DBMS provides database server functionality*
  Not necesarily: DBMS can work in embedded mode (an application has exclusive access to the

database, and the DBMS is embedded into the application)

- *DBMS has a server component and a client*
  Not necesarily: There are embedded DBMSs for which there's no server, but only an app that has access to the database.
- *DBMS = database server software*
  Not necesarily, but frequently
- *DBMS operates on a server/client architecture*
  Not necesarily

About your specific question ("Difference between database server software and DBMS")... well, many DBMSs *are* database servers. As the piece of software responsible for the storage and management of the data, the DBMS has also the responsability of dealing with the specific way that "users" (either human or apps) access the data. When working with embedded DBMSs, there's no server: the data can be stored in files accessible only to the app in which the DBMS is embedded, or even (in some cases) in the same file as the app. The more common client-server architecture has some characteristics (depending on your specific needs, they can be advantages or obstacles):

- The data is stored in files managed by the server component of the DBMS
- The DBMS has a client component, which is responsible for the users' access and manipulation of

the data

- The server part can also implement security measures (e.g. permissions) for the access and manipulation of the data
- The client can be a specific tool of the DBMS (e.g. MySQL has a server component and a command-line client), a separated program (e.g. MySQL Workbench, which can connect directly to a MySQL server) or a library which can be used to connect your app to the server (e.g. The MySQL connector/J is a Java library which can be used by a Java program to access the MySQL server)
- A client-server architecture allows you to create databases that are accessible by one or more users, each one with specific permissions

... and so on

Share Improve this answer Follow

edited Jun 20, 2020 at 9:12

Community Bot
1 ● 1

answered Nov 13, 2014 at 0:46

Barranka
21k ● 14 ● 71 ● 88