**Laboratoire 11: 5% de la note globale**

-- RÉINITIALISER TOUT UTILISATEUR ET CONNEXION

-- Revoquer tous les droits de l'utilisateur et supprimer tous les objets qui lui sont associes.

SETUSER

GO

USE BD\_Sales;

GO

REVOKE INSERT FROM user\_nom\_test

GO

REVOKE SELECT FROM user\_nom\_test

GO

DROP USER user\_nom\_test

GO

-- Vérifiez si le produit existe avant de le supprimer

IF EXISTS (SELECT 1 FROM [production].[products] WHERE [product\_id] = 400)

BEGIN

DELETE FROM [production].[products] WHERE [product\_id] = 400;

END

-- Supprimer la connexion (login) :

USE BD\_Sales

GO

DROP LOGIN login\_nom\_test

GO

-- VÉRIFIEZ que la réinitialisation fonctionne, LA SORTIE APRÈS TOUT UTILISATEUR ET CONNEXION

SELECT name, sid, create\_date, modify\_date, default\_database\_name, is\_disabled

FROM sys.sql\_logins

ORDER BY name;**Avant de commencer le Ouput**

-- vous donner une liste de tous les logins SQL sur votre instance SQL Server, triés par nom de login.

SELECT name, sid, create\_date, modify\_date, default\_database\_name, is\_disabled

FROM sys.sql\_logins

ORDER BY name;

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

1. Créer un login et l’associer à un utilisateur créé

-- 1) Creer un login

USE BD\_Sales

GO

CREATE LOGIN login\_nom\_test WITH PASSWORD=N'passwordTest123'

GO

-- Afficher output

-- vous donner une liste de tous les logins SQL sur votre instance SQL Server, triés par nom de login.

SELECT name, sid, create\_date, modify\_date, default\_database\_name, is\_disabled

FROM sys.sql\_logins

ORDER BY name;

Graphical user interface, application, Word

Description automatically generated

-- 1) Creer l'associer a un utilisateur cree

USE BD\_Sales;

GO

CREATE USER user\_nom\_test FOR LOGIN login\_nom\_test

GO

-- Afficher output

-- Cela affichera une liste de tous les utilisateurs, ainsi que leur type

-- (soit une connexion Windows, soit une connexion SQL) et leur mode d'authentification

-- (soit une authentification Windows, soit une authentification SQL Server).

-- Pour voir les utilisateurs de la base de données, vous pouvez exécuter une requête similaire sur la base de données spécifique :

SELECT name, type\_desc, authentication\_type\_desc FROM sys.database\_principals WHERE type\_desc IN ('WINDOWS\_USER', 'SQL\_USER');

Table

Description automatically generated

Graphical user interface

Description automatically generated

A picture containing timeline

Description automatically generated

* 1. Donner le droit d’insertion à l’utilisateur

USE BD\_Sales;

GO

GRANT INSERT ON DATABASE::BD\_SALES TO user\_nom\_test

GO

Graphical user interface, application

Description automatically generated

GRANT INSERT ON [production].[brands] TO user\_nom\_test;

GRANT INSERT ON [production].[categories] TO user\_nom\_test;

GRANT INSERT ON [production].[products] TO user\_nom\_test;

GRANT INSERT ON [production].[stocks] TO user\_nom\_test;

GRANT INSERT ON [sales].[customers] TO user\_nom\_test;

GRANT INSERT ON [sales].[order\_items] TO user\_nom\_test;

GRANT INSERT ON [sales].[orders] TO user\_nom\_test;

GRANT INSERT ON [sales].[staffs] TO user\_nom\_test;

GRANT INSERT ON [sales].[stores] TO user\_nom\_test;

Graphical user interface, application

Description automatically generated

GO

SET IDENTITY\_INSERT [production].[products] ON;

