TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

VIỆN KỸ THUẬT CÔNG NGHỆ



**ĐỒ ÁN MÔN HỌC**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**Đề tài**

**THỐNG KÊ, PHÂN TÍCH HOẠT ĐỘNG BÁN HÀNG**

GVHD: ThS. Nguyễn Thế Bảo

SVTH: 1. Nguyễn Thị Thanh Hằng - 2124802050024

2. Nguyễn Hữu Nghĩa - 2124802050013

3. Trần Trung Nguyên - 2124802050007

4. Trịnh Anh Tuấn - 2124801040059

Lớp: KTCN.CQ.07

Tháng 12/2021

**MỤC LỤC**

[**LỜI MỞ ĐẦU ii**](#_Toc120808490)

[**CHƯƠNG 1. QUẢN LÝ CƠ SỞ DỮ LIỆU (1đ) 1**](#_Toc120808491)

[1. Giới thiệu Cơ sở dữ liệu 1](#_Toc120808492)

[2. Mô tả Cơ sở dữ liệu: 1](#_Toc120808493)

[3. Sơ đồ mối quan hệ 6](#_Toc120808494)

[**CHƯƠNG 2. CÁC THAO TÁC TRÊN CƠ SỞ DỮ LIỆU (6.5đ) 7**](#_Toc120808495)

[1. Tạo các View 7](#_Toc120808496)

[2. Xây dựng các Stored procedure 8](#_Toc120808497)

[3. Xây dựng các Funtion 10](#_Toc120808498)

[4. Xây dựng các Trigger và Transaction 12](#_Toc120808499)

[5. Tạo các user 13](#_Toc120808500)

[**CHƯƠNG 3. KẾT LUẬN (0.5đ) 14**](#_Toc120808501)

[1. Kết quả đã làm được 14](#_Toc120808502)

[2. Hạn chế (chưa làm được) 14](#_Toc120808503)

[3. Thuận lợi và khó khăn 14](#_Toc120808504)

# LỜI MỞ ĐẦU

Cơ sở dữ liệu AdventureWorks là cơ sở dữ liệu mẫu được Microsoft xuất bản ban đầu để chỉ ra cách thiết kế cơ sở dữ liệu SQL Server bằng cách sử dụng SQL Server 2008. AdventureWorks là mẫu OLTP và AdventureWorksDW là mẫu kho dữ liệu. Thiết kế cơ sở dữ liệu đã tiến triển kể từ khi AdventureWorks được xuất bản lần đầu tiên. Để có cơ sở dữ liệu mẫu tận dụng các tính năng mới hơn của SQL Server, hãy xem WideWorldImporters. Lưu ý rằng AdventureWorks đã không thấy bất kỳ thay đổi đáng kể nào kể từ phiên bản 2012. Sự khác biệt duy nhất giữa các phiên bản AdventureWorks khác nhau là tên của cơ sở dữ liệu và mức độ tương thích của cơ sở dữ liệu. Để cài đặt cơ sở dữ liệu AdventureWorks với mức độ tương thích cơ sở dữ liệu của phiên bản SQL Server của bạn, bạn có thể cài đặt từ tệp sao lưu dành riêng cho phiên bản hoặc từ tập lệnh cài đặt. Có 3 phiên bản AdventureWorks (backup) là OLTP (dành cho hầu hết các khối lượng công việc xử lý giao dịch trực tuyến điển hình), Data Warehouse - DW (dành cho khối lượng công việc lưu trữ dữ liệu) và Lightweight – LT (là phiên bản nhẹ và giảm của mẫu OLTP).

# CHƯƠNG 1. QUẢN LÝ CƠ SỞ DỮ LIỆU (1đ)

1. **Giới thiệu Cơ sở dữ liệu**

Cơ sở dữ liệu minh họa AdventureWorks lấy bối cảnh trên dữ liệu của 1 công ty hư cấu có tên Adventure Works Cycles. Đây được cho là 1 công ty sản xuất đa quốc gia có qui mô lớn. Công ty sản xuất và kinh doanh xe đạp làm từ kim loại và các chất liệu tổng hợp. Thị trường của công ty này bao gồm cả khu vực Bắc Mỹ, Châu Âu và Châu Á. Trong khi trụ sở chính của công ty lại được đặt ở Bothell, Washington gồm có 290 nhân viên, công ty cũng có 1 vài nhóm nhân viên kinh doanh khu vực ở các thị trường hoạt động của mình.

