Thiết kế CSDL:

1. Thu thập yêu cầu về lưu trữ cơ sở dữ liệu
2. ERD - Entity Relationship Diagram
3. RDM – Relation Data Model
4. Thiết kế CSDL vật lý
5. Cài đặt CSDL

Table\_Duan

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Note |
| MaDuan | Char(10) | Primary key | Mã dự án |
| TenDuan | nVarchar(100) | Not null | Tên dự án |
| KinhPhi | float | >=0 | Kinh phí |

Table\_NhanVien

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Note |
| MaNV | Char(10) | Primary key | Mã nhân viên |
| HoLot | nVarchar(50) | Not null | Họ lót |
| TenNV | nVarchar(50) | Not null | Tên nhân viên |
| SDT | Char(10) | >=0 | Số điện thoại |
| MaPhong | Char(10) | foreign key | Mã phòng |

Table\_NV\_DA

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Note |
| MaNV | Char(10) | foreign key | Mã nhân viên |
| MaDuan | Char(10) | foreign key | Mã dự án |
| Sogio | float | >=0 | Số giờ |

create database qlnv\_280524;

use qlnv\_280524;

create table phong

( MaPhong char(10),

TenPhong nvarchar(50) not null,

Diadiem nvarchar(100),

SoNV int,

constraint pk\_MaPhong primary key(MaPhong),

constraint ch\_SoNV check(SoNV >= 0)

);

create table duan

( MaDuan char(10),

TenDuan nvarchar(100) not null,

KinhPhi float,

constraint pk\_MaDuan primary key(MaDuan),

constraint ch\_KinhPhi check(KinhPhi >= 0)

);

create table nhanvien

( MaNV char(10),

HoLot nvarchar(50) not null,

TenNV nvarchar(50) not null,

SDT varchar(10),

MaPhong char(10),

constraint pk\_MaNV primary key(MaNV),

constraint fk\_MaPhong foreign key(MaPhong) references phong(MaPhong)

);

