

*DataBase*

*Training Assignments*

|  |  |
| --- | --- |
| **Program Code** |  |
| **Issue/Revision** | **x/y** |
| **Effective date** | **06/2023** |
| **Author** | **dao.nguyenquang@vti.com.vn** |

**Answer Assignment Day 01**

# Tạo DB để quản lý thông tin nhân viên của VTI có thiết kế các bảng dữ liệu như bên dưới.

* **Bảng Department:**
  + DepartmentID: định danh của phòng ban (auto increment)
  + DepartmentName: tên đầy đủ của phòng ban (VD: sale, marketing, …)
* **Bảng Position:**
  + PositionID: định danh của chức vụ (auto increment)
  + PositionName: tên chức vụ (Dev, Test, Scrum Master, PM)
* **Bảng Account:**
  + AccountID: định danh của User (auto increment)
  + Email.
  + Username.
  + FullName.
  + DepartmentID: Phòng ban của user trong hệ thống.
  + PositionID: Chức vụ của User.
  + CreateDate: Ngày tạo tài khoản.

|  |
| --- |
| /\*============================== CREATE DATABASE =======================================\*/  /\*======================================================================================\*/  DROP DATABASE IF EXISTS TestingSystem\_DB;  CREATE DATABASE TestingSystem\_DB;  USE TestingSystem\_DB;  /\*============================== CREATE TABLE=== =======================================\*/  /\*======================================================================================\*/  -- create table 1: Department  DROP TABLE IF EXISTS Department;  CREATE TABLE Department(  DepartmentID TINYINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  DepartmentName VARCHAR(30) NOT NULL  );  -- create table 2: Posittion  DROP TABLE IF EXISTS Position;  CREATE TABLE `Position` (  PositionID TINYINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  PositionName ENUM('Dev','Test','Scrum Master','PM') NOT NULL  );  -- create table 3: Account  DROP TABLE IF EXISTS `Account`;  CREATE TABLE `Account`(  AccountID TINYINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  Email VARCHAR(50) NOT NULL UNIQUE KEY,  Username VARCHAR(50) NOT NULL UNIQUE KEY,  FullName VARCHAR(50) NOT NULL,  DepartmentID TINYINT UNSIGNED NOT NULL,  PositionID TINYINT UNSIGNED NOT NULL,  CreateDate DATETIME DEFAULT NOW(),  FOREIGN KEY(DepartmentID) REFERENCES Department(DepartmentID) ON DELETE CASCADE,  FOREIGN KEY(PositionID) REFERENCES `Position`(PositionID) ON DELETE CASCADE  ); |

# Thực hiện thêm mới dữ liệu vào các bảng dữ liệu vừa tạo.

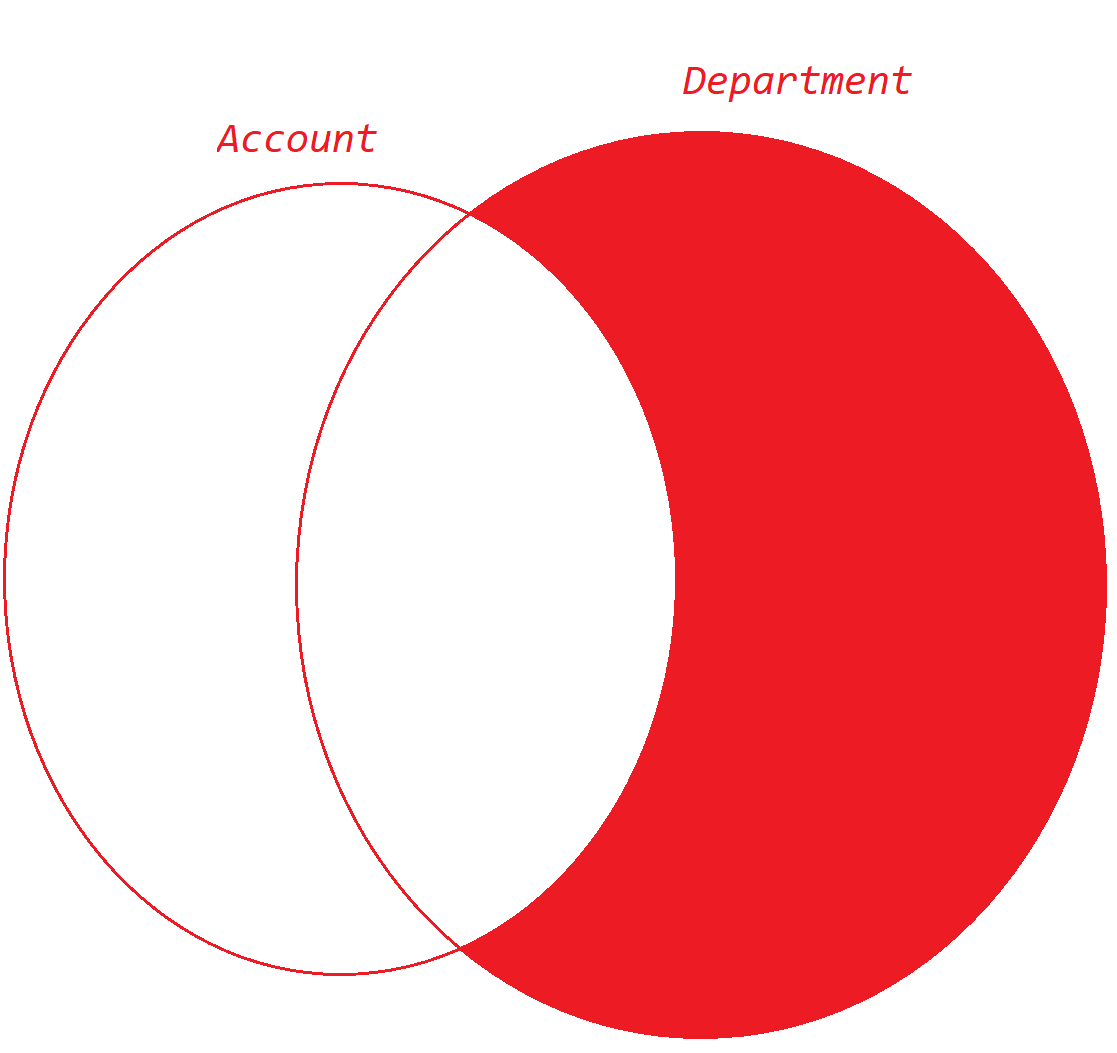
* Thực hiện thêm mới vào mỗi bảng dữ liệu 10 bản ghi.