Vào năm 2000, công ty Adventure Works Cycles có mua lại 1 nhà máy sản xuất nhỏ là Importadores Neptuno ở Mexico. Nhà máy này tham gia sản xuất 1 vài thành phần thiết yếu trong toàn bộ dây chuyền sản xuất ra thành phẩm của công ty. Các thành phần được nhà máy này sản xuất sẽ được chuyển tới trụ sở chính ở Bothell để lắp ráp thành thành phẩm. Năm 2001, Importadores Neptuno trở thành nhà máy duy nhất sản xuất phà phân phối dòng sản phẩm xe đạp du lịch.

Kết thúc 1 năm tài chính thành công, công ty Adventure Works Cycles đang tìm cách mở rộng thị phần bằng cách tập trung vào các hoạt động bán hàng cho các khách hàng quan trọng nhất của họ, mở rộng thông tin sản phẩm thông qua hệ thống Website đồng thời tiết giảm chi phí bán hàng bằng cách giảm chi phí sản xuất.

1. **Mô tả Cơ sở dữ liệu:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Production.Product** | | | |
| **Description:** Sản phẩm đã bán hoặc được sử dụng để sản xuất các sản phẩm đã bán. | | | |
| **Serial No.** | **Logical Name** | **Data Type** | **NULL** |
| 1 | ProductID | int |  |
| 2 | Name | nvarchar(50) |  |
| 3 | ProductNumber | nvarchar(50) |  |
| 4 | MakeFlag | bit |  |
| 5 | FinishedGoodsFlag | bit |  |
| 6 | Color | nvarchar(15) | Y |
| 7 | SafetyStockLevel | smallint |  |
| 8 | ReorderPoint | smallint |  |
| 9 | StandardCost | money |  |
| 10 | ListPrice | money |  |
| 11 | Size | nvarchar(5) | Y |
| 12 | SizeUnitMeasureCode | nchar(3) | Y |
| 13 | WeightUnitMeasureCode | nchar(3) | Y |
| 14 | Weight | decimal(8, 2) |  |
| 15 | DaysToManufacture | int | Y |
| 16 | ProductLine | nchar(2) | Y |
| 17 | Class | nchar(2) | Y |
| 18 | Style | nchar(2) | Y |
| 19 | ProductSubcategoryID | int | Y |
| 20 | ProductModelID | int | Y |
| 21 | SellStartDate | datetime |  |
| 22 | SellEndDate | datetime | Y |
| 23 | DiscontinuedDate | datetime | Y |
| 24 | rowguid | uniqueidentifier |  |
| 25 | ModifiedDate | datetime |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Sales.SalesOrderHeader** | | | |
| **Description:** Thông tin chung về đơn đặt hàng. | | | |
| **Serial No.** | **Logical Name** | **Data Type** | **NULL** |
| 1 | SalesOrderID | int |  |
| 2 | RevisionNumber | tinyint |  |
| 3 | OrderDate | datetime |  |
| 4 | DueDate | datetime |  |
| 5 | ShipDate | datetime | Y |
| 6 | Status | tinyint |  |
| 7 | OnlineOrderFlag | bit |  |
| 8 | SalesOrderNumber | nvarchar(25) |  |
| 9 | PurchaseOrderNumber | nvarchar(25) |  |
| 10 | AccountNumber | nvarchar(15) | Y |
| 11 | CustomerID | int |  |
| 12 | SalesPersonID | int | Y |
| 13 | TerritoryID | int | Y |
| 14 | BillToAddressID | int |  |
| 15 | ShipToAddressID | int |  |
| 16 | ShipMethodID | int |  |
| 17 | CreditCardID | int | Y |
| 18 | CreditCardApprovalCode | varchar(15) | Y |
| 19 | CurrencyRateID | int | Y |
| 20 | SubTotal | money |  |
| 21 | TaxAmt | money |  |
| 22 | Freight | money |  |
| 23 | TotalDue | money |  |
| 24 | Comment | nvarchar(128) | Y |
| 25 | rowguid | uniqueidentifier |  |
| 26 | ModifiedDate | datetime |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Sales.SalesOrderDetail** | | | |
| **Description:** Các sản phẩm riêng lẻ được liên kết với một đơn đặt hàng cụ thể. Xem SalesOrderHeader. | | | |
| **Serial No.** | **Logical Name** | **Data Type** | **NULL** |
| 1 | SalesOrderID | int |  |
| 2 | SalesOrderDetailID | int |  |
| 3 | CarrierTrackingNumber | nvarchar(25) | Y |
| 4 | OrderQty | smallint |  |
| 5 | ProductID | int |  |
| 6 | SpecialOfferID | int |  |
| 7 | UnitPrice | money |  |
| 8 | UnitPriceDiscount | money |  |
| 9 | LineTotal | numeric(38,6) |  |
| 10 | rowguid | uniqueidentifier |  |
| 11 | ModifiedDate | datetime |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Production.ProductListPriceHistory** | | | |
| **Description:** Thay đổi giá niêm yết của sản phẩm theo thời gian. | | | |
| **Serial No.** | **Logical Name** | **Data Type** | **NULL** |
| 1 | ProductID | int |  |
| 2 | StartDate | datetime |  |
| 3 | EndDate | datetime | Y |
| 4 | ListPrice | money |  |
| 5 | ModifiedDate | datetime |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Production.ProductSubcategory** | | | |
| **Description:** Các danh mục phụ của sản phẩm. Xem bảng ProductCategory. | | | |
| **Serial No.** | **Logical Name** | **Data Type** | **NULL** |
| 1 | ProductSubcategoryID | int |  |
| 2 | ProductCategoryID | int |  |
| 3 | Name | nvarchar(50) |  |
| 4 | rowguid | uniqueidentifier |  |
| 5 | ModifiedDate | datetime |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Production.ProductCategory** | | | |
| **Description:** Phân loại sản phẩm cấp cao. | | | |
| **Serial No.** | **Logical Name** | **Data Type** | **NULL** |
| 1 | ProductCategoryID | int |  |
| 2 | Name | nvarchar(50) |  |
| 3 | rowguid | uniqueidentifier |  |
| 4 | ModifiedDate | datetime |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Production.ProductModel** | | | |
| **Description:** Phân loại mô hình sản phẩm. | | | |
| **Serial No.** | **Logical Name** | **Data Type** | **NULL** |
| 1 | ProductModelID | int |  |
| 2 | Name | nvarchar(50) |  |
| 3 | CatalogDescription | xml |  |
| 4 | Instructions | xml |  |
| 5 | rowguid | uniqueidentifier |  |
| 6 | ModifiedDate | datetime |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Production.ProductReview** | | | |
| **Description:** Đánh giá của khách hàng về sản phẩm họ đã mua. | | | |
| **Serial No.** | **Logical Name** | **Data Type** | **NULL** |
| 1 | ProductReviewID | int |  |
| 2 | ProductID | int |  |
| 3 | ReviewerName | nvarchar(50) |  |
| 4 | ReviewDate | datetime |  |
| 5 | EmailAddress | nvarchar(50) |  |
| 6 | Rating | int |  |
| 7 | Comments | nvarchar(3850) | Y |
| 8 | ModifiedDate | datetime |  |

