A logo of a university of windsor

Description automatically generated

**University of Windsor**

Lab 1

COMP 8157 | Advanced Database Topics – Summer 2025

Names:

Aleena Ali Azeem - 110190830

Simranpreet Kaur - 110189426

Priyadharshan Reddy S - 110191285

Part 1

-- 1. Create a database named Priyadharshan\_Aleena\_Simran\_SalesDB

CREATE DATABASE Priyadharshan\_Aleena\_Simran\_SalesDB;

GO

A close-up of a text

AI-generated content may be incorrect.

-- 2. Create SalesData table

CREATE TABLE SalesData (

sale\_id INT PRIMARY KEY,

customer\_name VARCHAR(255),

customer\_email VARCHAR(255),

product\_name VARCHAR(255),

purchase\_date DATE,

quantity\_purchased INT,

sale\_amount DECIMAL(10, 2)

);

GO

A screenshot of a white background

AI-generated content may be incorrect.

-- 3. Populate the SalesData table with 100 rows of different sample data

DECLARE @saleCounter INT = 1;

BEGIN TRAN;

WHILE @saleCounter <= 100

BEGIN

INSERT INTO SalesData (

sale\_id,

customer\_name,

customer\_email,

product\_name,

purchase\_date,

quantity\_purchased,

sale\_amount

)

VALUES (

@saleCounter,

'Buyer\_' + CONVERT(VARCHAR, @saleCounter),

'buyer' + CONVERT(VARCHAR, @saleCounter) + '@domain.com',

'Item\_' + CONVERT(VARCHAR, (@saleCounter % 15 + 1)),

DATEADD(DAY, -@saleCounter, GETDATE()),

(@saleCounter % 7) + 1,

CONVERT(DECIMAL(10,2), @saleCounter \* 12.75)

);

SET @saleCounter = @saleCounter + 1;

END

COMMIT TRAN;

GO

A row of words on a white background

AI-generated content may be incorrect.

-- 4. Apply vertical partitioning

-- i. SalesInfo table

CREATE TABLE SalesInfo (

sale\_id INT FOREIGN KEY REFERENCES SalesData(sale\_id),

customer\_name VARCHAR(255),

customer\_email VARCHAR(255),

product\_name VARCHAR(255),

CONSTRAINT PK\_SalesInfo PRIMARY KEY CLUSTERED (sale\_id)

);

GO

-- Populate SalesInfo

INSERT INTO SalesInfo (sale\_id, customer\_name, customer\_email, product\_name)

SELECT sale\_id, customer\_name, customer\_email, product\_name

FROM SalesData;

GO

A white background with black text

AI-generated content may be incorrect.

-- ii. PurchaseDetails table

CREATE TABLE PurchaseDetails (

sale\_id INT FOREIGN KEY REFERENCES SalesData(sale\_id),

purchase\_date DATE,

quantity\_purchased INT,

sale\_amount DECIMAL(10, 2),

CONSTRAINT PK\_PurchaseDetails PRIMARY KEY CLUSTERED (sale\_id)

);

GO

-- Populate PurchaseDetails

INSERT INTO PurchaseDetails (sale\_id, purchase\_date, quantity\_purchased, sale\_amount)

SELECT sale\_id, purchase\_date, quantity\_purchased, sale\_amount

FROM SalesData;

GO

A screenshot of a computer

AI-generated content may be incorrect.

-- 5. Evaluate query performance (with statistics enabled)

SET STATISTICS IO ON;

SET STATISTICS TIME ON;

-- Example search for sale\_id like '%3%'

SELECT \* FROM SalesData WHERE sale\_id LIKE '%3%';

SELECT \* FROM SalesInfo WHERE sale\_id LIKE '%3%';

SELECT \* FROM PurchaseDetails WHERE sale\_id LIKE '%3%';

SET STATISTICS IO OFF;

SET STATISTICS TIME OFF;

GO

A white page with black text

AI-generated content may be incorrect.

-- 6. Create stored procedure GetSaleDetails

CREATE PROCEDURE GetSaleDetails

@SaleID INT

AS

BEGIN

SELECT

si.sale\_id,

si.customer\_name,

si.customer\_email,

si.product\_name,

pd.purchase\_date,

pd.quantity\_purchased,

pd.sale\_amount

FROM SalesInfo si

INNER JOIN PurchaseDetails pd ON si.sale\_id = pd.sale\_id

WHERE si.sale\_id = @SaleID;

END;

GO

A screenshot of a computer

AI-generated content may be incorrect.

-- 7. Execute GetSaleDetails for first 10 entries

DECLARE @loop INT = 1;

WHILE @loop <= 10

BEGIN

EXEC GetSaleDetails @SaleID = @loop;

SET @loop = @loop + 1;

END

GO

A row of words on a white background

AI-generated content may be incorrect.

