**LAB 2**

Author:

|  |  |
| --- | --- |
| **Abhirup Ranjan** | **110091866** |

Submitted To:

|  |
| --- |
| **Professor** |
| **Dr. Shafaq Khan** |

**Table of Contents**

[1. Part 1: 2](#_Toc136940950)

[2. Part 2 10](#_Toc136940951)

# **Part 1:**

1. **Create a database <yourfirstname>vertical.**

CREATE DATABASE AbhirupVertical;

A screenshot of a computer

Description automatically generated with low confidence

1. **Create a table “Product” table with the following columns: id, name, description, price, category, brand, and quantity. (Note: Insert 10 rows of data in this table)**

CREATE TABLE Product

(

ID int IDENTITY (1,1) NOT NULL,

Name varchar (50),

Description varchar (100),

Price decimal(10,2),

Category varchar (20),

Brand varchar (20),

Quantity int

CONSTRAINT Product\_PK PRIMARY KEY CLUSTERED (Id)

)

A screenshot of a computer

Description automatically generated with medium confidence

INSERT INTO dbo.Product(Name,Description, Price, Category, Brand, Quantity)

VALUES

('Name1','Des1', 10.00, 'Category1', 'BrandName1', 10),

('Name2','Des2', 20.0, 'Category2', 'BrandName2', 20),

('Name3','Des3', 30.3, 'Category3', 'BrandName3', 30),

('Name4','Des4', 40.4, 'Category4', 'BrandName4', 40),

('Name5','Des5', 50.5, 'Category5', 'BrandName5', 50),

('Name6','Des6', 60.6, 'Category6', 'BrandName6', 60),

('Name7','Des7', 70.7, 'Category7', 'BrandName7', 70),

('Name8','Des8', 80.8, 'Category8', 'BrandName8', 80),

('Name9','Des9', 90.9, 'Category9', 'BrandName9', 90),

('Name10','Des10', 100.11, 'Category10', 'BrandName10', 10);

A screenshot of a computer

Description automatically generated with medium confidence

SELECT \* FROM Product;

A screenshot of a computer

Description automatically generated with medium confidence

1. Apply vertical partitioning by dividing the above table into two partition tables:

* “ProductBasic” table (Columns: id, name, description and category).

CREATE TABLE ProductBasic

(

Id int FOREIGN KEY REFERENCES Product (Id),

Name varchar(50),

Description varchar (100),

Category varchar (20),

CONSTRAINT PK\_ProductBasic PRIMARY KEY CLUSTERED (Id)

)

A screenshot of a computer

Description automatically generated with low confidence

INSERT INTO ProductBasic(ID, Name, Description, Category)

VALUES

(1,'Name1','Des1', 'Category1'),

(2,'Name2','Des2', 'Category2'),

(3,'Name3','Des3', 'Category3'),

(4,'Name4','Des4', 'Category1'),

(5,'Name5','Des5', 'Category2'),

(6,'Name6','Des6', 'Category3'),

(7,'Name7','Des7', 'Category1'),

(8,'Name8','Des8', 'Category2'),

(9,'Name9','Des9', 'Category3'),

(10,'Name10','Des10', 'Category1');

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

SELECT \* FROM ProductBasic;

A screenshot of a computer

Description automatically generated with medium confidence

* “ProductDetails” table (Columns: id, price, brand, quantity)

CREATE TABLE ProductDetails

(

ID int NOT NULL,

Price decimal(10,2),

Brand varchar (20),

Quantity int

CONSTRAINT PK\_ProductDetails PRIMARY KEY CLUSTERED (Id)

)

A picture containing text, screenshot, font

Description automatically generated

INSERT INTO ProductDetails(ID,Price, Brand, Quantity)

VALUES

(1,10.00,'BrandName1', 10),

(2,20.01,'BrandName2', 20),

(3,30.03,'BrandName3', 30),

(4,40.04,'BrandName4', 40),

(5,50.05,'BrandName5', 50),

(6,60.06,'BrandName6', 60),

(7,70.07,'BrandName7', 70),

(8,80.08,'BrandName8', 80),

(9,90.09,'BrandName9', 90),

(10,100.11,'BrandName10', 100);

SELECT \* FROM ProductDetails;

A screenshot of a computer

Description automatically generated with medium confidence

1. Calculate the query performance of each table by retrieving the same ‘id’ from three tables.

SET STATISTICS IO ON

SET STATISTICS TIME ON

SELECT ID

FROM Product

WHERE ID = 1

SET STATISTICS IO OFF

SET STATISTICS TIME OFF

A screenshot of a chat

Description automatically generated with medium confidence

A picture containing text, receipt, screenshot, algebra

Description automatically generated

SET STATISTICS IO ON

SET STATISTICS TIME ON

SELECT ID

FROM ProductBasic

WHERE ID = 1

SET STATISTICS IO OFF

SET STATISTICS TIME OFF

A screenshot of a computer

Description automatically generated with low confidence

SET STATISTICS IO ON

SET STATISTICS TIME ON

SELECT ID

FROM ProductDetails

WHERE ID = 1

SET STATISTICS IO OFF

SET STATISTICS TIME OFF

A screenshot of a computer

Description automatically generated with low confidence

1. Retrieve basic information of all products in a specific category from the “ProductBasic” table.

SELECT \* FROM ProductBasic

WHERE

Category = 'Category1'