|  |
| --- |
| /\*============================== INSERT DATABASE =======================================\*/  /\*======================================================================================\*/  -- Add data Department  INSERT INTO Department(DepartmentName)  VALUES  ('Marketing' ),  ('Sale' ),  ('Bảo vệ' ),  ('Nhân sự' ),  ('Kỹ thuật' ),  ('Tài chính' ),  ('Phó giám đốc'),  ('Giám đốc' ),  ('Thư kí' ),  ('Bán hàng' );    -- Add data position  INSERT INTO Position (PositionName )  VALUES ('Dev' ),  ('Test' ),  ('Scrum Master' ),  ('PM' );  -- Add data Account  INSERT INTO `Account`(Email , Username , FullName , DepartmentID , PositionID, CreateDate)  VALUES ('Email1@gmail.com' , 'Username1' ,'Fullname1' , '5' , '1' ,'2020-03-05'),  ('Email2@gmail.com' , 'Username2' ,'Fullname2' , '1' , '2' ,'2020-03-05'),  ('Email3@gmail.com' , 'Username3' ,'Fullname3' , '2' , '2' ,'2020-03-07'),  ('Email4@gmail.com' , 'Username4' ,'Fullname4' , '3' , '4' ,'2020-03-08'),  ('Email5@gmail.com' , 'Username5' ,'Fullname5' , '4' , '4' ,'2020-03-10'),  ('Email6@gmail.com' , 'Username6' ,'Fullname6' , '6' , '3' ,'2020-04-05'),  ('Email7@gmail.com' , 'Username7' ,'Fullname7' , '2' , '2' , NULL ),  ('Email8@gmail.com' , 'Username8' ,'Fullname8' , '8' , '1' ,'2020-04-07'),  ('Email9@gmail.com' , 'Username9' ,'Fullname9' , '2' , '2' ,'2020-04-07'),  ('Email10@gmail.com' , 'Username10' ,'Fullname10' , '10' , '1' ,'2020-04-09'); |

# Thực hiện truy vấn dữ liệu từ DB vừa tạo

* Lấy ra danh sách tất cả các phòng ban trên hệ thống.
* Lấy ra id của phòng ban có tên là “Sale”.
* Lấy ra danh sách nhân viên trên hệ thống, bao gồm cả thông tin Department và Position tương ứng.
* Lấy ra thông tin account có full name dài nhất.
* Đếm số lượng nhân viên trong từng phòng ban.
* Lấy ra danh sách các phòng ban có >3 nhân viên.
* Tìm phòng ban chưa có nhân viên nào.