1. **Diagram

   Description automatically generatedSơ đồ mối quan hệ**

# CHƯƠNG 2. CÁC THAO TÁC TRÊN CƠ SỞ DỮ LIỆU (6.5đ)

1. **Tạo các View**
   1. View đơn giản trên một bảng:

CREATE VIEW ViewProductionID AS

SELECT ProductCategoryID

FROM Production.ProductCategory;

SELECT \*

FROM ViewProductionID

* 1. View đơn giản trên nhiều bảng:
  2. ***Nhân dịp ngày 20/10 là ngày phụ nữ Việt Nam nhằm tôn vinh vẻ đẹp của người phụ nữ và để cảm ơn những đóng góp tích cực vào công cuộc xây dựng đất nước thì công ty sẽ chọn ra 3 khách hàng may mắn nhất để nhận 1 vé máy bay đi Hokkaidō (Khách hàng đó phải là người đã mua sản phẩm của công ty, nếu mua càng nhiều tỷ lệ trúng giải càng cao, doanh thu từ những khách hàng đã mua và có đánh giá sản phẩm phải từ 2000 đô trở lên), trường hợp có nhiều tỉ lệ trùng nhau thì lấy random. Tạo 1 view có tên luckyPeople để hiển thị xác suất trúng giải, tên của người may mắn, email, tên sản phẩm được người đó mua nhiều nhất.*** (View phức tạp):