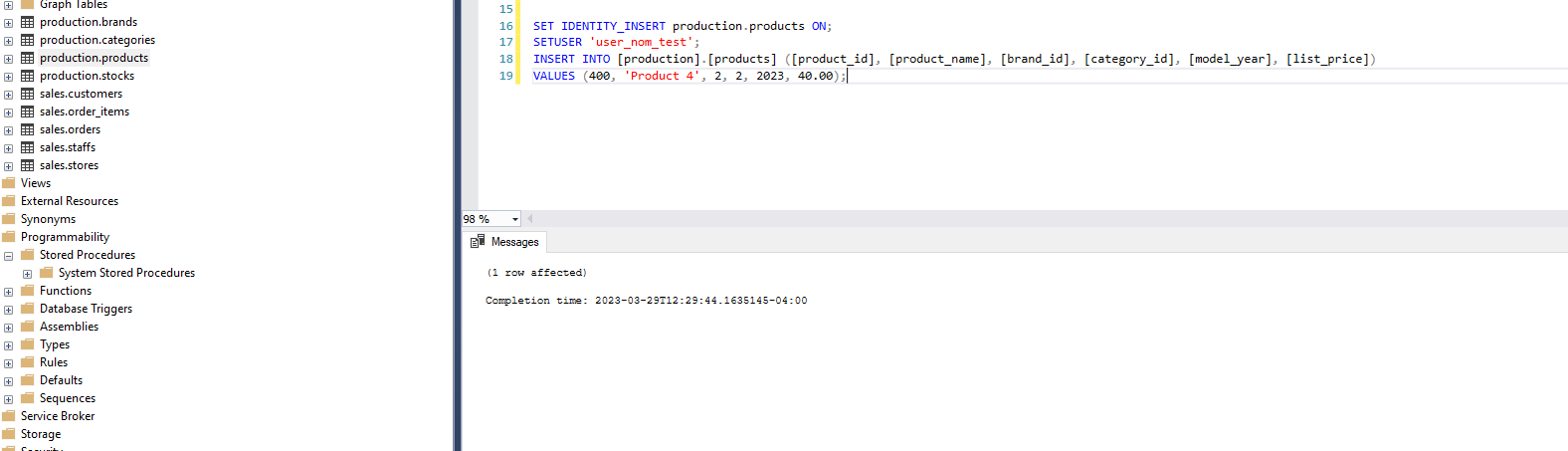
GO

SETUSER 'user\_nom\_test'

GO

INSERT INTO [production].[products] ([product\_id], [product\_name], [brand\_id], [category\_id], [model\_year], [list\_price])

VALUES (400, 'Product 4', 2, 2, 2023, 40.00);



Graphical user interface, application

Description automatically generated

* 1. Donner le droit de sélection à l’utilisateur

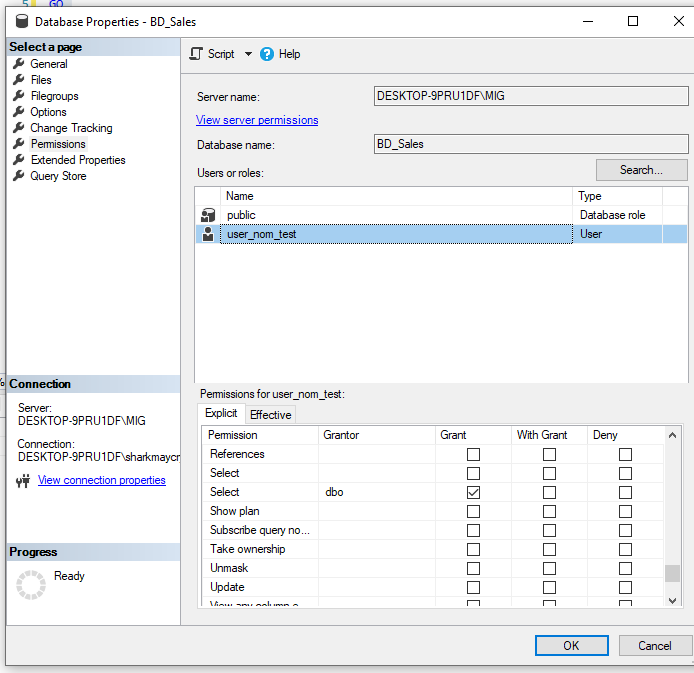
SETUSER

USE BD\_Sales;

GO

GRANT SELECT ON DATABASE::BD\_SALES TO user\_nom\_test

GO



GRANT SELECT ON [production].[brands] TO user\_nom\_test;

GRANT SELECT ON [production].[categories] TO user\_nom\_test;

GRANT SELECT ON [production].[products] TO user\_nom\_test;

GRANT SELECT ON [production].[stocks] TO user\_nom\_test;

GRANT SELECT ON [sales].[customers] TO user\_nom\_test;

GRANT SELECT ON [sales].[order\_items] TO user\_nom\_test;

GRANT SELECT ON [sales].[orders] TO user\_nom\_test;

GRANT SELECT ON [sales].[staffs] TO user\_nom\_test;

GRANT SELECT ON [sales].[stores] TO user\_nom\_test;

Graphical user interface, application

Description automatically generated

SETUSER 'user\_nom\_test';

SELECT \* FROM [production].[products] WHERE [product\_id] = 400;