|  |
| --- |
| -- - Lấy ra danh sách tất cả các phòng ban trên hệ thống.  SELECT \* FROM Department;  -- - Lấy ra id của phòng ban có tên là “Sale”.  SELECT DepartmentID FROM Department WHERE DepartmentName= "Sale";  -- - Lấy ra danh sách nhân viên trên hệ thống, bao gồm cả thông tin Department và Position tương ứng.  SELECT A.Email, A.Username , A.FullName, D.DepartmentName, P.PositionName  FROM `Account` A  INNER JOIN Department D ON A.DepartmentID = D.DepartmentID  INNER JOIN Position P ON A.PositionID = P.PositionID;  -- - Lấy ra thông tin account có full name dài nhất.  -- Sử dụng SubQuery  SELECT \*  FROM `Account`  WHERE LENGTH(Fullname) = (SELECT MAX(LENGTH(Fullname)) FROM `Account`)  ORDER BY Fullname DESC;  -- Sử dụng CTE  WITH CTE\_MaxLengFullname AS(  SELECT MAX(LENGTH(Fullname)) AS MaxLength FROM `Account`  )  SELECT \*  FROM `Account`  WHERE LENGTH(Fullname) = (SELECT MaxLength FROM CTE\_MaxLengFullname)  ORDER BY Fullname DESC;  -- - Đếm số lượng nhân viên trong từng phòng ban.  SELECT \* FROM `Account`;  SELECT a.DepartmentID, d.DepartmentName, COUNT(a.DepartmentID) AS Amount FROM `Account` a  INNER JOIN department d ON a.DepartmentID = d.DepartmentID  GROUP BY a.DepartmentID;  -- - Lấy ra danh sách các phòng ban có >3 nhân viên.  SELECT D.DepartmentName, count(a.DepartmentID) AS SL FROM account A  INNER JOIN department D ON a.DepartmentID = D.DepartmentID  GROUP BY A.DepartmentID  HAVING COUNT(A.DepartmentID) >3;  -- - Tìm phòng ban chưa có nhân viên nào.  SELECT D.DepartmentName FROM `Account` A  RIGHT JOIN Department D ON A.DepartmentID = D.DepartmentID  WHERE A.DepartmentID IS NULL |

****

# Thực hành các lệnh làm thay đổi dữ liệu

* Update thông tin của nhân viên có id = 5 thành tên "Nguyễn Đạo" và email thành daonq@vti.com.vn
* Xóa nhân viên có id=5.

|  |
| --- |
| -- - Update thông tin của nhân viên có id = 5 thành tên "Nguyễn Đạo" và email thành "daonq@vti.com.vn"  SELECT \* FROM `Account`;  UPDATE `Account`  SET Fullname = 'Nguyễn Đạo',  Email = 'daonq@vti.com.vn'  WHERE AccountID = 5;  -- - Xóa nhân viên có id=5.  DELETE FROM `Account` WHERE AccountID = 5; |

# Thực hành Store Procedure

* Tạo store Procedure để người dùng nhập vào tên phòng ban và in ra tất cả các  
  nhân viên thuộc phòng ban đó.

|  |
| --- |
| DROP PROCEDURE IF EXISTS sp\_GetAccFromDep;  DELIMITER $$  CREATE PROCEDURE sp\_GetAccFromDep(IN in\_dep\_name VARCHAR(50))  BEGIN  SELECT A.AccountID, A.FullName, D.DepartmentName FROM `account` A  INNER JOIN department D ON D.DepartmentID = A.DepartmentID  WHERE D.DepartmentName = in\_dep\_name;  END$$  DELIMITER ;  Call sp\_GetAccFromDep('Sale'); |

# Thực hành Trigger

* Tạo trigger không cho phép người dùng xóa tài khoản có email là admin@gmail.com (đây là tài khoản admin, không cho phép user xóa).
* Tạo trigger Không cho phép người dùng thêm bất kỳ user nào vào department "Sale" nữa, khi thêm thì hiện ra thông báo "Department Sale cannot add more user".

|  |
| --- |
| -- - Tạo trigger không cho phép người dùng xóa tài khoản có email là admin@gmail.com (đây là tài khoản admin, không cho phép user xóa).  DROP TRIGGER IF EXISTS trigger\_delete\_account;  DELIMITER $$  CREATE TRIGGER trigger\_delete\_account  BEFORE DELETE ON `Account`  FOR EACH ROW  BEGIN  DECLARE v\_Email VARCHAR(50);  SET v\_Email = 'admin@gmail.com';  IF (OLD.Email = v\_Email) THEN  SIGNAL SQLSTATE '12345'  SET MESSAGE\_TEXT = 'This User Admin, U cant delete it!!';  END IF;  END $$  DELIMITER ;  DELETE FROM account A WHERE A.AccountID = 1;  -- - Tạo trigger Không cho phép người dùng thêm bất kỳ user nào vào department "Sale" nữa, khi thêm thì hiện ra thông báo "Department Sale cannot add more user".  DROP TRIGGER IF EXISTS TrG\_NotAddUserToSale;  DELIMITER $$  CREATE TRIGGER TrG\_NotAddUserToSale  BEFORE INSERT ON `account`  FOR EACH ROW  BEGIN  DECLARE v\_depID TINYINT;  SELECT d.DepartmentID INTO v\_depID FROM department d WHERE d.DepartmentName = 'Sale';  IF (NEW.DepartmentID = v\_depID) THEN  SIGNAL SQLSTATE '12345'  SET MESSAGE\_TEXT = 'Cant add more User to Sale Department';  END IF;  END$$  DELIMITER ;  INSERT INTO `account` (`Email`, `Username`, `FullName`, `DepartmentID`, `PositionID`, `CreateDate`)  VALUES ('1', '1', '1', '2', '1', '2020-11-13 00:00:00'); |