CREATE VIEW luckyPeople AS

SELECT TOP 3 CONVERT(float, Count(\*)) /

(SELECT Count(\*) FROM Production.ProductReview) \* 100

AS PercentS,

ReviewerName,

EmailAddress, Production.Product.Name

FROM Production.ProductReview, Production.Product

WHERE (SELECT SUM(ListPrice)

FROM Production.ProductReview as a, Production.Product as b

WHERE a.ProductID = b.ProductID

) >= 2000 AND Production.ProductReview.ProductId = Production.Product.ProductId

GROUP BY

EmailAddress,

ReviewerName,

Production.Product.Name

ORDER BY PercentS DESC, NEWID()

-- Random all row bằng NEWID() xong rồi mới sort theo desc

* 1. ***Tạo 1 view tên là showProductFew để xem các sản phẩm, có từ 3 đánh giá trở xuống. Hiển thị productId, tên sản phẩm, tên người đánh giá, email của họ.*** (View phức tạp):

CREATE VIEW showProductFew AS

SELECT

a.ProductID, b.Name, a.ReviewerName, a.Email

FROM

Production.ProductReview as a,

Production.Product as b

WHERE a.Rating <= 3 AND a.ProductID = b.ProductID

* 1. ***Vì sản phẩm có mã là 965 có 1 vài khách hàng đánh giá dưới 3 vì tên sản phẩm không rõ ràng, cho nên công ty quyết định đổi lại tên sản phẩm là ‘Jogger Product’ nếu sản phẩm có màu ‘Yellow’ và mã sản phẩm là 965***. (View phức tạp cập nhật dữ liệu):

UPDATE showProductFew

SET Name = 'Orange Product'

FROM Production.Product, showProductFew

WHERE showProductFew.ProductID = 965

AND showProductFew.ProductID = Production.Product.ProductID

AND Color = 'Yellow'

1. **Xây dựng các Stored procedure**
   1. ***Tạo 1 thủ tục kiểm tra thông tin chi tiết của các sản phẩm bao gồm: productId, nhận xét của người đánh giá sản phẩm, số lượng tồn kho tối thiểu, giá bán, đánh giá sản phẩm do người đánh giá đưa ra.*** (Thủ tục không tham số)

CREATE PROCEDURE checkInfoProduct

AS

BEGIN

SELECT a.ProductId, a.Comments, b.ReorderPoint, b.ListPrice, a.Rating

FROM Production.Review as a, Production.Product as b

WHERE a.ProductId = b.ProductId

END

EXECUTE checkInfoProduct;

* 1. ***Tạo 1 thủ tục tính tổng tất cả giá bán của những sản phẩm từ giá X trở lên, với X là tham số truyền vào, nếu không có thì mặc định là từ 1000 ($) trở lên.*** (Thủ tục có tham số mặc định)

CREATE PROCEDURE sumListPrice(@X money = 1000)

AS

BEGIN

SELECT SUM(ListPrice) AS N’Tổng giá bán’

FROM Production.Product

WHERE ListPrice >= @X

END

EXECUTE sumListPrice 3000;

* 1. ***Viết một thủ tục dùng để xem tên sản phẩm và giá bán từ đầu năm Y mà bản ghi đó được cập nhật lần cuối, với một tham số đầu vào và một tham số đầu ra. Tham số @Y nhận giá trị đầu vào theo chỉ định khi gọi thủ tục, tham số @product\_count được sử dụng để chứa giá trị trả về của thủ tục.*** (Thủ tục có tham số output)

CREATE PROCEDURE uspFindProductByModel(@Y INT, @product\_count INT OUTPUT)

AS

BEGIN

SELECT Name, ListPrice

FROM Production.Product

WHERE YEAR(ModifiedDate) = @Y;

SELECT @product\_count = @@ROWCOUNT;

END

DECLARE @sumRowEffected int;

EXECUTE uspFindProductByModel

@Y = 2014,

@product\_count = @sumRowEffected OUT;

SELECT CONVERT(nvarchar(100), @sumRowEffected) + N' cái' as N'Tổng số sản phẩm';

* 1. ***Tạo thủ tục kiểm tra số lượng tồn kho tối thiểu khi nhập vào ProductID.*** (Thủ tục có tham số input)

CREATE PROCEDURE checkReorderPoint(@productId int)

AS

BEGIN

SELECT ReorderPoint

FROM Production.Product

WHERE ProductId = @productId;

END

EXECUTE checkReorderPoint 1;

