**CÂU HỎI AUDIT MODULE 3**

**DATABASE**

STT Nội dung câu hỏi

1. Cơ sở dữ liệu (database) là gì?

-Là nơi lưu trữ và thao tác các dữ liệu của 1 trang web.

2 Phân biệt các câu lệnh DML & DDL? DML, DDL bao gồm các câu lệnh nào?

### -DDL (Data Definition Language) :Ngôn ngữ định nghĩa dữ liệu.

* **Lệnh CREATE**: Tạo một bảng, một View của bảng, hoặc đối tượng khác trong Database.
* **Lệnh ALTER**: Sửa đổi một đối tượng Database đang tồn tại, ví dụ như một bảng.
* **Lệnh DROP**: Xóa toàn bộ một bảng, một View của bảng hoặc đối tượng khác trong một Database.

-DML(Data Manipulation Language): Ngôn ngữ thao tác dữ liệu.

* **Lệnh SELECT**: Lấy các bản ghi cụ thể từ một hoặc nhiều bảng.
* **Lệnh INSERT**: Tạo một bản ghi.
* **Lệnh UPDATE**: Sửa đổi các bản ghi.
* **Lệnh DELETE**: Xóa các bản ghi.

-DCL( Data Control Language): Ngôn ngữ điều khiển dữ liệu.

* **Lệnh GRANT**: Trao một quyền tới người dùng.
* **Lệnh REVOKE**: Thu hồi quyền đã trao cho người dùng.

-TCL(Transaction Control Language): Ngôn ngữ điều khiển giao dịch.

***COMMIT***: Lệnh commit được sử dụng để lưu vĩnh viễn bất kỳ giao dịch nào vào cơ sở dữ liệu.

**ROLLBACK**: Lệnh này khôi phục cơ sở dữ liệu về trạng thái đã cam kết cuối cùng, nó cũng được sử dụng với lệnh savepoint để chuyển đến một savepoin trong một giao dịch.

**SAVEPOINT**: Lệnh Savepoint được sử dụng để lưu tạm thời một giao dịch để bạn có thể quay lại thời điểm đó bất cứ khi nào cần thiết.

3 Trình bày về mô hình thực thể mối quan hệ (ERD - Entity-Relationship Diagram)? Trong mô hình thực thể mối quan hệ có những thành phần nào?

ERD - Entity-Relationship Diagram: (sơ đồ quan hệ thực thể) thể hiện các thực thể có trong database và mqh giữa chúng vs nhau.

1.Từ yêu cầu suy ra các thực thể

2.XĐ cá thuộc tính cho mỗi thực thể.

3.XĐ mqh 1-1,1-n,nn.

4 Các bước chuyển đổi từ ERD -> Table?

5 bước.

1.Biến đổi thực thể đến table.

-Thuộc tính khóa của thực thể chính là khóa chính của table.

-Các thuộc tính thông thường của thực thể là thuộc tính của table.

2.Biến đổi thuộc tính đa trị.

-Tạo ra table mới và xđ mqh giữa đa trị và table là 1-n hay n-n.

3.Biến đổi mqh 1-1.

-Đặt thuộc tính khóa ở phía tùy chọn sang phía bắt buộc.

4.Biến đổi mqh 1-n.

-Đặt thuộc tính khóa ở phía 1 sang n.

5.Biến đổi mqh n-n.

-Sinh ra 1 table mới có khóa chính là tập hợp của 2 khóa chính của 2 thực thể tham gia và mqh 1 khóa phức hợp.

5 Các bước chuẩn hóa dữ liệu?

Normalization from đây là bước review cuối cùng sau khi đã chuyển sang table.

1NF ->2NF->3NF->BCNF/boyce codd

6 Ràng buộc (constraint) là gì? Các loại ràng buộc trong MySQL?

- **Ràng buộc (Constraint)** là các quy tắc được áp dụng trên các cột dữ liệu của một bảng. Chúng được sử dụng để kiểm tra tính hợp lệ của dữ liệu đầu vào, đảm bảo tính chính xác, độ tin cậy và tính toàn vẹn của dữ liệu trong database.