Graphical user interface, text, application

Description automatically generated

* 1. Enlever le droit d’insertion à l’utilisateur

SETUSER

USE BD\_Sales;

GO

REVOKE INSERT ON DATABASE::BD\_SALES FROM user\_nom\_test;

GOGraphical user interface, application

Description automatically generated

REVOKE INSERT ON [production].[brands] TO user\_nom\_test;

REVOKE INSERT ON [production].[categories] TO user\_nom\_test;

REVOKE INSERT ON [production].[products] TO user\_nom\_test;

REVOKE INSERT ON [production].[stocks] TO user\_nom\_test;

REVOKE INSERT ON [sales].[customers] TO user\_nom\_test;

REVOKE INSERT ON [sales].[order\_items] TO user\_nom\_test;

REVOKE INSERT ON [sales].[orders] TO user\_nom\_test;

REVOKE INSERT ON [sales].[staffs] TO user\_nom\_test;

REVOKE INSERT ON [sales].[stores] TO user\_nom\_test;

GO

Graphical user interface, application

Description automatically generated

GO

SET IDENTITY\_INSERT [production].[products] ON;

GO

SETUSER 'user\_nom\_test'

GO

INSERT INTO [production].[products] ([product\_id], [product\_name], [brand\_id], [category\_id], [model\_year], [list\_price])

VALUES (410, 'Product 410', 2, 2, 2023, 410.00);

Graphical user interface, text, application

Description automatically generated

1. Créer une transaction permettant de supprimer les produits qui n’ont jamais été commandés

USE BD\_Sales;

PRINT @@TRANCOUNT;

BEGIN TRAN;

PRINT @@TRANCOUNT;

Graphical user interface, text, application, email

Description automatically generated

-- Supprimer la table temporaire si elle existe deja

IF OBJECT\_ID('tempdb..#products\_to\_delete') IS NOT NULL

DROP TABLE #products\_to\_delete;

-- Trouver les produits qui n'ont jamais été commandés

SELECT p.product\_id, p.product\_name

-- Cette transaction cree une table temporaire #products\_to\_delete pour stocker les identifiants des produits qui n ont jamais ete commandes

INTO #products\_to\_delete

FROM production.products p

LEFT JOIN sales.order\_items oi ON p.product\_id = oi.product\_id

WHERE oi.product\_id IS NULL

ORDER BY p.product\_id;

-- vérifier que la table a bien été remplie avec les identifiants de produits qui n'ont jamais été commandés

SELECT \* FROM #products\_to\_delete

ORDER BY product\_id ASC;

Graphical user interface, text, application

Description automatically generated

-- Supprimer les stocks correspondants

DELETE FROM production.stocks

WHERE product\_id IN (SELECT product\_id FROM #products\_to\_delete);

-- Supprimer les produits correspondants

DELETE FROM production.products

WHERE product\_id IN (SELECT product\_id FROM #products\_to\_delete);

-- Vérifier que les produits ont bien été supprimés

SELECT \* FROM production.products;

GO

SELECT \* FROM production.stocks;

Graphical user interface, application, table

Description automatically generated

-- Valider la transaction

COMMIT;

PRINT @@TRANCOUNT;

-- Annuler toutes les modifications apportées à la base de données depuis le début de la transaction en cours.

ROLLBACK;

PRINT @@TRANCOUNT;

-- Vérifier que les produits ont bien été, annuler pour toutes les modifications apportées

SELECT \* FROM production.products;

GO

SELECT \* FROM production.stocks;

Table

Description automatically generated

PRINT @@TRANCOUNT;

Graphical user interface, text, application, Teams

Description automatically generated

1. Créer une procédure stockée affichant le montant des commandes comprise entre 50 et 300.

spGetEachOrdersBetween50And300

USE BD\_Sales

GO

CREATE PROCEDURE spGetEachOrdersBetween50And300 AS

BEGIN

SELECT orders.order\_id, CAST(SUM(order\_items.quantity \* order\_items.list\_price \* (1 - order\_items.discount)) AS decimal(10, 2)) AS total\_amount