create table NV\_DA

( MaNV char(10),

MaDuan char(10),

Sogio float,

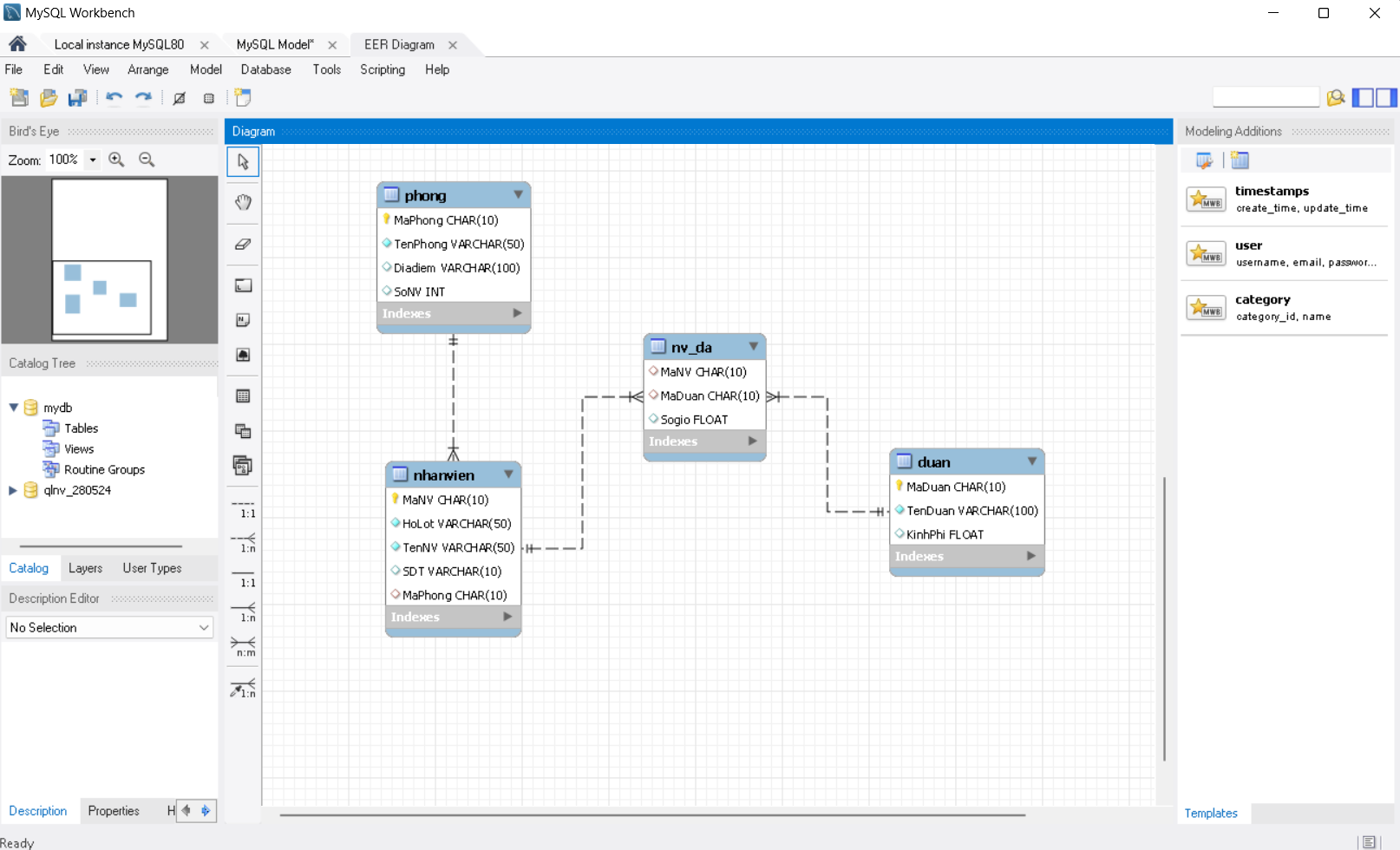
constraint fk\_MaNV foreign key(MaNV) references nhanvien(MaNV),

constraint fk\_MaDuan foreign key(MaDuan) references duan(MaDuan),

constraint ch\_Sogio check(Sogio >= 0)

);

Diagram



INSERT INTO phong

VALUES ('P1', N'PHÒNG TỔ CHỨC', N'ĐÀ NẴNG', 0);

INSERT INTO phong

VALUES ('P2', N'PHÒNG IT', N'ĐÀ NẴNG', 0);

INSERT INTO phong

VALUES ('P3', N'PHÒNG KẾ TOÁN', N'ĐÀ NẴNG', 0);

SELECT \* FROM PHONG;

INSERT INTO nhanvien

VALUES ('NV01', N'Nguyễn Văn', N'A', N'0987324561', N'P1'),

('NV02', N'Nguyễn Văn', N'B', N'0987324561', N'P1'),

('NV03', N'Phạm Thị', N'C', N'0987324561', N'P3'),

('NV04', N'Nguyễn Anh', N'D', N'0987324561', N'P2'),

('NV05', N'Nguyễn Thị', N'E', N'0987324561', N'P2');

SELECT \* FROM nhanvien;

INSERT INTO duan

VALUES ('DA01', N'Đầu tư chứng khoáng', 10000),

('DA02', N'Phát triển website bán hàng tự động', 150000),

('DA03', N'Phát triển trí tuệ nhân tạo có cảm súc', 300000);

SELECT \* FROM duan;

INSERT INTO NV\_DA

VALUES ('NV01', N'DA01', 10000),

('NV02', N'DA01', 150000),

('NV01', N'DA03', 300000);

II. Sửa cấu trúc db

Alter table <tên bảng>

Add <Tên trường> <kiểu dữ liệu>

Drop column <Tên trường>

Alter column <Tên trường> <Kiểu dữ liệu>

Add/ Drop constraint

/\* Thêm trường địa chỉ vào bảng nhân viên \*/

alter table nhanvien

add diachi nvarchar(100);

