



Câu hỏi ôn tập chương 1 - Bài tập Hệ quản trị csdl

Hệ quản trị cơ sở dữ liệu (Trường Đại học Sư phạm Kỹ Thuật Thành phố Hồ Chí Minh)



Scan to open on Studocu

Chương 1.

1. Định nghĩa Cơ sở dữ liệu (database)?

- Một tập hợp các dữ liệu có liên quan với nhau trong một lĩnh vực cụ thể.

Vd. Một phần CSDL trong một ứng dụng quản lý SV của một trường đại học bao gồm các dữ liệu liên quan tới SV, các môn học và điểm thi, v.v...

2. Các chức năng của hệ quản trị CSDL

- Cung cấp môi trường tạo lập cơ sở dữ liệu
- Cung cấp môi trường cập nhật và khai thác dữ liệu
- Cung cấp công cụ điều khiển truy cập vào CSDL

3. Ngôn ngữ định nghĩa dữ liệu là gì? Cho ví dụ.

- Các câu lệnh được sử dụng để định nghĩa và quản lý các đối tượng CSDL như bảng, khung, chỉ mục, được gọi là ngôn ngữ định nghĩa dữ liệu.

VD: Về cơ bản ngôn ngữ định nghĩa dữ liệu gồm:

CREATE: Định nghĩa và tạo mới đối tượng CSDL

ALTER: Thay đổi định nghĩa của đối tượng CSDL

DROP: Xoá CSDL đã có

4. Ngôn ngữ thao tác dữ liệu là gì? Cho ví dụ

- Ngôn ngữ thao tác dữ liệu là ngôn ngữ để người dùng diễn tả yêu cầu cập nhật hay khai thác thông tin, các thao tác gồm:

- + Cập nhật (nhập, sửa, xoá dữ liệu);
- + Khai thác (sắp xếp, tìm kiếm, kết xuất báo cáo, ...)

5. Ngôn ngữ kiểm soát dữ liệu và ngôn ngữ kiểm soát transaction? cho ví dụ.

- Ngôn ngữ kiểm soát dữ liệu (DCL) là một tập hợp con của SQL và cho phép các quản trị viên cơ sở dữ liệu cấu hình truy cập bảo mật cho các cơ sở dữ liệu quan hệ.

VD: GRANT, REVOKE, DENY

- Ngôn ngữ kiểm soát transaction (TCL) là một ngôn ngữ máy tính và là một tập con của SQL, được sử dụng để điều khiển quá trình xử lý giao dịch trong cơ sở dữ liệu

VD: COMMIT, ROLLBACK, SAVEPOINT, AUTOCOMMIT

6. Nhiệm vụ của công cụ kiểm soát, điều khiển truy cập vào CSDL?

- + Duy trì tính nhất quán của CSDL
- + Khôi phục CSDL khi có sự cố
- + Phát hiện và ngăn chặn sự truy cập không được phép
- + Tổ chức và điều khiển các truy cập đồng thời

+ Quản lý các mô tả dữ liệu.

7. Các loại đối tượng sử dụng CSDL?

+ Người quản trị CSDL (DBA – DataBase Administrator)

- Cấp quyền khai thác CSDL
- Xem xét việc sử dụng CSDL và thu hồi tài nguyên
- Bảo mật và thời gian đáp ứng yêu cầu của hệ thống

+ Người thiết kế CSDL (Database designer)

- Hiểu yêu cầu của người dùng và tạo một thiết kế đáp ứng yêu cầu.
- Xác định dữ liệu cần lưu trong CSDL và cấu trúc CSDL

+ Người dùng cuối (End user)

- Người dùng ngẫu nhiên
- Người dùng thường xuyên
- Người dùng cho các công việc phức tạp
- Người dùng một mình

+ Phân tích viên hệ thống (System Analyst) và lập trình viên ứng dụng (Application Programmer)

- Phân tích viên hệ thống xác định yêu cầu của người dùng và xây dựng bản đặc tả cho những giao tác đáp ứng yêu cầu người dùng.
- Lập trình viên cài đặt đặc tả thành chương trình, kiểm tra, dò lỗi, ghi sơ liệu và bảo trì những giao tác này

8. Hệ quản trị CSDL (DBMS) là gì?

- Hệ quản trị cơ sở dữ liệu (DBMS – Database Management System) là hệ thống được xây dựng để quản lý tự động các cơ sở dữ liệu một cách có trật tự. Các hoạt động quản lý bao gồm lưu trữ, tìm kiếm, chỉnh sửa và xóa thông tin trong một nhóm dữ liệu nhất định.

