
T-SQL Exercises

View Definition

-- Task 1

-- Write a **SELECT** statement to return the **productid**, **productname**, **supplierid**, **unitprice**, and **discontinued** columns from the **Production.Products** table.

-- Filter the results to include only products that belong to **categoryid** equals 1.

```
SELECT productid, productname, supplierid, unitprice, discontinued
FROM Production.Products
WHERE categoryid = 1;
GO
```

-- Modify the T-SQL code to include the following supplied T-SQL statement. Put this statement before the **SELECT** clause:

-- **CREATE VIEW Production.ProductsBeverages AS**

-- Execute the complete T-SQL statement. This will create an object view named **ProductBeverages** under the **Production** schema.

```
CREATE VIEW Production.ProductsBeverages
AS
SELECT productid, productname, supplierid, unitprice, discontinued
FROM Production.Products
WHERE categoryid = 1;
GO
```

-- Task 2

-- Write a **SELECT** statement to return the **productid** and **productname** columns from the **Production.ProductsBeverages** view.

-- Filter the results to include only products where supplierid equals 1.

```
SELECT productid, productname
FROM Production.ProductsBeverages
WHERE supplierid = 1;
GO
```

-- Task 3

-- The IT department has written a T-SQL statement that adds an ORDER BY clause to the view created in task 1.

-- Execute the provided code. What happened? What is the error message? Why did the query fail?

-- Modify the supplied T-SQL statement by including the TOP (100) PERCENT option. The query should look like this:

-- Execute the modified T-SQL statement. By applying the needed changes, you have altered the existing view.

-- Notice that you are still using the ORDER BY clause.

```
ALTER VIEW Production.ProductsBeverages AS
SELECT
    productid, productname, supplierid, unitprice, discontinued
FROM Production.Products
WHERE categoryid = 1
ORDER BY productname;
GO
```

Answer - Error: The ORDER BY clause is invalid in views, inline functions, derived tables, subqueries, and common table expressions, unless TOP, OFFSET or FOR XML is also specified.

VIEW does not take ORDER BY clause

```
ALTER VIEW Production.ProductsBeverages AS
SELECT TOP(100)
    productid, productname, supplierid, unitprice, discontinued
FROM Production.Products
WHERE categoryid = 1
ORDER BY productname;
GO
```

-- Task 4

-- The IT department has written a T-SQL statement that adds an additional calculated column to the view created in task 1.

-- Execute the provided query. What happened? What is the error message? Why did the query fail?

-- Apply the changes needed to get the T-SQL statement to execute properly.

```
ALTER VIEW Production.ProductsBeverages AS
SELECT
    productid, productname, supplierid, unitprice, discontinued,
    CASE WHEN unitprice > 100. THEN N'high' ELSE N'normal' END
FROM Production.Products
WHERE categoryid = 1;
GO
```

Answer - Error: Create View or Function failed because no column name was specified for column 6.

```
ALTER VIEW Production.ProductsBeverages AS
SELECT
    productid, productname, supplierid, unitprice, discontinued,
```

```
        CASE WHEN unitprice > 100. THEN N'high' ELSE N'normal' END AS unitpriceLevel
FROM Production.Products
WHERE categoryid = 1;
GO
```

-- Task 5

-- Remove the created view by executing the provided T-SQL statement. Execute this code exactly as written inside a query window.

```
IF OBJECT_ID(N'Production.ProductsBeverages', N'V') IS NOT NULL
    DROP VIEW Production.ProductsBeverages;
```