* 1. ***Tạo thủ tục nhập ProductID rồi xuất ra ngày và giờ bản ghi được cập nhật lần cuối ở bảng Production.Product.*** (Thủ tục có tham số input)

CREATE PROCEDURE show\_modifiedDate(@id int)

AS

BEGIN

SELECT ModifiedDate

FROM Production.Product

WHERE @id = ProductID;

END

EXECUTE show\_list\_modifiedDatee 1;

1. **Xây dựng các Funtion**
   1. ***Trả về tên của sản phẩm khi nhập productId.*** (Hàm trả về kiểu vô hướng)

CREATE FUNCTION FIND\_NAME\_CATEGORY(@productId int)

RETURNS VARCHAR(10)

AS

BEGIN

DECLARE @result varchar(10)

SELECT @result = Name

FROM Production.Product

WHERE @productId = ProductID

RETURN @result

END

SELECT dbo.FIND\_NAME\_CATEGORY(3) as productName

* 1. ***Trả về số ngày cần thiết để sản xuất sản phẩm khi nhập productID.*** (Hàm trả về kiểu vô hướng)

CREATE FUNCTION FIND\_DaysToManufacture(@productId int)

RETURNS INT

AS

BEGIN

DECLARE @result int

SELECT @result = DaysToManufacture

FROM Production.Product

WHERE @productId = ProductID

RETURN @result

END

SELECT dbo.FIND\_DaysToManufacture(316) as N'Số ngày cần để sản xuất';

* 1. ***Tạo 1 Function trả về bảng có sẳn có 1 tham số truyền vào là @year chính là năm bản ghi được chỉnh sửa lần cuối. Yêu cầu trả về bảng chứa các thông tin như: tên sản phẩm, màu sắc, kích thước sản phẩm.*** (Hàm trả về bảng có sẳn)

CREATE FUNCTION infoBasicProduct (@year int)

RETURNS TABLE

AS

RETURN SELECT Name, Color, Size

FROM Production.Product

WHERE YEAR(ModifiedDate) = @year

SELECT \* FROM dbo.infoBasicProduct(2008)

* 1. ***Trả về 1 bảng chi tiết các thông tin đánh giá của khách hàng về sản phẩm mà họ đã mua khi nhập email.*** (Hàm trả về bảng có sẳn)

CREATE FUNCTION ProductReviewEmail(@email nvarchar(50))

RETURNS TABLE

AS

RETURN SELECT \*

FROM Production.ProductReview

WHERE EmailAddress = @email

SELECT \* FROM dbo.ProductReviewEmail('john@fourthcoffee.com')

* 1. ***Nhập email sau đó trả về, mức độ của sản phẩm đó được gọi là cột modeProduct quy định nếu rating >= 4 thì sản phẩm được đánh giá là “Thích”, ngược lại "Không thích" sau đó là comments và đánh giá sản phẩm khách hàng đó đã cho.*** (Hàm trả về bảng tự định nghĩa)

CREATE FUNCTION CALC\_AVG\_RATING (@email nvarchar(50))

RETURNS @newTable table (modeProduct nvarchar(50), Comment nvarchar(1000), rating int)

AS

BEGIN

INSERT INTO @newTable

SELECT modeProduct = IIF(RATING >= 4, N'Thích', N'Không thích'),Comments ,RATING

FROM Production.ProductReview

WHERE EmailAddress = @email

RETURN

END

SELECT \* FROM dbo.CALC\_AVG\_RATING('john@fourthcoffee.com')

1. **Xây dựng các Trigger và Transaction**
   1. ***In ra thông tin đối với những ProductId có số 9 ở đầu sau khi thực hiện thao tác INSERT trên bảng Production.ProductReview.*** (Trigger Insert)

CREATE TRIGGER insert\_trigger

ON Production.ProductReview

FOR INSERT

AS

BEGIN

SELECT \*

FROM Production.ProductReview

WHERE ProductID like '9%'

END

* 1. ***Đếm số lượng comments theo từng sản phẩm trong bảng Production.ProductReview sau khi thực hiện thao tác DELETE trên bảng Production.ProductReview. (Trigger Delete)***

CREATE TRIGGER number\_of\_comments

ON Production.ProductReview

FOR DELETE

AS

BEGIN

SELECT ProductID, count(Comments) AS 'Number of comments'

FROM Production.ProductReview

GROUP BY ProductID

END

* 1. ***In ra các đánh giá sản phẩm của những khách hàng đã mua sau khi thực hiện thao tác UPDATE trên bảng Production.ProductReview.*** (Trigger Update)