9. Kể tên một vài DBMS thông dụng

- Các Hệ QT CSDL: SQL-Server, DB2, Paradox, Informix, Oracle, MongoDB, DB2, Redis, Elasticsearch. ...

10. Một hệ CSDL?

- Bao gồm một DBMS cùng với một CSDL. Đôi khi có thể bao gồm cả chương trình ứng dụng.

11. Siêu dữ liệu (meta-data)?

- Hệ CSDL không chỉ lưu CSDL mà còn lưu trữ định nghĩa về cấu trúc dữ liệu và các ràng buộc trong catalog. Thông tin được lưu trữ trong catalog được gọi là siêu dữ liệu.

12. Ưu điểm của DBMS

- Cung cấp nhiều phương thức để lưu trữ và truy xuất các dữ liệu có nhiều định dạng khác nhau bằng cách sử dụng ngôn ngữ truy vấn (Query Language).
- Dễ dàng được duy trì.
- DBMS hỗ trợ nhiều ứng dụng có thể sử dụng trong cùng một hệ cơ sở dữ liệu với thời gian phát triển và duy trì được tối ưu nhất.
- Bảo mật và toàn vẹn dữ liệu với hoạt động cập nhật và sao lưu dữ liệu tối thiểu.
- DBMS còn tương thích với các ngôn ngữ lập trình ứng dụng như Java và Python nhằm mục đích hỗ trợ người dùng kết nối cơ sở dữ liệu với bất kỳ ứng dụng hay một trang web nào.
- Có chức năng sao lưu tự động và phục hồi khi cần.
- Cho phép người dùng có thể truy cập và chia sẻ dữ liệu.

13. Mô hình dữ liệu (data model) là gì?

- Một tập hợp các khái niệm dùng để mô tả cấu trúc của cơ sở dữ liệu, là một mô hình trừu tượng tổ chức các phần tử của dữ liệu và chuẩn hóa cách chúng liên quan với nhau và với các thuộc tính của các thực thể trong thế giới thực.

14. Database designer có nhiệm vụ?

- Thiết kế cơ sở dữ liệu (database designer hay DBD) có nhiệm vụ xác định những dữ liệu nào cần được lưu trữ, cấu trúc của những dữ liệu ấy, phương pháp thể hiện và lưu trữ các dữ liệu này.

15. System Analyst có nhiệm vụ?

- Định cấu hình và lập kế hoạch cho hệ thống máy tính và cơ sở hạ tầng cho một doanh nghiệp.
- Trợ giúp trong quá trình phát triển hệ thống để tạo ra thiết kế tổng thể.
- Tạo và giúp viết tài liệu và yêu cầu.
- Thực hiện kiểm tra hệ thống.
- Tương tác với các nhà phát triển và hiểu mọi giới hạn hoặc các vấn đề đã biết của chương trình.
- Tương tác với người dùng cuối và thu thập các yêu cầu và yêu cầu.

16. Application Programmer có nhiệm vụ?

- là một loại người dùng cơ sở dữ liệu (database user), họ là các chuyên gia máy tính viết các chương trình ứng dụng. Họ là các lập trình viên và phát triển các chương trình như C, C++, Java,... Họ cung cấp các giao diện người dùng (user interface) để người dùng khác có thể tương tác với cơ sở dữ liệu.

17. Lược đồ (schema) là gì?

- Schema trong SQL Server là gì Lược đồ (schema) là tập hợp các đối tượng cơ sở dữ liệu bao gồm table, view, trigger, stored procedure, index.

18. Phân loại mô hình dữ liệu

- Mô hình phân cấp
- Mô hình mạng
- Mô hình quan hệ
- Mô hình liên kết
- Mô hình hướng đối tượng

19. Mô hình dữ liệu mức khái niệm?

- Mô hình dữ liệu mức cao (mức khái niệm): cung cấp các khái niệm gần với cách người dùng cảm nhận về dữ liệu.

Vd. Mô hình thực thể kết hợp là mô hình dữ liệu mức khái niệm.

20. Mô hình dữ liệu mức vật lý ?

- Mô hình dữ liệu mức thấp (mức vật lý): mô tả cách dữ liệu được lưu trữ trên máy tính.

Vd. File trên máy tính được mô tả bởi định dạng file, chỉ mục, ...

21. Mô hình dữ liệu mức logic ?

- Mô hình dữ liệu mức logic: mức trung gian giữa 2 mức trên để người dùng có thể hiểu nhưng cũng gần với cách dữ liệu được tổ chức trên đĩa cứng. Mô hình này dấu đi chi tiết lưu trữ dữ liệu trên đĩa nhưng có thể cài đặt trên máy tính.