A screenshot of a computer

Description automatically generated with medium confidence

1. Retrieve the price and brand of a specific product from the “ProductDetails” table.

SELECT Price, Brand FROM ProductDetails

WHERE

Brand = 'BrandName1'

A screenshot of a computer

Description automatically generated with low confidence

# **Part 2**

1. Create a database <yourfirstname>horizontal.

CREATE DATABASE AbhirupHorizontal

A picture containing text, font, screenshot

Description automatically generated

1. Create a table “Birthday” table with the following columns: s.no, name, date, month (01 - 06) and year. (Note: Insert 20 rows of data in this table).

CREATE TABLE Birthday

(

SNo int IDENTITY (1,1) NOT NULL,

Name varchar (50),

Date int,

Month int,

Year int,

CONSTRAINT Birthday\_PK PRIMARY KEY CLUSTERED (SNo)

);

A screenshot of a computer

Description automatically generated with low confidence

INSERT INTO Birthday(Name, Date, Month, Year)

VALUES

('Abhirup', 26, 3, 1994),

('Rohit', 1, 4, 1993),

('Rat', 2, 5, 1992),

('A', 3, 6, 1991),

('B', 4, 7, 1990),

('C', 5, 8, 1989),

('D', 6, 9, 1988),

('E', 7, 10, 1987),

('F', 8, 11, 1986),

('G', 9, 12, 1985),

('H', 10, 1, 1984),

('I', 11, 2, 1983),

('J', 12, 3, 1982),

('K', 13, 4, 1981),

('L', 14, 5, 1994),

('M', 15, 6, 1995),

('N', 16, 7, 1996),

('O', 17, 8, 1997),

('P', 18, 9, 1998),

('Q', 19, 10, 1999);

A screenshot of a computer

Description automatically generated with medium confidence

SELECT \* FROM Birthday

A screenshot of a computer screen

Description automatically generated with medium confidence

3) Create filegroups within the database to divide them by month.

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP January;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP February;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP March;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP April;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP May;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP June;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP July;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP August;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP September;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP October;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP November;

ALTER DATABASE AbhirupHorizontal ADD FILEGROUP December;

A screenshot of a computer

Description automatically generated with medium confidence

**Show the created file Groups.**

SELECT data\_space\_id, name AS AvailableFilegroups, type\_desc FROM sys.filegroups

WHERE type = 'FG'

A screenshot of a computer

Description automatically generated

Assigning physical storage location on local machine

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Jan],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_1.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [January]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Feb],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_2.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [February]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Mar],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_3.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [March]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Apr],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_4.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [April]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [May],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_5.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [May]

ALTER DATABASE [Rohithorizontal]

ADD FILE

(

NAME = [Jun],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_6.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [June]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Jul],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_7.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [July]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Aug],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_8.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [August]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Sep],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_9.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [September]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Oct],

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\Rohithorizontal\_10.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [October]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Nov],

FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_11.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

) TO FILEGROUP [November]

ALTER DATABASE [AbhirupHorizontal]