/\*2. Xóa trường địa chỉ của bảng nhân viên \*/

alter table nhanvien

drop column diachi;

/\*3. Thêm 1 trường giới tính và cso ràng buộc miền giá trị "Nam, Nữ, Khác" \*/

Alter table nhanvien

add gioitinh nvarchar(5);

/\*4. Thêm ràng buộc \*/

alter table nhanvien;

Homework 28-05-24

Table\_CUSTOMER

|  |  |  |  |
| --- | --- | --- | --- |
| Field\_Name | Datatype | Constraints | Note |
| CUSTOMER\_ID | Varchar(10) | Primary key | ID KHÁCH HÀNG |
| CUSTOMER\_NAME | nVarchar(50) | Not null | TÊN KHÁCH HÀNG |
| CUSTOMER\_ADDRESS | nVarchar(50) | Not null | ĐỊA CHỈ KHÁCH HÀNG |
| CITY | nVarchar(50) | Not null | THÀNH PHỐ |
| STATE | nVarchar(10) | Not null | TÌNH TRẠNG |
| POSTAL\_CODE | Varchar(10) | Not null | MÃ BƯU ĐIỆN |

Table\_ORDER

|  |  |  |  |
| --- | --- | --- | --- |
| Field\_Name | Datatype | Constraints | Note |
| ORDER \_ID | Varchar(10) | Primary key | ID ĐẶT HÀNG |
| ORDER \_DATE | DATE | Not null | NGÀY ĐẶT HÀNG |
| CUSTOMER\_ID | Varchar(10) | foreign key | ID KHÁCH HÀNG |

Table\_PRODUCT

|  |  |  |  |
| --- | --- | --- | --- |
| Field\_Name | Datatype | Constraints | Note |
| PRODUCT \_ID | Varchar(10) | Primary key | ID SẢN PHẨM |
| PRODUCT \_DESCRIPTION | nVarchar(100) | Not null | MÔ TẢ |
| PRODUCT\_FINISH | nVarchar(10) | Not null | SẢN PHẨM HOÀN THIỆN |
| STANDARD\_PRICE | FLOAT | >=0 | GIÁ CHUẨN |
| STATE | nVarchar(10) | Not null | TÌNH TRẠNG |
| PRODUCT \_UNE\_ID | Varchar(10) | Not null | ID SẢN PHẨM UNE |

Table\_ORDER\_UNE

|  |  |  |  |
| --- | --- | --- | --- |
| Field\_Name | Datatype | Constraints | Note |
| ORDER \_ID | Varchar(10) | foreign key | ID ĐẶT HÀNG |
| PRODUCT \_ID | Varchar(10) | foreign key | ID SẢN PHẨM |
| ORDERED\_QUANTITY | INT | >=0 | SỐ LƯỢNG SẢN PHẨM |

create database pvms;

CREATE TABLE T\_CUSTOMER (

CUSTOMER\_ID VARCHAR(10) ,

CUSTOMER\_NAME NVARCHAR(50) NOT NULL,

CUSTOMER\_ADDRESS NVARCHAR(50) NOT NULL,

CITY NVARCHAR(50) NOT NULL,

STATE NVARCHAR(10) NOT NULL,

POSTAL\_CODE VARCHAR(10) NOT NULL,

constraint pk\_CUSTOMER\_ID primary key(CUSTOMER\_ID)

);

CREATE TABLE T\_ORDER (

ORDER\_ID VARCHAR(10) ,

ORDER\_DATE DATE NOT NULL,

CUSTOMER\_ID VARCHAR(10),

constraint pk\_ORDER\_ID primary key(ORDER\_ID),

constraint fk\_CUSTOMER\_ID foreign key(CUSTOMER\_ID) references T\_CUSTOMER(CUSTOMER\_ID)

);

