

بسم تعالی

نام و نام خانوادگی : سهیل نعمت الهی

موضوع : Impact of Database Design on Application Performance

دانشکده : ملی و مهارت آیت الله خامنه ای میناب

درس : آزمایشگاه پایگاه داده 2

استاد : احمد زاده

## تأثیر طراحی پایگاه داده بر عملکرد برنامه

### تحلیل تأثیر روش‌های مختلف طراحی پایگاه داده بر عملکرد برنامه در استفاده از MS SQL Server

طراحی پایگاه داده یکی از مهم‌ترین عوامل در توسعه سیستم‌های نرم‌افزاری است که تأثیر زیادی بر عملکرد و کارایی برنامه‌ها دارد. در این تحقیق، به بررسی تأثیر روش‌های مختلف طراحی پایگاه داده بر عملکرد برنامه‌ها در محیط MS SQL Server پرداخته می‌شود. به‌عنوان یکی از پایگاه‌های داده محبوب در میان توسعه‌دهندگان، با قابلیت‌هایی مانند بهینه‌سازی کوئری‌ها، ایندکس‌گذاری و پردازش تراکنش‌ها ابزارهای مختلفی برای مدیریت داده‌ها ارائه می‌دهد.

#### ۱. ساختار پایگاه داده و تأثیر آن بر عملکرد

یکی از مهم‌ترین جنبه‌های طراحی پایگاه داده، انتخاب ساختار مناسب برای ذخیره‌سازی داده‌ها است. انتخاب بین پایگاه داده‌های رابطه‌ای (Relational Database) یا غیررابطه‌ای (NoSQL) می‌تواند تأثیر قابل توجهی بر عملکرد برنامه داشته باشد. در MS SQL Server، انتخاب جداول، کلیدهای اصلی و خارجی و ساختارهای ایندکس می‌تواند به‌طور مستقیم بر سرعت اجرای کوئری‌ها، زمان پاسخ‌دهی و مصرف منابع تأثیر بگذارد.

```
CREATE CLUSTERED INDEX idx_clustered ON MyTable(Column1);
```

```
CREATE NONCLUSTERED INDEX idx_nonclustered ON MyTable(Column2);
```

## ۲. ایندکس گذاری و تأثیر آن بر سرعت جستجو

ایندکس ها ابزارهای مهمی برای بهینه سازی جستجوها در پایگاه داده هستند. طراحی صحیح ایندکس ها می تواند زمان پاسخ دهی کوئری ها را به طور قابل توجهی کاهش دهد. در **MS SQL Server** ، انتخاب ایندکس های مناسب و به روز نگه داشتن آن ها می تواند به سرعت پردازش درخواست ها کمک کند. همچنین، ایندکس های ترکیبی و ایندکس های غیررابطه ای (**Non-Clustered Indexes**) می توانند عملکرد بهتری در جستجوهای پیچیده داشته باشند.

## ۳. نرمال سازی و دنرمال سازی

در فرآیند طراحی پایگاه داده، نرمال سازی (**Normalization**) و دنرمال سازی (**Denormalization**) دو روش رایج هستند که تأثیر زیادی بر عملکرد دارند. نرمال سازی با هدف کاهش افزونگی داده ها و بهبود یکپارچگی می تواند منجر به پیچیدگی های بیشتر در کوئری ها شود، زیرا به جداول متعدد نیاز دارد. از طرف دیگر، دنرمال سازی ممکن است سرعت کوئری ها را افزایش دهد، اما ممکن است به از دست دادن یکپارچگی داده ها و افزایش افزونگی منجر شود. در **MS SQL Server** ، تصمیم گیری بین این دو روش بستگی به نوع کاربرد و نیاز به تعادل بین کارایی و یکپارچگی داده ها دارد.

