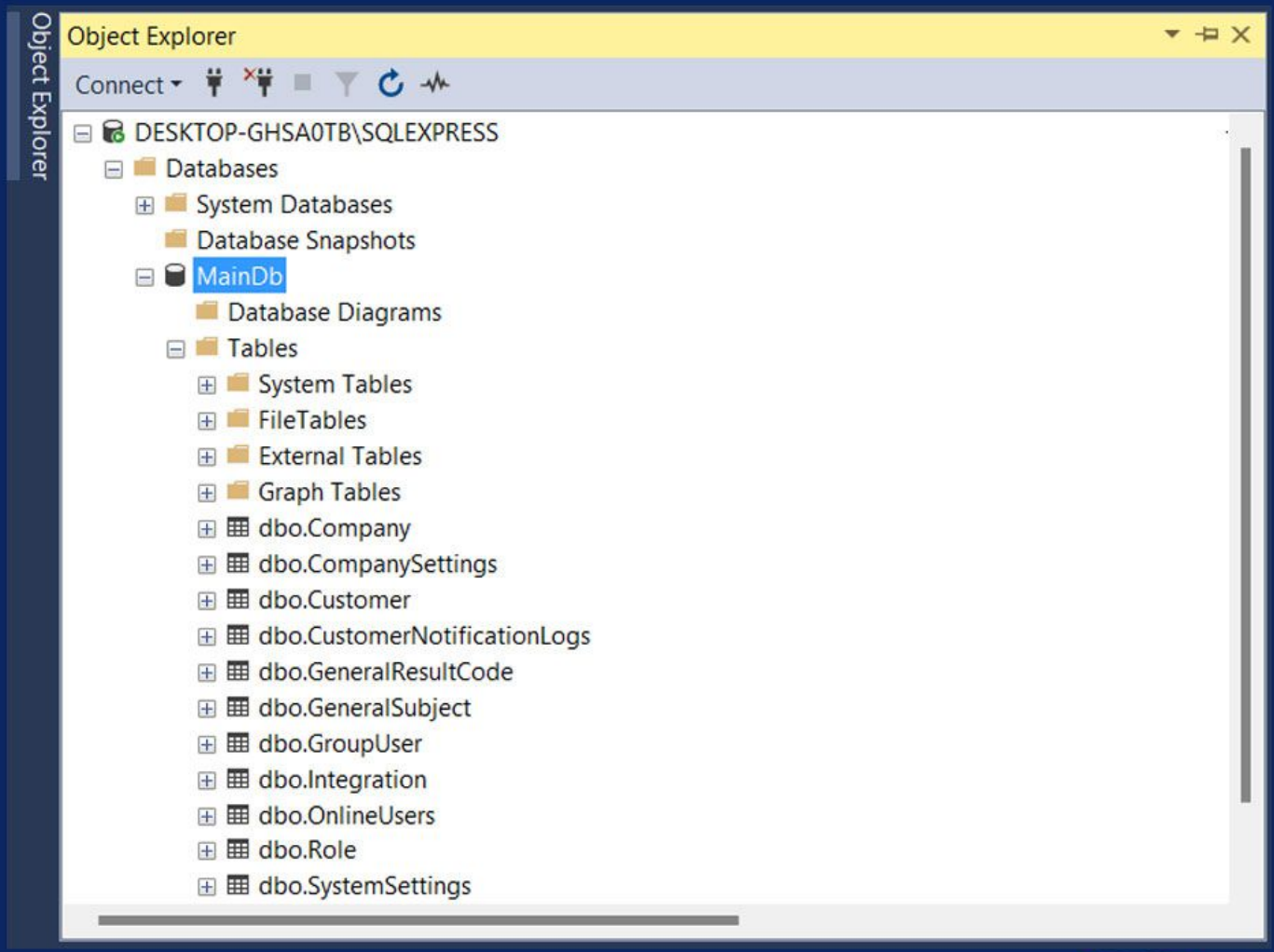
# TSQL ile Dinamik Bölümlenmiş Veritabanı Mimarisi Nasıl Oluşturulur?

TSQL ile dinamik bölümlenmiş veritabanı mimarisi kurarak, anlık verinin işlenmesinin önemli olduğu sistemlerde yüksek performans sağlanabilmektedir.

**Peki bu sistemler nelerdir?**

- Toplu SMS
- Chat Destek Sistemleri
- Otoyol HGS Sistemleri
- Cihaz Anlık Durum İzleme Sistemleri

# 1- Tüm müşterilerimizi ilgilendiren genel tablolar, stored procedure ve functionları ana veritabanımızda bulunduruyoruz.