CREATE TABLE T\_PRODUCT (

PRODUCT\_ID VARCHAR(10) ,

PRODUCT\_DESCRIPTION NVARCHAR(100) NOT NULL,

PRODUCT\_FINISH NVARCHAR(10) NOT NULL,

STANDARD\_PRICE FLOAT ,

STATE NVARCHAR(10) NOT NULL,

PRODUCT\_UNE\_ID VARCHAR(10) NOT NULL,

constraint pk\_PRODUCT\_ID primary key(PRODUCT\_ID),

constraint ch\_STANDARD\_PRICE check(STANDARD\_PRICE>=0)

);

CREATE TABLE T\_ORDER\_UNE (

ORDER\_ID VARCHAR(10),

PRODUCT\_ID VARCHAR(10),

ORDERED\_QUANTITY INT ,

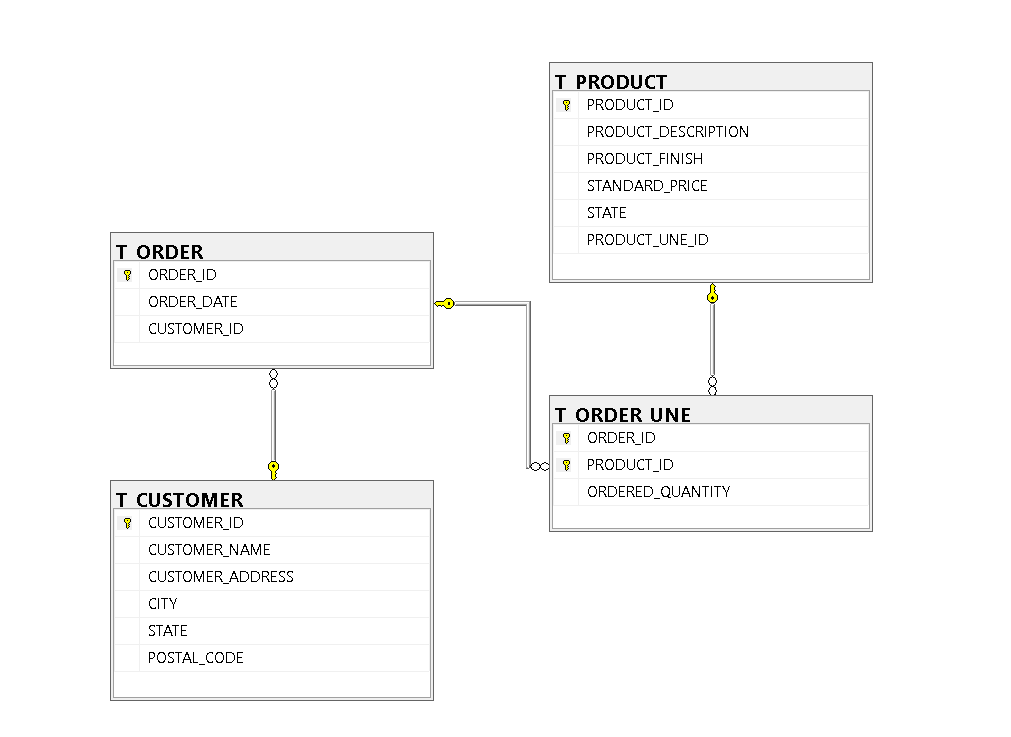
constraint pk\_ORDER\_UNE primary key(ORDER\_ID,PRODUCT\_ID),

constraint fk\_ORDER\_ID foreign key(ORDER\_ID) references T\_ORDER(ORDER\_ID),

constraint fk\_PRODUCT\_ID foreign key(PRODUCT\_ID) references T\_PRODUCT(PRODUCT\_ID),

constraint ch\_ORDERED\_QUANTITY check(ORDERED\_QUANTITY>=0)

);



Buổi 2 – 04-06-24

1. Cập nhật dữ liệu

Update <Table>

Set <Tên trường> = <Giá trị>

[Where <điều kiện>]

[From <Tên Bảng>];

/\* vd1: cập nhật địa điểm là Hội An cho phòng1 \*/

UPDATE phong

SET DiaDiem = N'Hội An'

WHERE MaPhong = 'P1';

2.