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    CustomerName NVARCHAR(100),  
    ProductName NVARCHAR(100),  
    OrderDate DATE  
);
```

```
CREATE TABLE Customers (  
    CustomerID INT PRIMARY KEY,  
    CustomerName NVARCHAR(100)  
);
```

```
CREATE TABLE Products (  
    ProductID INT PRIMARY KEY,  
    ProductName NVARCHAR(100)  
);
```

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID),  
    ProductID INT FOREIGN KEY REFERENCES Products(ProductID),  
    OrderDate DATE  
);
```

جدول دنرمال شده :

```
CREATE TABLE OrdersSummary (  
    OrderID INT PRIMARY KEY,  
    CustomerName NVARCHAR(100),  
    ProductName NVARCHAR(100),  
    OrderDate DATE,  
    TotalAmount DECIMAL(10, 2)  
);
```

#### ۴. استفاده از تراکنش‌ها و قفل‌ها

مدیریت تراکنش‌ها و قفل‌ها در MS SQL Server می‌تواند تأثیر زیادی بر عملکرد داشته باشد. تراکنش‌ها در هنگام انجام عملیات چندگانه بر روی داده‌ها اطمینان می‌دهند که داده‌ها به صورت یکپارچه و درست ذخیره می‌شوند. با این حال، استفاده نادرست از قفل‌ها یا تراکنش‌ها می‌تواند منجر به کاهش عملکرد و ایجاد مسائل مربوط به همزمانی (Concurrency) شود. طراحی بهینه برای مدیریت تراکنش‌ها و قفل‌ها می‌تواند به کاهش زمان‌های انتظار و افزایش کارایی کمک کند.

#### نحوه مدیریت تراکنش‌ها :

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

```
BEGIN TRANSACTION;
```

```
UPDATE MyTable SET Column1 = 'NewValue' WHERE Column2 = 'Condition';
```

```
COMMIT TRANSACTION;
```

## ۵. پیش پردازش و کش کردن

استفاده از کش‌ها (Caching) و بهینه‌سازی کوئری‌ها یکی دیگر از جنبه‌های مهم طراحی پایگاه داده است MS SQL Server . قابلیت‌هایی برای ذخیره‌سازی موقت نتایج کوئری‌ها و استفاده مجدد از آن‌ها دارد. این امر می‌تواند به شدت زمان پاسخ‌دهی برنامه را کاهش دهد. همچنین، پیش‌پردازش داده‌ها و استفاده از پردازش‌های دسته‌ای (Batch Processing) می‌تواند به کاهش بار پایگاه داده و بهبود عملکرد کمک کند.

## ۶. مدیریت داده‌های بزرگ (Big Data) و مقیاس پذیری

با توجه به رشد سریع داده‌ها و نیاز به ذخیره‌سازی و پردازش داده‌های بزرگ، طراحی پایگاه داده باید به گونه‌ای باشد که بتواند مقیاس‌پذیر باشد MS SQL Server. ویژگی‌هایی مانند جدول‌های پارتیشن‌بندی شده (Partitioned Tables) و شاردینگ (Sharding) را ارائه می‌دهد که امکان تقسیم داده‌ها بر اساس معیارهای خاص را فراهم می‌کند. این ویژگی‌ها می‌توانند بار پایگاه داده را در هنگام پردازش حجم بالای داده کاهش دهند و عملکرد را بهبود بخشند. طراحی مناسب برای مقیاس‌پذیری می‌تواند به‌ویژه در برنامه‌هایی که نیاز به پردازش داده‌های بزرگ دارند، مانند برنامه‌های تجزیه و تحلیل داده و یادگیری ماشین، به شدت مفید باشد.

## ۷. استفاده از نمایه‌ها (Views) و ذخیره‌سازی داده‌های محاسباتی

در برخی از برنامه‌ها، نیاز به پردازش داده‌ها در زمان واقعی (Real-time) وجود دارد که ممکن است به کاهش عملکرد منجر شود. در این شرایط، استفاده از نمایه‌ها (Views) که نتایج محاسبات پیچیده را پیش از زمان به‌صورت ذخیره‌شده فراهم می‌آورند، می‌تواند از فشار به پایگاه داده بکاهد MS SQL Server. از Materialized Views پشتیبانی می‌کند که می‌توانند داده‌های محاسباتی را از قبل ذخیره کنند و باعث کاهش بار پردازشی در هنگام اجرای کوئری‌ها شوند.