Vd. Mô hình quan hệ, mô hình phân cấp, mô hình mạng là các mô hình dữ liệu mức logic.

22. Kiến trúc 3 mức trừu tượng của một hệ CSDL? Công dụng của nó?

+ Theo kiến trúc ANSI-PARC, một CSDL có 3 mức biểu diễn

+ Kiến trúc 3 mức giúp tách biệt các ứng dụng người dùng với CSDL vật lý.

+ Thay đổi về khía cạnh vật lý trong lưu trữ như thay thiết bị nhớ thứ cấp có thể không làm ảnh hưởng đến cấu trúc bên trong CSDL; Người quản trị CSDL có thể thay đổi cấu trúc tổng quát hay cấu trúc khái niệm của CSDL mà không làm ảnh hưởng đến tất cả người dùng.

+ Bao gồm: Mức vật lý, mức logic/quan niệm, mức ngoài/ view.

23. Mức vật lý trong kiến trúc 3 mức của 1 hệ CSDL?

- Mức vật lý: Sử dụng mô hình dữ liệu vật lý để mô tả cấu trúc lưu trữ vật lý của CSDL

24. Mức logic trong kiến trúc 3 mức trừu tượng của 1 hệ CSDL?

- Mức logic/quan niệm: Dấu đi chi tiết về cấu trúc lưu trữ vật lý. Dùng mô hình dữ liệu logic để mô tả cái gì được lưu trữ trong CSDL và mối quan hệ giữa các dữ liệu đó.

25. Mức view trong kiến trúc 3 mức trừu tượng của 1 hệ CSDL?

- Mức ngoài/ view: Mô tả một phần của CSDL cho một nhóm người dùng quan tâm và dấu đi phần còn lại của CSDL khỏi nhóm người dùng đó.

26. Phân loại Tính độc lập dữ liệu ?

- + Kiến trúc ba mức có thể được dùng để giải thích khái niệm độc lập dữ liệu.
- + Tính độc lập dữ liệu có thể định nghĩa như khả năng thay đổi lược đồ tại một mức của một hệ CSDL mà không phải thay đổi lược đồ tại mức cao hơn kế tiếp.
- + Có hai tính độc lập dữ liệu: Tính độc lập vật lý và tính độc lập logic.

27. Tính độc lập vật lý?

- Tính độc lập vật lý: là khả năng thay đổi lược đồ vật lý mà không phải thay đổi lược đồ quan niệm. Vì vậy lược đồ ngoài cũng không cần phải thay đổi.

Ví dụ: từ dữ liệu Sinhvien(MaSV: string, Hoten: char(20), tenKhoa: char(20)) ta thêm chỉ mục Hoten để tìm kiếm theo họ tên hiệu quả hơn sẽ làm thay đổi cấu trúc lưu trữ nhưng không phải thay đổi lược đồ quan niệm.

28. Tính độc lập logic là gì?

- Tính độc lập logic: là khả năng thay đổi lược đồ quan niệm mà không phải thay đổi lược đồ ngoài hay chương trình ứng dụng.

Ví dụ: Với lược đồ khái niệm về Sinh viên ta có thể mở rộng CSDL bằng cách thêm dữ liệu về DiemTB: float sẽ không làm thay đổi lược đồ ngoài hay chương trình ứng dụng đang có.

Assignment:

1. Why would you choose a database system instead of simply storing data in operating system files? When would it make sense *not* to use a database system?

A database is an integrated collection of data, usually so large that it has to be stored on secondary storage devices such as disks or tapes. This data can be maintained as a collection of operating system files, or stored in a DBMS (database management system). The advantages of using a DBMS are:

- Data independence and efficient access. Database application programs are independent of the details of data representation and storage. The conceptual and external schemas provide independence from physical storage decisions and logical design decisions respectively. In addition, a DBMS provides efficient storage and retrieval mechanisms, including support for very large files, index structures and query optimization.
- Reduced application development time. Since the DBMS provides several important functions required by applications, such as concurrency control and crash recovery, high level query facilities, etc., only application-specific code needs to be written. Even this is facilitated by suites of application development tools available from vendors for many database management systems.
- Data integrity and security. The view mechanism and the authorization facilities of a DBMS provide a powerful access control mechanism. Further, updates to the data that violate the semantics of the data can be detected and rejected by the DBMS if users specify the appropriate integrity constraints.
- Data administration. By providing a common umbrella for a large collection of data that is shared by several users, a DBMS facilitates maintenance and data administration tasks. A good DBA can effectively shield end-users from the chores of fine-tuning the data representation, periodic back-ups etc.
- Concurrent access and crash recovery. A DBMS supports the notion of a transaction, which is conceptually a single user's sequential program. Users can write transactions as if their programs were running in isolation against the database. The DBMS executes the actions of transactions in an interleaved fashion

to obtain good performance, but schedules them in such a way as to ensure that conflicting operations are not permitted to proceed concurrently. Further, the DBMS maintains a continuous log of the changes to the data, and if there is a system crash, it can restore the database to a transaction-consistent state. That is, the actions of incomplete transactions are undone, so that the database state reflects only the actions of completed transactions. Thus, if each complete transaction, executing alone, maintains the consistency criteria, then the database state after recovery from a crash is consistent.