-- Xóa dữ liệu

-- DELETE FROM <table>

-- [WHERE <Điều kiện>]

3. Truy vấn dữ liệu

SECLEC [ALL] / [Top n] [Distinct] INTO <Tên bảng>

FROM <Tên bảng / View>

[WHERE <Điều kiện lọc> / <Điều kiện nối>]

[ORDER BY <Tên trường> ASC / DESC]

[GROUP BY <Trường nhóm>]

[HAVING <Điều kiện trên GROUP BY>]

Vd1: Hiển thị thông tin bảng nhân viên

SELECT \* FROM nhanvien;

Vd2: Hiển thị thông tin MaNV, TenNV

SELECT MaNV, TenNV FROM nhanvien;

Vd3: Hiển thị MaNV, HoLot, TenNV đổi tên trường thành tiếng việt

SELECT MaNV AS N’MÃ NHÂN VIÊN’, HoLot AS N‘HỌ LÓT’, TenNV AS N’TÊN NHÂN VIÊN’ FROM nhanvien;

Vd4: Hiển thị thông tin họ và tên của nhân viên

SELECT [HoLot] + ‘’ + [TenNV] AS N’Họ và tên’

FROM nhanvien;

Vd5: Tính tổng số nhân viên

SELECT COUNT (\*) AS N’TỔNG NHÂN VIÊN’

FROM nhanvien;

Vd6: Hiển thị 2 nhân viên đầu tiên

SELECT TOP 2 \* FROM nhanvien;

Vd7: Hiển thị 2 nhân viên đầu tiên và lưu vào bảng có tên NHANVIENTOP2

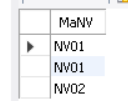
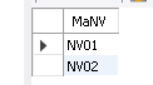
SELECT TOP 2 \* INTO NHANVIENTOP2 FROM nhanvien;

SELECT \* FROM NHANVIENTOP2;

Vd8: Hiển thị những nhân viên đã nhận dự án

SELECT DISTINCT MaNV

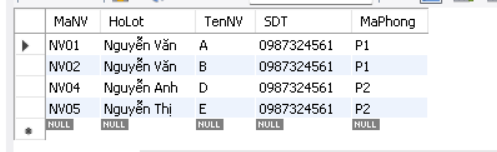
FROM NV\_DA;

Vd9: Hiển thị những nhân viên họ nguyễn

SELECT \* FROM NHANVIEN

WHERE HOLOT like ‘Nguyễn%’;



Vd10: Hiển thị những nhân viên N1, N7, N9

SELECT \* FROM NHANVIEN

WHERE (MANV = ‘N1’) OR (MANV = ‘N7’) OR (MANV = ‘N9’);

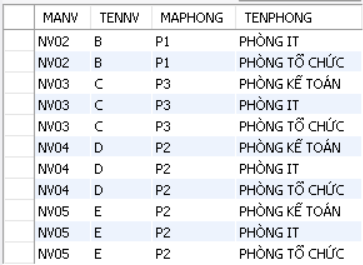
SELECT \* FROM NHANVIEN

WHERE MANV IN (‘N1’,’N7’,’N9’);

Vd11: Hiển thị thôn gtin MANV, TENNV, MAPHONG, TENPHONG

SELECT MANV, TENNV, A.MAPHONG, TENPHONG

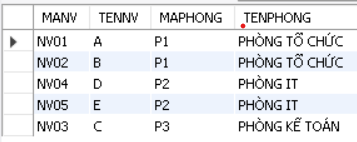
FROM NHANVIEN A, PHONG B



SELECT MANV, TENNV, A.MAPHONG, TENPHONG

FROM NHANVIEN A, PHONG B

WHERE A.MAPHONG = B.MAPHONG;