-- 8. Query to count transactions with sale\_amount > 800

SELECT COUNT(\*) AS transactionThreshold

FROM PurchaseDetails

WHERE sale\_amount > 800;

GO

A screenshot of a message box

AI-generated content may be incorrect.

Part 2

-- STEP 1: Create the database

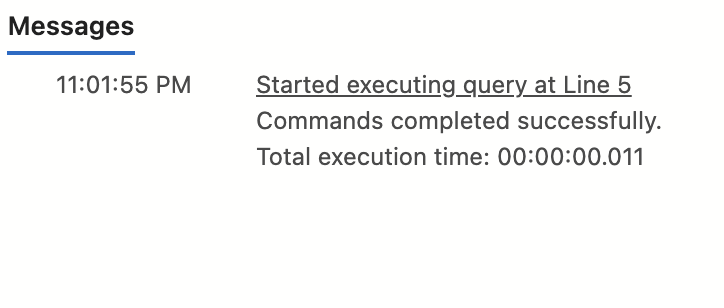
CREATE DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly;

A close-up of a text

AI-generated content may be incorrect.

-- Use the newly created database

USE Priyadharshan\_Aleena\_Simran\_Quarterly;



ALTER DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly ADD FILEGROUP [First\_Quarter];

ALTER DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly ADD FILEGROUP [Second\_Quarter];

ALTER DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly ADD FILEGROUP [Third\_Quarter];

ALTER DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly ADD FILEGROUP [Fourth\_Quarter];

-- STEP 3: Add physical files to the respective filegroups

ALTER DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly

ADD FILE (

NAME = First\_Quarter\_File,

FILENAME = '/var/opt/mssql/data/First\_Quarter.ndf',

SIZE = 5MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1MB

) TO FILEGROUP First\_Quarter;

ALTER DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly

ADD FILE (

NAME = Second\_Quarter\_File,

FILENAME = '/var/opt/mssql/data/Second\_Quarter.ndf',

SIZE = 5MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1MB

) TO FILEGROUP Second\_Quarter;

ALTER DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly

ADD FILE (

NAME = Third\_Quarter\_File,

FILENAME = '/var/opt/mssql/data/Third\_Quarter.ndf',

SIZE = 5MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1MB

) TO FILEGROUP Third\_Quarter;

ALTER DATABASE Priyadharshan\_Aleena\_Simran\_Quarterly

ADD FILE (

NAME = Fourth\_Quarter\_File,

FILENAME = '/var/opt/mssql/data/Fourth\_Quarter.ndf',

SIZE = 5MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1MB

) TO FILEGROUP Fourth\_Quarter;

A screenshot of a computer

AI-generated content may be incorrect.

-- STEP 4: Create partition function

CREATE PARTITION FUNCTION PriyadharshanByQuarter (INT)

AS RANGE LEFT FOR VALUES (1, 2, 3);

A close-up of a computer screen

AI-generated content may be incorrect.

-- STEP 5: Create partition scheme

CREATE PARTITION SCHEME PriyadharshanByQuarterADT

AS PARTITION PriyadharshanByQuarter

TO (First\_Quarter, Second\_Quarter, Third\_Quarter, Fourth\_Quarter);

A close-up of a text

AI-generated content may be incorrect.

-- STEP 6: Create partitioned table using the scheme

CREATE TABLE QuarterlyReports (

ReportID INT IDENTITY(1,1) NOT NULL,

ReportQuarter INT NOT NULL,

QuarterlyReport VARCHAR(MAX),

PRIMARY KEY (ReportID, ReportQuarter)

) ON PriyadharshanByQuarterADT(ReportQuarter);

A screenshot of a computer

AI-generated content may be incorrect.

-- STEP 7: Insert 100000 records randomly

DECLARE @i INT = 1;

WHILE @i <= 100000

BEGIN

INSERT INTO QuarterlyReports (ReportQuarter, QuarterlyReport)

VALUES (

1 + ABS(CHECKSUM(NEWID()) % 4), -- Random quarter between 1 and 4

REPLICATE('Priyadharshan', 100) -- Long dummy report

);

SET @i += 1;

END

A row of words on a white background

AI-generated content may be incorrect.

-- STEP 8: Count number of records in each partition

SELECT

$PARTITION.PriyadharshanByQuarter(ReportQuarter) AS PartitionNumber,

COUNT(\*) AS RecordCount

FROM QuarterlyReports

GROUP BY $PARTITION.PriyadharshanByQuarter(ReportQuarter);

A screenshot of a computer

AI-generated content may be incorrect.

STEP 9: Fetch records from partition 2 (Second Quarter)

SELECT \*

FROM QuarterlyReports

WHERE $PARTITION.PriyadharshanByQuarter(ReportQuarter) = 2;

A screenshot of a data

AI-generated content may be incorrect.

A document with text and images

AI-generated content may be incorrect.