- NOT NULL: Đảm bảo dữ liệu của cột không được nhận giá trị NULL.

- DEFAULT: Cung cấp một giá trị mặc định cho một cột trong trường hợp dữ liệu của cột không được nhập vào hay không được xác định.

-: UNIQUE: Đảm bảo rằng dữ liệu của cột là duy nhất, tất cả các giá trị trong một cột là khác nhau không trùng lặp.

- PRIMARY Key: Dùng để thiết lập khóa chính trên bảng, giá trị của cột làm khóa chính phải là duy nhất, không được trùng lặp. Việc khai báo ràng buộc khóa chính yêu cầu các cột phải NOT NULL.

- FOREIGN Key: Dùng để thiết lập khóa ngoại trên bảng, tham chiếu đến bảng khác thông qua giá trị của cột được liên kết. Giá trị của cột được liên kết phải là duy nhất trong bảng kia.

- CHECK: Đảm bảo tất cả các giá trị trong một cột thỏa mãn một số điều kiện.

- INDEX: Dùng để tạo và lấy dữ liệu từ cơ sở dữ liệu một cách nhanh chóng.

7 Khóa chính (Primary key) là gì? “Khóa chính chứa một hoặc nhiều thuộc tính khóa” nghĩa là gì?

-Khóa chính để định danh duy nhất mỗi bản ghi trong bảng của database

-Thiết lập mqh 1-n giữa 2 table database

-Dữ liệu có tính duy nhất, không chứa giá trị null

-Mỗi bảng chỉ có 1 Primary key

-Primary key có thể tạo ra dựa trên một cột hay nhiều cột của table.

\*Nghĩa là khóa đơn hoặc khóa phức hợp

Khóa đơn là 1 trường làm khóa chính

Khóa phức hợp nhiều trường làm khóa chính

8 Khóa ngoại (foreign key) là gì?

-Là cơ chế để tạo liên kết giữa các bảng trong cùng một cơ sở dữ liệu.

-Được đặt trên một cột của bảng này và tham chiếu đến khóa chính của 1 bảng khác

-Kiêu dữ liệu của khóa ngoại và khóa chính mà nó tham chiếu phải giống nhau.Thông thường thì các bảng đều sử dụng kiểu interger cho

Khóa chính.

9 Phân biệt DELETE và TRUNCATE?

-Lệnh **DELETE** chỉ xóa các hàng từ bảng dựa trên điều kiện đã được định nghĩa bởi mệnh đề WHERE hoặc xóa tất cả các hàng từ bảng nếu điều kiện này không được xác định. Nhưng nó **không** giải phóng bộ nhớ chứa bảng đó. Trong khi đó, lệnh TRUNCATE xóa tất cả các hàng từ bảng và **giải phóng bộ nhớ** chứa bảng đó

Không Reset id

* TRUNCATE không thể chạy được khi bảng bạn định xóa có foreign\_key (nếu bảng đó có foreign\_key trỏ đến chính nó thì bạn có thể xóa bình thường)
* Dữ liệu bị xóa bởi DELETE có thể restored được còn TRUNCATE thì không.
* Reset id về 1
* Tùy vào từng trường hợp thì sẽ dùng DELETE hoặc TRUNCATE, TRUNCATE sẽ luôn nhanh hơn vì nó tiêu tốn ít bộ nhớ hơn, các bạn có thể cân nhắc điều này khi cần xóa 1 bảng lớn vs nhiều record.

10 Các loại JOIN? Phân biệt?

-INNER JOIN (Hoặc JOIN):Lấy phần giao 2 bảng.