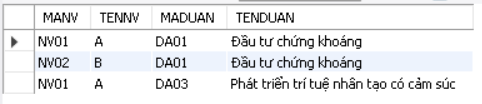


Vd12: Hiển thị thông tin MANV, TENNV, MADA, TENDA

SELECT nvda.MANV, nv.TENNV, nvda.MADUAN, da.TENDUAN

FROM NV\_DA nvda, NHANVIEN nv, DUAN da

WHERE nvda.MANV = nv.MANV && nvda.MADUAN = da.MADUAN;



Vd14: Cho biết có bao nhiêu nhân viên đã nhận dự án

SELECT count(distinct MANV) AS TONGNV

FROM NV\_DA;

Vd15: Hiển thị thông tin những dự án có mã là ‘d01, d05, d09’

Vd16: Hiển thị thông tin nhuwxg dự án có mã không phải là ‘d01, d05, d09’

Vd17: Hiển thị những dự án có kinh phí từ 1000 đến 5000

Vd18: Hiển thị những dự án có kinh phí nhỏ hơn 1000 và lớn hơn 5000

Vd19: Hiển thị thông tin nhân viên có tên sắp xếp tăng dần

Vd20: Hiển thị thông tin nhân viên có tên tăng dần và họ giảm dần

Vd21: Đếm xem mỗi phòng có bao nhiêu nhân viên

\*TRÊN WHERE LÀ ĐIỀU KIỆN TRÍCH LỌC KHÔNG ĐƯỢC SỬ DỤNG HÀM THỐNG KÊ

- HAVING LÀ ĐIỀU KIỆN TRÍCH LỌC THEO GROUP BY VÀ CÓ THỂ DÙNG CÁC HÀM THỐNG KÊ

11/06/2024

KHAI BÁO VÀ SỬ DỤNG BIẾN

1. BIẾN CỤC BỘ

DECLARE<TÊN BIẾN> <KIỂU DỮ LIỆU>

GÁN GIÁ TRỊ CHO BIẾN CỤC BỘ: SET

SET tên biến = giá trị

SET tên biến = biểu thức

SET tên biến = kết quả truy vấn

Vd: SET @Malop = ‘CMU-IS 401 AIS’

SET @Siso = (SELECT COUNT (\*) FROM SINHVIEN)

SET @Malophoc = ‘CMU’ + @Manganh + @Mamon + @Malop

2. BIẾN TOÀN CỤC @@

Ví dụ: @@error, @@rowcount: insert, update, delete

@@error = 0: Thao tác thành công

@@error = 1: Đang gặp lỗi

13/06/2024

THỦ TỤC LƯU TRÚ (STORED PROCEDURE - SP)

1. Khái niệm: Thủ tục được xây dựng từ các câu lệnh sql cơ bản và lưu trữ trên sql server. Muốn thực hiện thủ tục thì người sử dụng chỉ cần gửi 1 lời gọi hàm.

- Khi thủ tục được gọi lần đầu tiên thì nó sẽ được biên dịch qua 5 bước, những lời gọi tiếp theo ko cần biên dịch nữa.

2. Lợi ích mà thủ tục mang lại:

+ Cho phép điều chỉnh chu trình phù hợp

+ Tạo thủ tục 1 lần và gọi nó nhiều lần và nó hoàn toàn đọc lập với source code

+ Thủ tục cho phép thực thi nhanh hơn

+ Giảm bớt vấn đề về kẹt đường truyền mạng

+ Thủ tục có thể sử dụng trong vấn đề bảo mật của người dùng, người sử dụng sẽ được cấp quyền khi sử dụng thủ tục

3. Cú pháp

CREATE PROCEDURE <TÊN THỦ TỤC>

[@<TÊN THAM SỐ><KIỂU DỮ LIỆU>] [=GIÁ TRỊ] [OUTPUT]

AS

<CÂU LỆNH SQL>

17/06/2024

TRIGGER (BẪY LỖI)

1. Định nghĩa

TRIGGER là thủ tục lưu trú đặc biệt, nó sẽ tự động chạy mỗi khi có hành động liên quan đến nó, trigger ko có tham số, ko trả về giá trị giống như thủ tục

