Báo cáo MySql

1. Sql.
   1. Khái niệm

* Sql ( Structured Query Language) là ngôn ngữ truy vấn dữ liệu có cấu trúc và nó là ngôn ngữ máy tính nhằm thao tác với hệ cơ sở dữ liệu. Nó được sử dụng để quản lí hệ cơ sở dữ liệu trong hệ quản trị cơ sở dữ liệu quan hệ mà các dữ liệu được tổ chức thành các bảng và có mối quan hệ giữa các bảng với nhau. Câu lệnh SQL được sử dụng để truy xuất và cập nhật dữ liệu trong cơ sở dữ liệu.
* Hầu hết các hết các hệ thống cơ sở dữ liệu sử dụng SQL đều có phần mở rộng riêng của họ bổ sung độc quyền mà thường chỉ sử dụng trên hệ thóng của họ. Tuy nhiên các câu lệnh tiêu chuẩn như “SELECT”, “INSERT”, “UPDATE”, “DELETE”, “CREATE”, “DROP” gần như giống nhau và có thể sử dụng để thực hiện hầu hết tất cả mọi thứ trong nhu cầu sử dụng một cở sở dữ liệu.
  1. RDBMS ( Hệ quản trị cơ sở dữ liệu)
* RDBMS viết tắt của Relational Database Management System ( hệ quản trị cơ sở dữ liệu ).
* RDBMS là cơ sở cho sql và cho tất cả hệ thống cơ sở dữ liệu hiện tại như Mysql, Sql Server,…
* Dữ liệu trong RDBMS được lưu trong cơ sở dữ liệu đối tượng được gọi là các bảng. Bảng là tập hợp các mục dữ liệu có liên quan có chung các thuộc tính.
* Mỗi bảng được chia thành các thực thể nhỏ hơn gọi là các trường. Trường là một cột trong bảng được thiết kế để duy trì thông tin cụ thể về mọi bản ghi trong bảng,
* Mỗi bản ghi hay mỗi hàng là một mục riêng lẻ tồn tại trong bảng.
* Một cột là một thực thể chứa tất cả các thông tin liên kết đến một trường cụ thể trong bảng.
  1. Đặc điểm.
* Sql đơn giản để học
* Sql được sử dụng để truy cập dữ liệu từ hệ quản trị cơ sở dữ liệu quan hệ.
* Sql được sử dụng để thực hiện truy vấn với cơ sở dữ liệu.
* Sql được sử dụng để mô tả dữ liệu
* Sql được sử dụng để xác định dữ liệu trong cơ sở dữ liệu và thao tác nó khi cần thiết.
* Sql được sử dụng để tạo và xóa các cơ sở dữ liệu, các bảng.
* Sql cho phép người dùng có thể đặt quyền trên bảng, view, …