-LEFT OUTER JOIN (Hoặc LEFT JOIN): Trả lại tất cả các dòng từ bảng bên trái, và các dòng đúng với điều kiện từ bảng bên phải

-RIGHT OUTER JOIN (Hoặc RIGHT JOIN): Trả lại tất cả các hàng từ bảng bên phải, và các dòng thỏa mãn điều kiện từ bảng bên trái

-FULL OUTER JOIN khóa trả về tất cả các bản ghi khi có sự trùng khớp trong các bản ghi bảng bên trái (table1) hoặc bên phải (table2).

11 Phân biệt JOIN và UNION?

- JOIN trong SQL được sử dụng để kết hợp dữ liệu từ nhiều bảng dựa trên một điều kiện phù hợp giữa chúng. Dữ liệu được kết hợp bằng cách sử dụng câu lệnh JOIN sẽ dẫn đến các cột mới.

- UNION trong SQL được sử dụng để kết hợp tập hợp kết quả của hai hoặc nhiều câu lệnh SELECT. Dữ liệu được kết hợp bằng cách sử dụng câu lệnh UNION thành các kết quả thành các hàng riêng biệt mới.

|  |  |
| --- | --- |
| join | union |
| JOIN kết hợp dữ liệu từ nhiều bảng dựa trên một điều kiện phù hợp giữa chúng. | SQL kết hợp tập hợp kết quả của hai hoặc nhiều câu lệnh SELECT. |
| Nó kết hợp dữ liệu thành các cột mới. | Nó kết hợp dữ liệu thành các hàng mới |
| Số cột được chọn từ mỗi bảng có thể không giống nhau. | Số cột được chọn từ mỗi bảng phải giống nhau. |
| Kiểu dữ liệu của các cột tương ứng được chọn từ mỗi bảng có thể khác nhau. | Kiểu dữ liệu của các cột tương ứng được chọn từ mỗi bảng phải giống nhau. |
| Nó có thể không trả về các cột riêng biệt. | Nó trả về các hàng riêng biệt. |

12 Phân biệt UNION và UNION ALL?

* **UNION:** chỉ giữ các bản ghi *duy nhất*
* **UNION ALL:** lưu giữ tất cả các bản ghi, kể cả các *bản sao*
* Phải có cùng số cột được truy xuất trong mỗi câu lệnh SELECT để được kết hợp.
* Các cột được truy xuất phải theo cùng một thứ tự trong mỗi câu lệnh SELECT.
* Các cột được truy xuất phải có kiểu dữ liệu tương tự.

13 Phân biệt EXIST và IN?

Toán tử EXISTS được sử dụng để kiểm tra sự tồn tại của bất kỳ bản ghi nào trong một truy vấn con.

Toán tử IN cho phép bạn chỉ định nhiều giá trị trong một WHEREmệnh đề.

Toán tử INlà cách viết tắt của nhiều ORđiều kiện.

|  |  |
| --- | --- |
| EXISTS | IN |
| Nó được sử dụng để kiểm tra sự tồn tại của dữ liệu trong một truy vấn con. Nói cách khác, nó xác định liệu giá trị có được trả về hay không. | Nó được sử dụng để giảm thiểu nhiều điều kiện OR. |
| Nó không so sánh các giá trị giữa truy vấn con và truy vấn mẹ. | Nó so sánh các giá trị giữa truy vấn con (truy vấn con) và truy vấn mẹ |
| Nó dừng để thực hiện thêm khi đáp ứng điều kiện tích cực duy nhất. | Nó quét tất cả các giá trị bên trong khối IN. |
| Nó trả về TRUE hoặc FALSE. Do đó, chúng tôi không thể sử dụng nó để so sánh các giá trị NULL. | Nó có thể trả về TRUE, FALSE hoặc NULL. Do đó, chúng ta có thể sử dụng nó để so sánh các giá trị NULL. |
| Chúng ta chỉ có thể sử dụng nó trên các truy vấn con. | Chúng ta có thể sử dụng nó trên các truy vấn con cũng như với các giá trị. |
| Nó thực thi nhanh hơn khi kết quả truy vấn con lớn. Nó hiệu quả hơn IN vì nó xử lý các giá trị Boolean hơn là các giá trị của chính nó. | Nó thực thi nhanh hơn khi kết quả truy vấn con ít hơn. |

14 Phân biệt AND và OR?

Toán tử **AND** cho phép sự tồn tại của nhiều điều kiện trong mệnh đề WHERE của một câu lệnh SQL.