2. Các loại TRIGGER

- INSERT

- UPDATE

- DELETE

- TRIGGER TỔNG HỢP

3. Cú pháp

- Để tạo trigger ta dùng cú pháp:

CREATE TRIGGER <TÊN TRIGGER> ON <TABLE\_NAME>

FOR [INSERT, UPDATE, DELETE]

AS

BEGIN

CÁC CÂU LỆNH SQL

END

- Xóa trigger: DROP TRIGGER <TRIGGER\_NAME>

22-06-2024

HÀM NGƯỜI DÙNG

Khác với thủ tục hàm sẽ trả về một giá trị tại lời gọi của nó

Cú pháp

CREATE FUNCTION function\_name

( [@parameter\_name parameter\_data\_type] )

RETURNS [return Data\_type]

AS

BEGIN

Return [scalar value/select command]

END

HÀM SẼ TRẢ VỀ 1 TRONG 2 GIÁ TRỊ

- HÀM VÔ HƯỚNG – LƯU VÀO SCALAR VALUE FUNCTIONS

- HÀM TRẢ VỀ TABLE – LƯU VÀO TABLE VALUE FUNCTIONS

CHỈ MỤC – INDEX

1. Thiết kế khóa index: Khóa dành cho CSDL lớn, được thiết lập từ một hoặc nhiều cột dữ liệu cả bảng dữ liệu. Các giá trị của khóa index sẽ đưuọc sắp xếp và lưu trữ theo một danh sách. Và mỗi giá trị của khóa là duy nhất. Cấu trúc B\_tree để duyệt dữ liệu. Tăng tốc độ truy vấn.

2. Các loại chỉ mục – index:

- unique index: là chỉ mục duy nhất nhưng dữ liệu cho phép để trống

- clustered index (primary key): dữ liệu thật sự được sắp xếp vật lý theo chỉ mục. Thông thường khóa chính là kiểu clustered index, mỗi bảng thường là một khóa chính

- non-clusterd index(logic): là chỉ mục theo kiểu logic, dữ liệu thật sự không sắp xếp vật lý mà chỉ sắp xếp logic trên bộ nhớ tạm

Để view dữ liệu lên thì dùng nút là con trỏ chỉ đến vị trí vật lý trên đĩa để lấy dữ liệu

CÚ PHÁP:

CREATE CLUSTERED/ NON CLUSTERED / UNIQUE INDEX <TEN INDEX>

ON TABLE (TRUONG INDEX)

DROP INDEX TABLE(TEN INDEX)

Nên đặt chỉ mục index trong các trường hợp sau đây:

- Có nhu cầu truy xuất thường xuyên các bảng ghi trong table theo một số thuộc tính nào đó

Vd: Nhân viên ngân hàng có nhu cầu thường xuyên xem giao dịch với tổng tiền > 1000 thì nên đặt câu lệnh chỉ mục trên trường nào? Viết câu lệnh chỉ mục đó

CREATE NONCLUSTERED INDEX TONGINDEX

ON GIAODICH(TONGTIEN)

SELECT \* FROM GIAODICH

WHERE TONGTIEN > 1000

Nhân viên ngân hàng có nhu cầu truy xuất thường xuyên xem giao dịch trong ngày

CREATE NONCLUSTERED INDEX GDINDEX

ON GIAODICH(NGAYGD)

TRƯỜNG HỢP 2: NÊN CÀI ĐẶT CHỈ MỤC INDEX TRONG TRƯỜNG HỢP TẬP THUỘC TÍNH THAM GIA VÀO PHÉP KẾT NỐI CỦA 1

SINHVIEN(MASV, HOTEN, SDT)

KETQUA(MASV, MAMON, DIEM)

VD1: VIẾT CÂU LỆNH SQL THƯỜNG XUYÊN XEM ĐIỂM CHO SV: MASV, HOTEN, MAMON, DIEM

SELECT S.MASV, S.HOTEN, K.MAMON, K.DIEM

FROM SINHVIEN S, KETQUA K

WHERE (V.MASV = K.MASV) AND (V.MASV = ‘XXX’)