1. NoSql
   1. Khái niệm

* Dữ liệu trong NoSql không phải được chia thành các bảng.
* Mô hình dữ liệu NoSql cho phép dữ liệu được lồng nhau trong một cấu trúc dữ liệu duy nhất
* Cơ sở dữ liệu NoSql sử dụng nhiều mô hình dữ liệu để truy cập va quản lí dữ liệu. Các loại cơ sở dữ liệu này được tối ưu hóa dàng riêng cho các ứng dụng yêu cầu mô hình dữ liệu linh hoạt có lượng dữ liệu lớn và độ trễ thấp, có thể đạt được bằng cách giảm bớt một số hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác
* Cơ sở dữ liệu NoSql đang được sử dụng ngày càng nhiều trong các ứng dụng có dữ liệu lớn và ứng dụng nền web real-time. Hướng tiếp cận bao gồm: tính đơn giản trong thiết kế, mở rộng theo chiều ngang cho các cụm máy chủ đơn giản hơn, kiểm soát tính khả dụng tốt hơn. Cấu trúc dữ liệu được thiết kế cho các cơ sở dữ liệu NoSql khác với cấu trúc dữ liệu được sử dụng mặc định trong các cơ sở dữ liệu quan hệ, khiến nó thao tác nhanh hơn trong NoSql. Cơ sở dữ liệu NoSql thích hợp cho từng trường hợp cụ thể cho vấn đề mà nó giải quyết. Đôi khi cấu trúc dữ liệu được thiết kế dưới dạng NoSql được xem là linh hoạt hơn các bảng cơ sở dữ liệu kiểu quan hệ. Nhiều kho lưu trữ NoSql duy sinh tính nhất quán để ưu tiên tính sẵn có, dung lượng của phân vùng và tốc độ. Rào cản đối với việc áp dụng các cơ sở dữ liệu NoSql bao gồm việc sử dụng các ngôn ngữ truy vấn mức thấp ( thay vì sử dụng Sql).
* Hầu hết các cơ sở dữ liệu Nosql đưa ra một khái niệm chung trong đó các thay đổi
  1. Đặc điểm
* Có khả năng mở rộng cao
* Hỗ trợ giản đồ linh hoạt
* Có thể xử lí cả dữ liệu phi cấu trúc và bán cấu trúc.
* Không có mối quan hệ quá phức tạp
* Có hiệu suất cao
  1. Phân loại
* Lưu trữ Key – value: Dữ liệu được biểu diễn như một mảng các cặp key – value, như vậy mỗi key chỉ có thể xuất hiện 1 lần trong mảng với 1 giá trị. Cơ sở dữ liệu key – value sử dụng trong trường hợp cần lưu trữ một lượng lớn dữ liệu mà không cần phải thực hiện cách truy vấn phức tạp để lấy nó. Ví dụ: Redis, DynanoDB,..
* Lưu trữ Document: Lưu trữ dữ liệu tương tự như đối tượng JSON. Mỗi tài liệu chứa các cặp trường – giá trị. Các giá trị thường có thể là một loạt các loại bao gồm những thứ string, số, array, các đối tượng và các cấu trúc của nó thường phù hợp với các đối tượng trong code. Bởi vì sự đa dạng của các loại giá trị trường và ngôn ngữ truy vấn mạnh mẽ, lưu trữ Document sử dụng trong nhiều trường hợp. Nó có thể mở rộng quy mô ngang để chứa các khối dữ liệu lớn. Ví dụ: MongoDB,..
* Lưu trữ cột: Lưu trữ dữ liệu trong các bảng, hàng, cột động. Nó cung cấp nhiều tính linh hoạt trên cơ sở dữ liệu vì mỗi hàng không cần thiết có cùng một cột. Nó được sử dụng khi lưu trữ dữ liệu lớn và có thể dự đoán những gì mô hình truy vấn dữ liệu có thể có. Lưu trữ cột thường sử dụng để lưu trữ dữ liệu Internet of thing và hồ sơ người dùng. Ví dụ: Cassandra, Hbase,…
* Lưu trữ dạng đồ thị: Dữ liệu lưu trữ các nút và các cạnh. Các nút thường lưu trữ thông tin về con người, địa điểm, sự vật trong khi các cạnh lưu trữ thông tin về mối quan hệ giữa các nút. Lưu trữ dạng đồ thị thường được sử dụng khi cần phải truy vấn qua các mối quan hệ để tìm kiếm như mạng xã hội, phát hiện gian lận hay các công cụ khuyến nghị. Ví dụ: Neo4j, JanusGraph,…

1. Join

* Join được sử dụng để kết hợp các hàng từ hai hay nhiều bảng dựa trên các cột liên quan giữa chúng
* Có 4 loại Join:

+ Inner Join: Trả về các bản ghi có giá trị phù hợp trong 2 bảng

+ Left (Outer) Join: Trả về tất cả các bản ghi từ bảng bên trái và các bản ghi phù hợp ở bên phải.

+ Right (outer) Join: Trả về tất cả các bản ghi từ bảng bên phải và các bản ghi phù hợp ở bên trái.

+ Full (outer) Join: trả về tất cả bản ghi khi nó ở bảng trái hoặc bản phải

1. Function

* Function là một đối tượng trong cơ sở dữ liệu bao gồm nhiều câu lệnh được nhóm lại với nhau và được tạo ra với mục đích sử dụng lại.
* Function luôn trả về giá trị.
* Function gồm 2 loại:

+ Function hệ thống.