Toán tử OR được sử dụng để kết hợp nhiều điều kiện trong mệnh đề WHERE của câu lệnh SQL.

15 Phân biệt GROUP BY và ORDER BY?

- Mệnh [đề GROUP BY](https://www.javatpoint.com/sql-group-by) được sử dụng trong các truy vấn SQL để tổ chức dữ liệu có các giá trị thuộc tính giống nhau. Thông thường, chúng tôi sử dụng nó với [câu lệnh SELECT](https://www.javatpoint.com/sql-select) . Luôn nhớ rằng chúng ta phải đặt mệnh đề GROUP BY sau [mệnh đề WHERE](https://www.javatpoint.com/sql-where) . Ngoài ra, nó có nhịp độ trước mệnh đề ORDER BY.

- Mệnh [đề ORDER BY](https://www.javatpoint.com/sql-order-by) được sử dụng trong các truy vấn SQL để sắp xếp dữ liệu được trả về bởi một truy vấn theo thứ tự tăng dần hoặc giảm dần. Nếu chúng ta bỏ qua thứ tự sắp xếp, nó sẽ sắp xếp kết quả tóm tắt theo thứ tự tăng dần theo mặc định. Mệnh đề ORDER BY, giống như mệnh đề GROUP BY, có thể được sử dụng cùng với câu lệnh SELECT. **ASC** biểu thị thứ tự tăng dần, trong khi **DESC** biểu thị thứ tự giảm dần.

16 Muốn nối chuỗi trong MySQL thì sử dụng gì?

-Nối chuỗi trong MySQL sử dụng hàm concat().

17 Yêu cầu tính điểm trung bình của tất cả các sinh viên thì làm thế nào?

18 Yêu cầu tính điểm trung bình của từng bạn sinh viên thì làm thế nào?

19 Yêu cầu hiển thị tên sinh viên và điểm trung bình tương ứng, chỉ hiển thị các bạn có điểm trung bình lớn hơn 5 thì làm thế nào?

20 Hiển thị danh sách sinh viên và lớp học tương ứng thì làm thế nào (hai bảng Student và Class có cột chung ClassID)?

21 Hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm thi?

22 Hiển thị danh sách sinh viên theo thứ tự giảm dần của họ tên, nếu họ tên trùng nhau thì sắp xếp theo tăng dần của tuổi?

23 Hiển thị các sinh viên có họ là ‘Le’ và tên là ‘Hai’?

24 Hiển thị các sinh viên có tên bắt đầu là ‘C’?

25 Làm sao để có thể tăng tốc độ truy vấn? Nhược điểm là gì?

- Index giúp tăng tốc các truy vấn SELECT chứa các mệnh đề WHERE hoặc ORDER, nhưng nó làm chậm việc dữ liệu nhập vào với các lệnh UPDATE và INSERT, delete.

-Chiếm dụng bộ nhớ

26 Các dạng tham số ở SP?

-IN: tương đương với tham trị (KNT)

OUT: mục đích lấy giá trị từ SP

INOUT: tương đương với cơ chế truyền tham chiếu

27 Phân biệt SP, Trigger, Function?

-Đều là tập hợp các câu lệnh để thực thi 1 nv nào đó

-SP không có kiểu trả về sử dụng lệnh column có thể lấy 1 hoặc nhieu giá trị có thể gọi Funcition

-Trigger: không có tham số và k có kết quả trả về tự động thực thi 1 nv nào đó.

-Funcition: có trả về chỉ trả về 1 giá trị duy nhất không thể goi SP.