

## **BÀI THỰC HÀNH 2**

### **Truy vấn Metadatas**

---

#### **1. Mục tiêu**

Sau khi thực hành bài này, sinh viên có thể biết truy vấn Metadatas thông qua Information\_schema của SQL SERVER 2000.

#### **2. Yêu cầu**

- Phục hồi CSDL: QL\_HANGHOA
- Sinh viên thực câu lệnh SQL sẵn có.
- Ghi nhận kết quả truy vấn.

#### **3. Bài tập**

- **SQL 1: Liệt kê các bảng của CSDL**

```
Select *
From Information_schema.TABLES
Where Table_type = 'BASE TABLE' and
      Table_Name <> 'dtproperties'
```

- **SQL 2: Liệt kê thông tin các cột của các bảng CSDL QLHANGHOA**

```
Select Column_name, Data_type, Character_maximum_length,
      Numeric_precision, Numeric_scale
From INFORMATION_SCHEMA.COLUMNS
Where Table_Name = 'DAILY' or Table_Name = 'HANGHOA' or
      Table_Name = 'MUA' or Table_Name = 'BAN'
```

- **SQL 3: Liệt kê các cột của các constraints**

```
Select Table_name, Constraint_name, Column_name
From INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE
Where Table_Name <> 'dtproperties'
Order by Table_name, Constraint_name
```

- **SQL 4: Liệt kê tên bảng và các constraints ở bảng đó**

```
Select Constraint_catalog, Table_name, Constraint_name,
      Constraint_type
From INFORMATION_SCHEMA.TABLE_CONSTRAINTS
Where Table_Name <> 'dtproperties'
Order by Table_name, Constraint_name
```

- **SQL 5: Liệt kê tên bảng, tên constraint, tên cột của các khóa**

```
Select Constraint_catalog, Table_name, Constraint_name,  
       Column_name  
From INFORMATION_SCHEMA.KEY_COLUMN_USAGE  
Where Table_Name <> 'dtproperties'  
Order by Constraint_catalog, Table_name
```

- **SQL 6: Liệt kê các check constraints**

```
Select *  
From information_schema.check_constraints
```

- **SQL 7: Liệt kê thông tin có liên quan về khóa chính của mỗi bảng**

```
Select c.Table_Name, cc.Constraint_Name, c.Column_Name  
From Information_schema.columns c,  
     Information_schema.constraint_column_usage cc,  
     Information_schema.table_constraints T  
Where c.Table_Name = cc.Table_Name and  
       c.Column_Name = cc.Column_Name and  
       cc.Table_Name = T.Table_Name and  
       cc.Constraint_Name = T.Constraint_Name and  
       T.Constraint_Type = 'PRIMARY KEY' and  
       cc.Table_Name <> 'dtproperties'  
Order by c.table_name
```

- **SQL 8: Liệt kê thông tin về khóa ngoại của mỗi bảng**

```
Select c.Table_Name, cc.Constraint_Name, c.Column_Name  
From Information_schema.Columns c,  
     Information_schema.Constraint_column_usage cc,  
     Information_schema.Table_Constraints T  
Where c.Table_Name = cc.Table_Name and  
       c.Column_name = cc.Column_Name and  
       cc.Table_name = T.Table_Name and  
       cc.Constraint_name = T.Constraint_Name and  
       T.Constraint_Type = 'FOREIGN KEY' and  
       cc.Table_Name <> 'dtproperties'  
Order by c.Table_name
```

- **SQL 9: Tạo một stored procedure cho phép sinh ra câu lệnh tạo bảng của một bảng nào đó (tên bảng là tham số)**

```
--Generate create table -----
Create procedure procGenSpForCreateTable
    @tablename varchar(255)
AS
Begin
    declare @sqlscript varchar(8000)
    declare @sqlscriptparms varchar(8000)
    declare @column_name nvarchar(128)
    declare @data_type nvarchar(128)
    declare @character_maximum_length int
    declare @numeric_precision int
    declare @numeric_scale int
    declare @is_nullable char(3)
    declare @default nvarchar(128)
    declare @newline char(1)
    declare @tab char(4)
    Declare @PK varchar(255)

    ---Create procedure header-----
    select @newline = char(13)
    select @tab='    '
    select @sqlscriptparms = ''
    select @sqlscript = 'Create Table  '+ @tablename

    --Create parameter list and string for insert-----
    declare columns_cursor cursor for
        select  column_name, data_type, character_maximum_length,
                numeric_precision, numeric_scale, is_nullable,
                column_default
        from    information_schema.columns
        where   table_name=@tablename
```

```

Open columns_cursor

fetch next from columns_cursor into
    @column_name, @data_type, @character_maximum_length,
    @numeric_precision, @numeric_scale, @is_nullable, @default

while @@fetch_status = 0
Begin
    if @sqlscriptparms <> ''
        select @sqlscriptparms = @sqlscriptparms + ',' +
            @newline + @tab
    else select @sqlscriptparms = @sqlscriptparms +
        @newline + @tab

    select @sqlscriptparms = @sqlscriptparms +
        rtrim(@column_name) + ' ' + @data_type +
        case
            when @data_type in ('char', 'varchar',
                'binary', 'varbinary', 'nchar', 'nvarchar')
then '(' + cast(@character_maximum_length as varchar(10)) + ')'
            when @data_type in ('decimal', 'numeric')
then '(' + cast(@numeric_precision as varchar(3)) + ',' +
                cast(@numeric_scale as varchar(3)) + ')'

            else ''
        end

    if @is_nullable = 'No'
        select @sqlscriptparms = @sqlscriptparms +
            ' Not Null'
    if @default is not null
        select @sqlscriptparms = @sqlscriptparms +
            ' DEFAULT ' + @default

```

```

        fetch next from columns_cursor into
            @column_name, @data_type,
            @character_maximum_length, @numeric_precision,
            @numeric_scale, @is_nullable, @default
    End

    Close columns_cursor
    Deallocate columns_cursor

    -- Create Primary Key
    Select @PK= @tab + 'Primary Key ('
    select  @PK=@PK+cc.column_name + ', '
    from    information_schema.constraint_column_usage cc
           Inner Join information_schema.table_constraints T
                   On T.table_name = cc.table_name
                   and cc.constraint_name = T.constraint_name
    where   T.table_name = @tablename and
           T.constraint_TYPE = 'PRIMARY KEY'

    if @PK = @tab + 'Primary Key (' --- No Primary Key
        Select @PK=@newline
    else Select @PK = ',' + @newline + left(@PK, len(@PK)-1) +
        ')' + @newline

    select @sqlscript = @sqlscript + @newline + '(' +
        @sqlscriptparms + @PK + ')' + @newline
    select @sqlscript = @sqlscript + 'Go' + @newline
    select @sqlscript = @sqlscript +
        '-----' +
        @newline

    print @sqlscript
End
-----
Execute procGenSpForCreateTable 'HANGHOA'

```

- **SQL 10:** Liệt kê các check constraints của 1 bảng nào đó, thông tin cần liệt kê: tên bảng, tên constraint, mệnh đề của check constraint
- **SQL 11:** Liệt kê thông tin của các khóa ngoại của 1 bảng, thông tin cần liệt kê: tên bảng, tên khóa ngoại, tên cột là khóa ngoại, bảng chứa khóa chính, tên bảng chứa khóa chính.