+ Function tự định nghĩa: Trả về giá trị vô hướng của các kiểu dữ liệu hoặc trả về bảng.

1. Procedure

* Procedure là một mã SQL chuẩn bị được lưu, do đó nó có thể được tái sử dụng.
* Nếu có một truy vấn SQL được viết lại nhiều lần, có thể lưu nó lại thành 1 procedure và sau đó chỉ cần gọi nó để thực hiện nó
* Procedure cũng có thể được truyền các tham số bên ngoài để procedure có thể hoạt động dựa trên các giá trị tham số truyền vào.
* Sử dụng EXCUTE để thực thi các stored procedure.
* Ưu điểm:

+ Nhanh: Nó có khả năng phân tích cú pháp và tối ưu hóa trong lần thực thi đầu tiên và một phiên bản dịch của chúng sẽ được lưu trong bộ nhớ để sử dụng trong lần sau.

+ Giảm thiểu quá tải: Với trường hợp sử dụng hàng trăm câu lệnh đơn được đồng thời gửi đi dẫn đến tình trạng quá tải. Stored procedure cải thiện vấn đề này bằng cách gửi theo trình tự xử lí.

+ Bảo mật: Phân quyền sử dụng cho các user, cấp quyền, giới hạn quyền cho các user thậm chí không cho phép họ được phép thực thi trực tiếp những stored procedure. Khi đó sẽ hạn chế các vấn đề xâm phạm dữ liệu trái phép.

1. Transaction

* Transaction là một tiến trình thực hiện một nhóm câu lệnh Sql, các câu lệnh này được thực thi một cách tuần tự và độc lập.
* Một transaction được thực hiện thành công khi tất cả câu lệnh đều thành công, khi đó tất cả các thay đổi dữ liệu được thực hiện trong trong transaction được lưu vào cơ sở dữ liệu.
* Nếu chỉ một câu lệnh trong transaction thất bại thì toàn bộ tiến trình sẽ thất bại, đồng nghĩa với việc dữ liệu phải được rollback về trạng thái ban đầu.
* Có 4 thuộc tính của transaction là:

+ Tính bảo toàn: đảm bảo tất cả câu lệnh trong nhóm thực thi thành công. Nếu không, transaction bị hủy bỏ tại thời điểm thất bại và tất cả những thao tác được khôi phục về ban đầu đồng nghĩ với việc không có sự thay đổi về dữ liệu trong cơ sở dữ liệu

+ Tính nhất quán: đảm bảo cơ sở dữ liệu thay đổi chính xác trạng thái khi một transaction được thực thi thành công

+ Tính độc lập: Các transaction hoạt động độc lập và minh bạch với nhau

* Các lệnh để điều khiển transaction:

+ Commit: để lưu thay đổi

+ Rollback: để quay lại trạng thái ban đầu trước khi thay đổi.

+ SavePoint: Tạo các điểm (point) trong các transaction để rollback. Có thể rollback về các điểm point.

+ Set transaction: đặt tên cho một transaction.

1. Index

* Index là bảng tra cứu đặc biệt mà cơ sở dữ liệu có thể sử dụng để giảm thời gian và hiệu xuất khi truy vấn.
* Index giúp tăng tốc độ các truy vấn SELECT chứa các mệnh đề WHERE và ORDER, nhưng nó làm chậm việc dữ liệu nhập vào với các câu lệnh UPDATE và INSERT.
* Các index có thể được tạo và xóa mà không ảnh hưởng đến dữ liệu
* Có 4 kiểu index:

+ Single-column index: tạo cho duy nhất 1 cột trong bảng

+ Unique index: tạo index duy nhất, được sử dụng để tăng hiệu năng và đảm bảo tính toàn vẹn dữ liệu. Một unique index không cho phép chèn giá trị trùng lặp vào bảng

+ Composite index: tạo index kết hợp dành cho hai hoặc nhiều cột trong một bảng

+ Implicit index: là index được tạo tự động bởi hệ quản trị cơ sở dữ liệu khi một bảng được tạo. Các implicit index được tạo tự động cho các ràng buộc Primary key và các ràng buộc Unique.