If these advantages are not important for the application at hand, using a collection of files may be a better solution because of the increased cost and overhead of purchasing and maintaining a DBMS.

2. What is logical data independence and why is it important?

Logical data independence means that users are shielded from changes in the logical structure of the data, i.e., changes in the choice of relations to be stored. For example, if a relation `Students(sid, sname, gpa)` is replaced by `Studentnames(sid, sname)` and `Studentgpas(sid, gpa)` for some reason, application programs that operate on the `Students` relation can be shielded from this change by defining a view `Students(sid, sname, gpa)` (as the natural join of `Studentnames` and `Studentgpas`). Thus, application programs that refer to `Students` need not be changed when the relation `Students` is replaced by the other two relations. The only change is that instead of storing `Students` tuples, these tuples are computed as needed by using the view definition; this is transparent to the application program.

3. Explain the difference between logical and physical data independence.

Logical data independence means that users are shielded from changes in the logical structure of the data, while physical data independence insulates users from changes in the physical storage of the data. We saw an example of logical data independence in the answer to Exercise 1.2. Consider the `Students` relation from that example (and now assume that it is not replaced by the two smaller relations). We could choose to store `Students` tuples in a heap file, with a clustered index on the `sname` field. Alternatively, we could choose to store it with an index on the `gpa` field, or to create indexes on both fields, or to store it as a file sorted by `gpa`. These storage alternatives are not visible to users, except in terms of improved performance, since they simply see a relation as a set of tuples. This is what is meant by physical data independence.

4. What are the responsibilities of a DBA? If we assume that the DBA is never interested in running his or her own queries, does the DBA still need to understand query optimization? Why?

The DBA is responsible for:

- Designing the logical and physical schemas, as well as widely-used portions of the external schema.
- Security and authorization.
- Data availability and recovery from failures.
- Database tuning: The DBA is responsible for evolving the database, in particular the conceptual and physical schemas, to ensure adequate performance as user requirements change.

A DBA needs to understand query optimization even if s/he is not interested in running his or her own queries because some of these responsibilities (database design and tuning) are related to query optimization. Unless the DBA understands the performance needs of widely used queries, and how the DBMS will optimize and execute these queries, good design and tuning decisions cannot be made.

5. Scrooge McNugget wants to store information (names, addresses, descriptions of embarrassing moments, etc.) about the many ducks on his payroll. Not surprisingly, the volume of data compels him to buy a database system. To save money, he wants to buy one with the fewest possible features, and he plans to run it as a stand-alone application on his PC clone. Of course, Scrooge does not plan to share his list with anyone. Indicate which of the following DBMS features Scrooge should pay for; in each case, also indicate why Scrooge should (or should not) pay for that feature in the system he buys.

Let us discuss the individual features in detail.

- A security facility is necessary because Scrooge does not plan to share his list with anyone else. Even though he is running it on his stand-alone PC, a rival duckster could break in and attempt to query his database. The database's security features would foil the intruder.
- Concurrency control is not needed because only he uses the database.
- Crash recovery is essential for any database; Scrooge would not want to lose his data if the power was interrupted while he was using the system.

- A view mechanism is needed. Scrooge could use this to develop “custom screens” that he could conveniently bring up without writing long queries repeatedly.
- A query language is necessary since Scrooge must be able to analyze the dark secrets of his victims. In particular, the query language is also used to define views.

6. Describe the structure of a DBMS. If your operating system is upgraded to support some new functions on OS files (e.g., the ability to force some sequence of bytes to disk), which layer(s) of the DBMS would you have to rewrite to take advantage of these new functions?

The architecture of a relational DBMS typically consists of a layer that manages space on disk, a layer that manages available main memory and brings disk pages into memory as needed, a layer that supports the abstractions of files and index structures, a layer that implements relational operators, and a layer that parses and optimizes queries and produces an execution plan in terms of relational operators. In addition, there is support for concurrency control and recovery, which interacts with the buffer management and access method layers.

The disk space management layer has to be rewritten to take advantage of the new functions on OS files. It is likely that the buffer management layer will also be affected.