# 2- Şirket tanımlaması yaparken şirkete özel veritabanı oluşturuyoruz.

Şirkete özel veritabanı oluştururken MainDb altında oluşturduğumuz Insert_DatabaseCreate StoredProcedure'ünü kullanıyoruz.

```
BEGIN

    Declare @Result nvarchar(100)
    Declare @YearMonth nvarchar(20)= Convert(nvarchar,Replace((convert(varchar(7), Convert(date,GETDATE()), 126)),'-','_'))
    IF(EXISTS(SELECT * FROM Company WITH(NOLOCK) WHERE CompanyName=@CompanyName and Status=1))
        BEGIN
            SET @Result = 'Bu isme ait şirket bulunmaktadır.'
        END
    ELSE
        BEGIN
            Insert Into Company(CompanyName, CreateDate, SiteUrl,EmailAddress, Phone, Status, DatabaseName)
                        select @CompanyName,GETDATE(),@SiteUrl,@EmailAddress,@Phone,@Status,@DatabaseName

            IF(@@ROWCOUNT > 0)
                BEGIN
                    DECLARE @ID int=SCOPE_IDENTITY()
                    INSERT INTO CompanySettings(CompanyID,AssigAutomaticCallToUser,MaximumChatOwnership)
                    SELECT @ID,@AssigAutomaticCallToUser,@MaximumChatOwnership

                    exec insert_DatabaseCreate @DatabaseName,@YearMonth

                    SET @Result ='1'
                END
            ELSE
                BEGIN
                    SET @Result= 'İşlem esnasında bir hata oluştu.'
                END
        END
        SELECT @Result
END
```

# 3- DatabaseCreate Scriptimiz

Şirkete özel veritabanı oluştururken MainDb altında oluşturduğumuz Insert_DatabaseCreate StoredProcedure'ünü kullanıyoruz.
Burada örnek olarak chat sisteminde müşteri mesajlarını aylık dizinlere böleceğimiz örnek yapıyı ele alıyoruz.
Tabloyu oluştururken aynı zamanda dinamik olarak indexlerimizi de oluşturmalıyız.

```
USE [MainDb]
GO
ALTER proc [dbo].[insert_DatabaseCreate]
    @DataBaseName nvarchar(100),
    @YearMonth nvarchar(20)
as
--exec insert_DatabaseCreate 'COMPANY_A_DB','2023_03'
begin
IF DB_ID(@DataBaseName) IS NULL
BEGIN
   exec('CREATE DATABASE '+@DataBaseName)
END
Declare @query nvarchar(max)=''
IF OBJECT_ID(@DataBaseName+'..CustomerMessage_'+@YearMonth) is null
Begin
    SET @query='CREATE TABLE '+@DataBaseName+'.[dbo].[CustomerMessage_'+@YearMonth+'](
        [ID] [bigint] IDENTITY(1,1) NOT NULL,
        [CustomerID] [bigint] NOT NULL,
        [TokenID] [bigint] NOT NULL,
        [Message] [nvarchar](max) NOT NULL,
        [TransactionDate] [datetime2](7) NOT NULL,
        [IsIncomingMessage] [bit] NOT NULL,
        [AgentID] [int] NULL,
        [ContentTypeID] [int] NULL,
        [BotID] [int] NULL,
        [BotContentID] [int] NULL,
        [BotParameter] [nvarchar](100) NULL,
        [ResponseTime] [time](7) NULL,
     CONSTRAINT [PK_CustomerMessage_'+@YearMonth +'] PRIMARY KEY CLUSTERED
    (
        [ID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
    CREATE INDEX [IX_CustomerMessage_'+@YearMonth +'_TokenID_TransactionDate] ON  '+@DataBaseName+'.[dbo].[CustomerMessage_'+@YearMonth +'] ([TokenID],[TransactionDate])
CREATE INDEX [IX_CustomerMessage_'+@YearMonth +'_TokenID] ON  '+@DataBaseName+'.[dbo].[CustomerMessage_'+@YearMonth +'] ([TokenID]) INCLUDE ([ID], [CustomerID], [Message], [TransactionDate])
CREATE INDEX [IX_CustomerMessage_'+@YearMonth +'_TokenID_ResponseTime] ON  '+@DataBaseName+'.[dbo].[CustomerMessage_'+@YearMonth +'] ([TokenID],[ResponseTime])'
    exec(@query)
End
End
```