ADD FILE

(

NAME = [Dec],

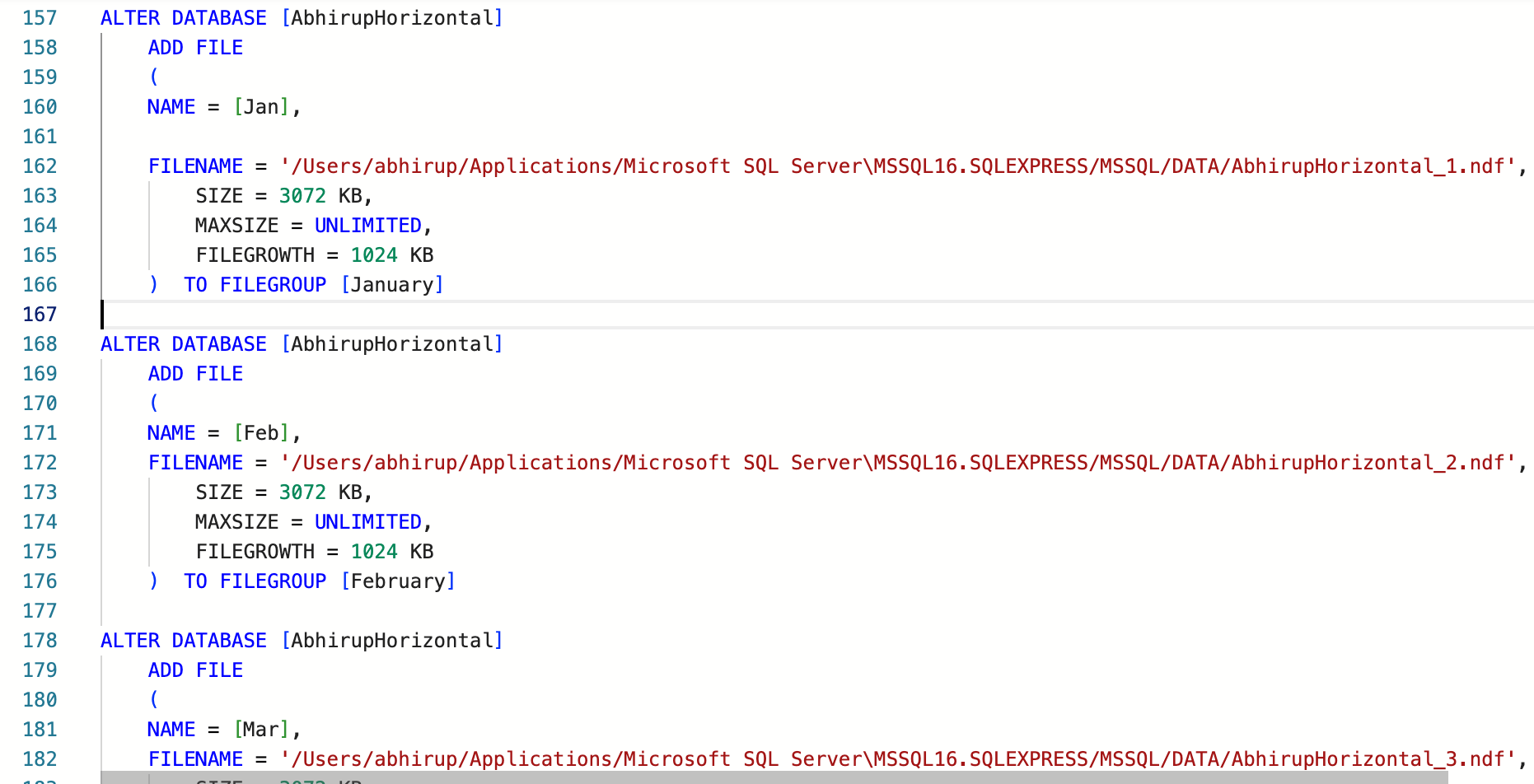
FILENAME = '/Users/abhirup/Applications/Microsoft SQL Server\MSSQL16.SQLEXPRESS/MSSQL/DATA/AbhirupHorizontal\_12.ndf',

SIZE = 3072 KB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024 KB

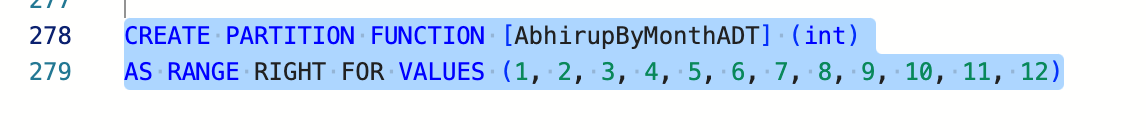
) TO FILEGROUP [December]



4) Create a partition function <yourfirstname>ByMonth (Note: The datatype of the month to be integer).

CREATE PARTITION FUNCTION [AbhirupByMonthADT] (int)

AS RANGE RIGHT FOR VALUES (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)



5) Create a partition scheme <yourfirstname>ByMonthADT.

CREATE PARTITION SCHEME AbhirupByMonthADT

AS PARTITION AbhirupByMonth

TO ( FG\_JAN,

FG\_JAN,

FG\_FEB,

FG\_MAR,

FG\_APR,

FG\_MAY,

FG\_JUN);

A screen shot of a computer

Description automatically generated with low confidence

6) create or modify a table and specify the partition scheme as the storage location to segment the data out and store it within the appropriate file group.

CREATE TABLE BirthdayMonth

(Id int IDENTITY (1,1) NOT NULL,

BirthdayMonth int NOT NULL,

PRIMARY KEY (BirthdayMonth, Id))

ON AbhirupByMonthADT(BirthdayMonth);

A picture containing text, screenshot, font, electric blue

Description automatically generated

INSERT BirthdayMonth VALUES

(1),

(2),

(3),

(4),

(5),

(5),

(6),

(7),

(8),

(9),

(10),

(11),

(12);

SELECT \* from BirthdayMonth

1. Write a query to check the number of records in each partition.

SELECT $PARTITION.AbhirupByMonthADT(BirthdayMonth) AS Partition, COUNT(\*) AS [COUNT] FROM BirthdayMonth

GROUP BY $PARTITION.AbhirupByMonthADT(BirthdayMonth) ORDER BY Partition;

A picture containing text, screenshot, font, line

Description automatically generated

8) Execute the records in partition number 3.

SELECT \* FROM BirthdayMonth

WHERE $PARTITION.AbhirupByMonthADT(BirthdayMonth) = 3

A screenshot of a computer

Description automatically generated with low confidence