CREATE TRIGGER trigger\_update

ON Production.ProductReview

FOR UPDATE

AS

BEGIN

SELECT \*

FROM Production.ProductReview

END

* 1. ***Đặt ReorderPoint về 0 nếu ProductID = 317*** (Transaction)

BEGIN TRANSACTION

UPDATE Production.Product

SET ReorderPoint = 0

WHERE ProductID = 317

COMMIT

SELECT \* FROM Production.Product

* 1. ***Quay lui lệnh đặt ReorderPoint về 0 nếu ProductID = 317*** (Transaction)

BEGIN TRANSACTION

UPDATE Production.Product

SET ReorderPoint = 0

WHERE ProductID = 317

ROLLBACK

SELECT \* FROM Production.Product

1. **Tạo các user**

+ Đầu tiên tạo tên đăng nhập có tên là: ‘nhom6’ và mật khẩu đăng nhập ‘0000’.

*Truy vấn*: CREATE LOGIN quocthang WITH PASSWORD = '0000'

+ Tạo user có tên là: ‘CEO’ và cấp quyền CONTROL trên database AdventureWorks2019 cho tài khoản có tên là ‘quocthang’.

*Truy vấn*:

USE AdventureWorks2019

CREATE USER CEO FOR LOGIN quocthang

GRANT CONTROL ON DATABASE::AdventureWorks2019 TO CEO

+ Tiếp theo tạo tên đăng nhập có tên là: ‘detai6’ và mật khẩu đăng nhập ‘0000’.

*Truy vấn*: CREATE LOGIN noname WITH PASSWORD = '0000'

+ Tạo user có tên là: ‘CSKH’ và cấp quyền truy cập (UPDATE, SELECT, ALTER) trên bảng Production.ProductReview cho tài khoản có tên là ‘noname’.

*Truy vấn*:

USE AdventureWorks2019

CREATE USER CSKH FOR LOGIN noname

GRANT UPDATE, SELECT, ALTER ON Production.ProductReview TO CSKH

# CHƯƠNG 3. KẾT LUẬN (0.5đ)

1. **Kết quả đã làm được**

Sau thời gian tìm hiểu, chúng em đã đạt được những kết quả sau:

* + - Tạo View
    - Xây dựng Stored Procedure
    - Xây dựng Function
    - Xây dựng Trigger và Transaction
    - Tạo user

1. **Hạn chế (chưa làm được)**
2. **Thuận lợi và khó khăn**
   1. **Thuận lợi:**

* Trong quá trình làm View, Store Procedure, Trigger,… nhóm em được lên mạng tìm hiểu các cú pháp và có nhiều tài liệu để nghiên cứu.
* Được làm cùng nhóm, được các bạn trong nhóm hỗ trợ.
  1. **Khó khan:**
* Lần đầu làm bài tiểu luận dạng tự đặt vấn đề,…nhóm 1 gặp khó khăn trong việc phải suy nghĩ, đưa ra vấn đề và giải quyết các thao tác trên cơ sở dữ liệu.

**TÀI LIỆU THAM KHẢO**

**[1]** Article (2022), “AdventureWorks sample databases”,

<https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms>

truy cập ngày 01/10/2022.

**[2]** Article (2022), “GRANT Database Permissions”,

<https://learn.microsoft.com/en-us/sql/t-sql/statements/grant-database-permissions-transact-sql?source=recommendations&view=sql-server-ver16>,

truy cập ngày 10/10/2022.

**[3]** iAnywhere Solutions, Inc (2010), “Granting permission on tables”,

<https://dcx.sap.com/1200/en/dbadmin/pergrtv.html>,

truy cập ngày 10/10/2022.

**[4]** StackOverFlow, “order by newid() – how does it work”,

<https://stackoverflow.com/questions/4979799/order-by-newid-how-does-it-work>,

truy cập ngày 22/10/2022.

**HƯỚNG DẪN ĐỊNH DẠNG**

1. Định dạng trang: lề trái 3cm, phải 2cm, trên 2.5cm, dưới 2.5cm
2. Font chữ: Times New Roman
3. Size: 13 cho toàn bộ nội dung (ngoại trừ các tiêu đề chương, phần thì size 14)
4. Nội dung văn bản canh đều 2 bên
5. Hình ảnh, bảng (nếu có) canh giữa, đặt tên hình, tên bảng theo thứ tự tăng dần