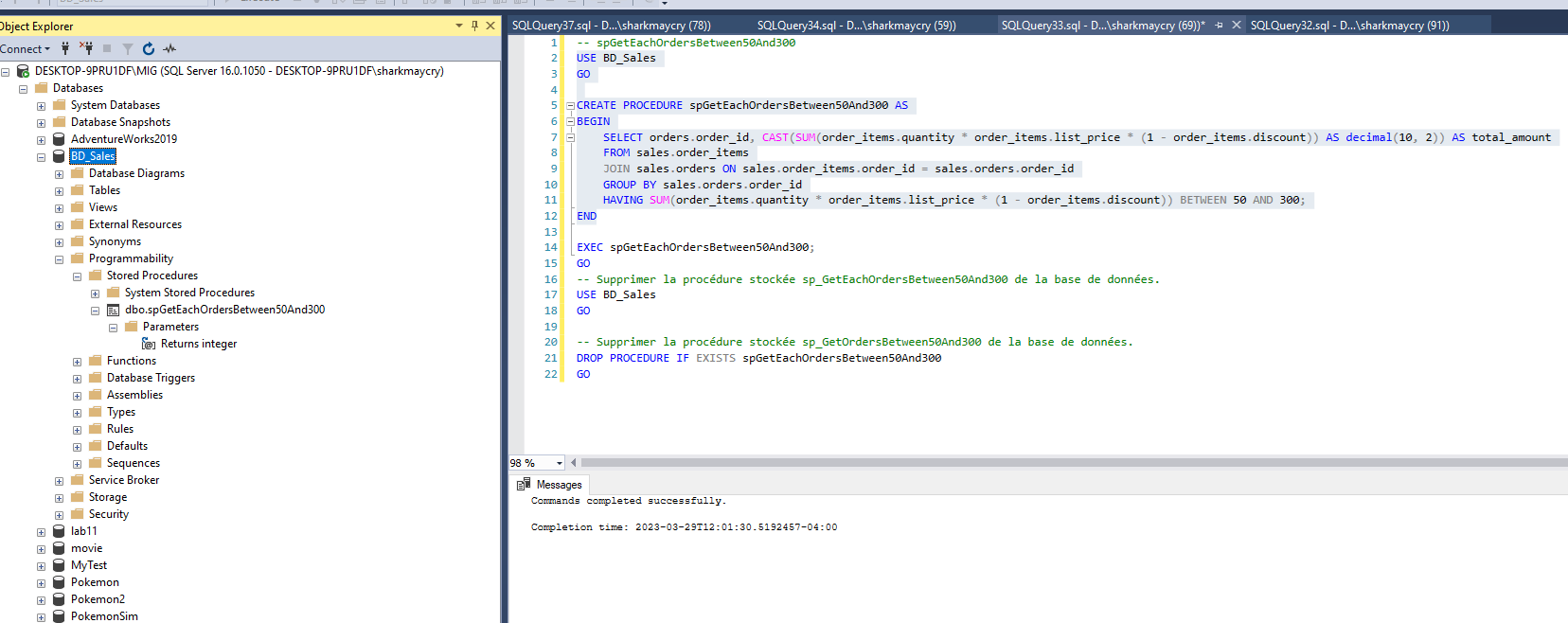
FROM sales.order\_items

JOIN sales.orders ON sales.order\_items.order\_id = sales.orders.order\_id

GROUP BY sales.orders.order\_id

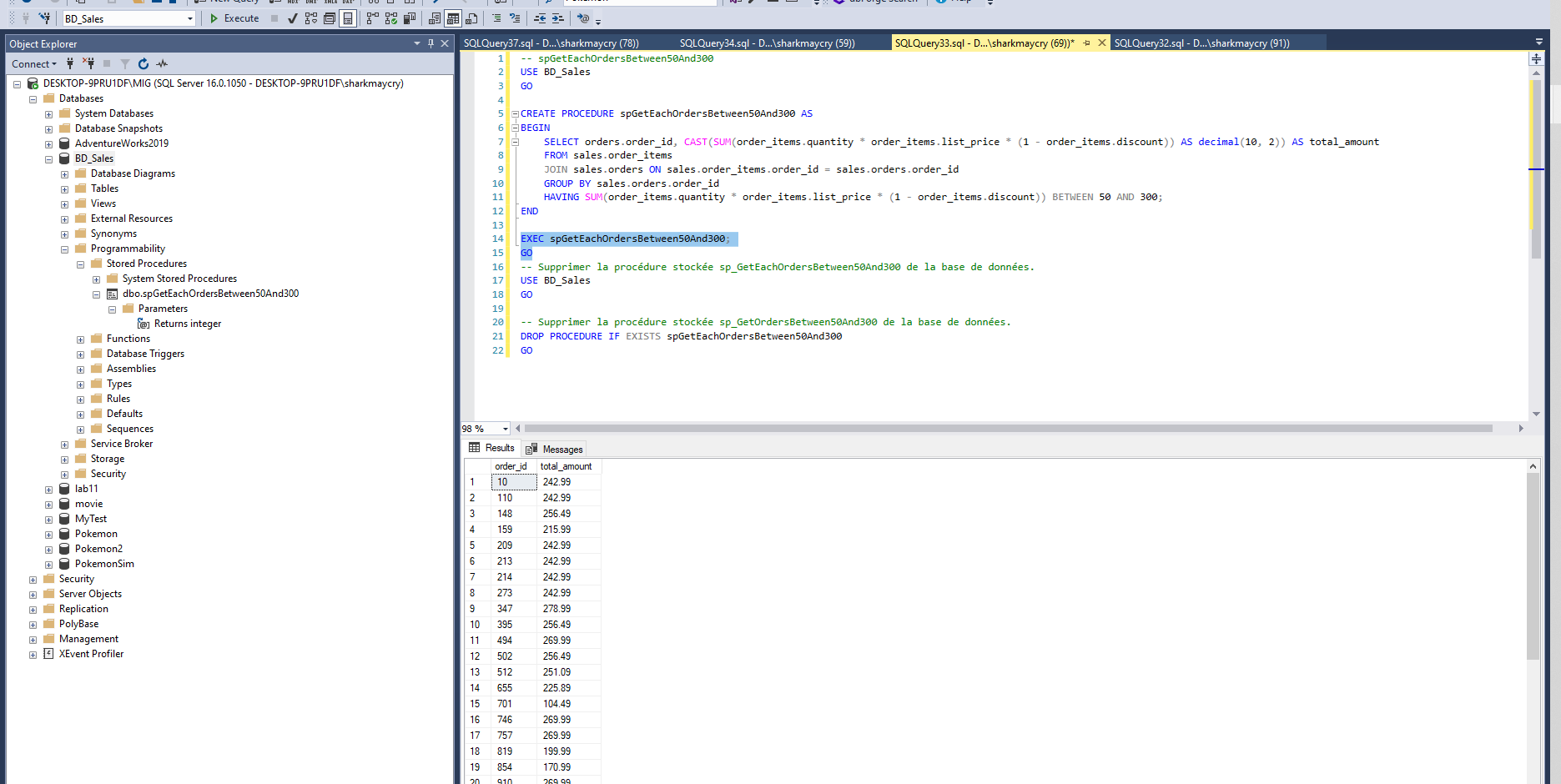
HAVING SUM(order\_items.quantity \* order\_items.list\_price \* (1 - order\_items.discount)) BETWEEN 50 AND 300;

END



EXEC spGetEachOrdersBetween50And300;

GO



-- Supprimer la procédure stockée sp\_GetEachOrdersBetween50And300 de la base de données.

USE BD\_Sales

GO

DROP PROCEDURE IF EXISTS spGetEachOrdersBetween50And300

GO

Graphical user interface, text, application

Description automatically generated

1. Créer une procédure stockée avec un paramètre de sortie qui calcule et retourne le chiffre d'affaires.

USE BD\_Sales;

GO

-- La procédure stockée prend deux dates en entrée

-- et calcule le chiffre d'affaires entre ces deux dates en interrogeant les tables des commandes et des articles.

-- La valeur calculée est stockée dans une variable de sortie @revenue de type DECIMAL(10, 2).

CREATE PROCEDURE CalculateSalesRevenue

@startDate DATE,

@endDate DATE,

@revenue DECIMAL(10, 2) OUTPUT

AS

BEGIN

SET NOCOUNT ON;

SELECT @revenue = SUM(oi.list\_price \* oi.quantity)

FROM sales.order\_items oi

JOIN sales.orders o ON oi.order\_id = o.order\_id

WHERE o.order\_date >= @startDate AND o.order\_date <= @endDate;

END;

GO

Graphical user interface, text, application

Description automatically generated

-- DECLARE @revenue DECIMAL(10, 2)

-- exécute la procédure stockée CalculateSalesRevenue en utilisant deux dates comme paramètres d'entrée

-- et stocke le chiffre d'affaires calculé dans la variable @revenue,

-- puis affiche la valeur stockée dans la variable @revenue

DECLARE @revenue DECIMAL(10, 2);

EXEC CalculateSalesRevenue '2016-01-01', '2016-01-31', @revenue OUTPUT;

SELECT "Chiffre d'affaires entre '2016-01-01' et '2016-01-31'" = @revenue;

Graphical user interface, text, application

Description automatically generated

-- Supprimer la procédure stockée CalculateSalesRevenue de la base de données.

DROP PROCEDURE IF EXISTS CalculateSalesRevenue;

Graphical user interface, text, application

Description automatically generated