# 4- Create Token Scripti

Yine örnek olarak chat sisteminde müşteriden gelen ilk talepte token açıyoruz.
Token açarken, işlem zamanına bakıp ilgili aya ait CustomerToken tablomuzun olup olmadığını kontrol ediyoruz. Eğer tablomuz bulunmuyor ise InsertDatabaseCreate procedure ile tablomuzu oluşturuyoruz.

```
USE [MainDb]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROC [dbo].[CreateToken]
(
@CompanyID int,
@CustomerID bigint,
@Name nvarchar(150),
@GsmNo nvarchar(20),
@Email nvarchar(100)
)
AS
--exec CreateToken 1,null,null,'Onur Özertürk','905440000000','','Whatsapp','',null
BEGIN
    Declare @DataBase nvarchar(100)
    Declare @TokenID bigint=0
    Declare @query nvarchar(max)=''

    Declare @YearMonth nvarchar(20)=CAST(Year(GETDATE()) as varchar)+'_'+ CAST(RIGHT('0' + RTRIM(MONTH(GETDATE())), 2) as varchar)
    IF OBJECT_ID(@DataBase+'..CustomerToken_'+@YearMonth) is null
        exec  [dbo].[insert_DatabaseCreate] @Database,@YearMonth

    if(@CustomerID>0 )
    BEGIN
        if(@TokenID=0 or @TokenID is null)
        Begin
            SET @query='INSERT INTO '+@DataBase+'..CustomerToken_'+FORMAT(GetDate(),'yyyy_MM')+' (CustomerID,TransactionDate,Status,SubjectID,ChatTypeID, AgentID)
                        SELECT '+Cast(@CustomerID as varchar)+','''+Convert(varchar,GETDATE(),121)+''',0,1,1,null'
            EXEC sp_ExecuteSql @query
        END
    END
END
```

# 5- Bölümlenmiş Veritabanı Mimarisinde Raporlama

Aylık olarak bölümlenmiş tablolarımızda iki tarih arasında rapor almak istediğimizde bu iki tarih arasında kaç adet tablomuzun bulunduğunu belirleyip sorgumuzu ilgili tabloları aşağıdaki örnek scriptimizde olduğu gibi dahil ediyoruz.

```
While(@CheckRow<=@HowManyMonthsToCheck)
Begin
    Set @CheckRow=@CheckRow+1
    if(LEN(@query)>0)
        SET @query+=' UNION ALL '
    SET @YearMonth= Convert(nvarchar,Replace((convert(varchar(7), Convert(date,@TmpStartDate), 126)),'-','_'))
    if(@FDYearMonth = @YearMonth)
        SET @HowManyMonthsToCheck=0
    if(@CheckRow>1)
        SET @TmpStartDate= CAST(convert(datetime2,convert(date,Convert(date,@TmpStartDate), 102),121) as varchar(35))
    if(@HowManyMonthsToCheck=0)
        SET @TmpFinishDate=CAST(convert(datetime2,CAST(convert(date,Cast(@FinishDate as varchar(10)), 102) as varchar(10))
                           +' 23:59:59.0000000',121) as varchar(35))
    Else
        SET @TmpFinishDate=CAST(convert(datetime2,CAST(convert(date,DATEADD(d, -1, DATEADD(m, DATEDIFF(m, 0, Cast(@TmpStartDate as varchar(10))) + 1, 0)), 102) as varchar(10))
                           +' 23:59:59.0000000',121) as varchar(35))
    IF OBJECT_ID(@DatabaseName+'.dbo.MessageResultReport_'+@YearMonth) is not null
    Begin
        SET @query+=' Select distinct mrr.TokenID as ID,u.Name+'' ''+u.Surname as [Agent],mrr.CustomerID, c.Name as [CustomerName]
        from '+@DatabaseName+'.dbo.MessageResultReport_'+@YearMonth +' as mrr with(nolock)
        LEFT JOIN [User] as u with(nolock) on mrr.AgentID=u.ID
        INNER JOIN Customer as c with(nolock) on c.ID=mrr.CustomerID
        LEFT JOIN '+@DatabaseName+'.dbo.CustomerToken_'+@YearMonth +' as ct with(nolock) on ct.CustomerID=mrr.CustomerID
                                    and ct.TransactionDate between '''+@TmpStartDate+''' and '''+@TmpFinishDate+'''
        LEFT JOIN GeneralResultCode as grc with(nolock) on grc.ID=mrr.ResultCodeID '

        SET @query+=' WHERE mrr.ReportDate between '''+@TmpStartDate+''' and '''+@TmpFinishDate+''' '

        if(LEN(@SearchText)>0 and @SearchText  is not null and @SearchText<>'null')
        Begin
            if(LEN(@SearchColumn)>0 and @SearchColumn  is not null)
                SET @query+=' and '+@SearchColumn+' like ''%'+@SearchText+'%'' '
        End
    End
    SET @TmpStartDate=DATEADD(SECOND,1,Convert(datetime2, @TmpFinishDate))
    SET @TmpFinishDate=DATEADD(SECOND,1,Convert(datetime2, @TmpFinishDate))
End
if(LEN(@query)>0)
Begin
    SET @query2='Select count(DISTINCT y.CustomerID) CustomerCount, sum(y.TotalMessageCount) TotalMessageCount,
                        avg(y.FirstReplyMinute) FirstReplyMinute,avg(y.DurationMinute) DurationMinute
                        from(Select x.* from ('+@query+') as x) as y '
End
exec(@query2)
```

# 6-Dinamik Şirket Veritabanı Yapısı