نمایه های ذخیره شده :

```
CREATE MATERIALIZED VIEW MyMaterializedView
WITH SCHEMABINDING
AS
SELECT Column1, COUNT_BIG(*)
FROM MyTable
GROUP BY Column1;

REFRESH MATERIALIZED VIEW MyMaterializedView;
```

استفاده از Index View برای کوئری های پیچیده :

```
CREATE VIEW OrdersSummaryView
WITH SCHEMABINDING
AS
SELECT
    CustomerID,
    COUNT(Orders) AS TotalOrders,
    SUM(TotalAmount) AS TotalSales
FROM Orders
GROUP BY CustomerID;

CREATE UNIQUE CLUSTERED INDEX idx_orders_summary ON OrdersSummaryView(CustomerID);
```

## ۸. تأثیر تراکنش‌ها بر عملکرد

در طراحی پایگاه داده، کنترل تراکنش‌ها و همزمانی دسترسی‌ها به داده‌ها می‌تواند تأثیر زیادی بر عملکرد داشته باشد. در MS SQL Server، سطوح ایزولاسیون تراکنش برای مدیریت دسترسی همزمان به داده‌ها و جلوگیری از مشکلاتی مانند روند داده‌های نادرست (Dirty Reads) و قفل‌های طولانی (Deadlocks) قابل تنظیم است. انتخاب سطح ایزولاسیون مناسب مثلاً Read Committed یا Serializable می‌تواند باعث افزایش یا کاهش کارایی شود. اگر تعداد زیادی تراکنش همزمان در حال انجام باشد، طراحی بهینه برای مدیریت این تراکنش‌ها می‌تواند از تأخیر و افت عملکرد جلوگیری کند.

## ۹. استفاده از Query Optimization Techniques

یکی از ویژگی‌های برجسته MS SQL Server، بهینه‌سازی خودکار کوئری‌ها است. در طراحی پایگاه داده، توجه به استراتژی‌های بهینه‌سازی کوئری بسیار مهم است. MS SQL Server از ابزارهایی مانند Execution Plans و Query Hints برای بهینه‌سازی اجرای کوئری‌ها استفاده می‌کند. همچنین، استفاده از Plan Guides می‌تواند به بهینه‌سازی دقیق‌تر کوئری‌ها کمک کند. طراحی کوئری‌ها به‌طور مؤثر می‌تواند زمان پاسخ‌دهی را به‌طور چشمگیری کاهش دهد و استفاده از منابع سیستم را بهینه کند.

```
SELECT *  
FROM Orders WITH (INDEX(idx_clustered))  
WHERE OrderDate >= '2024-01-01';
```

## ۱۰. پشتیبانی از عملیات پیچیده مانند JOIN و Subqueries

در بسیاری از برنامه‌ها، برای دستیابی به اطلاعات خاص نیاز به استفاده از JOIN یا Subqueries است. طراحی پایگاه داده باید به گونه‌ای باشد که این عملیات پیچیده به‌طور مؤثر انجام شود. در MS SQL Server، استفاده از ایندکس‌های مناسب و استراتژی‌های بهینه‌سازی کوئری می‌تواند به اجرای سریع‌تر این عملیات کمک کند. به‌عنوان مثال، ایندکس‌گذاری بر روی ستون‌های مرتبط در جداولی که اغلب با یکدیگر JOIN می‌شوند، می‌تواند سرعت کوئری‌ها را به‌طور چشمگیری افزایش دهد.

## ۱۱. پشتیبانی از تراکنش‌های توزیع شده (Distributed Transactions)

برخی از برنامه‌ها نیاز به ارتباط با چندین پایگاه داده یا سرویس مختلف دارند که به اصطلاح به آن‌ها تراکنش‌های توزیع شده می‌گویند. در این شرایط ، MS SQL Server از Microsoft Distributed Transaction Coordinator (MS DTC) پشتیبانی می‌کند که امکان انجام تراکنش‌های یکپارچه در چندین سرور را فراهم می‌آورد. طراحی صحیح برای مدیریت این نوع تراکنش‌ها می‌تواند به کاهش مشکلات همزمانی و تأخیر در پردازش کمک کند و عملکرد کلی سیستم را بهبود بخشد.

## ۱۲. استفاده از ویژگی‌های جدید SQL Server برای بهبود عملکرد

MS SQL Server در نسخه‌های جدید خود قابلیت‌های متعددی برای بهبود عملکرد ارائه داده است. ویژگی‌هایی مانند In-Memory OLTP ، Columnstore Indexes و Adaptive Query Processing می‌توانند به‌طور قابل توجهی کارایی سیستم را در مواقع پردازش داده‌های پیچیده یا حجم زیاد بهبود بخشند. این ویژگی‌ها در طراحی پایگاه داده باید به‌طور استراتژیک استفاده شوند تا از مزایای آن‌ها بهره‌مند شد.

## ۱۳. تأثیر فشرده‌سازی داده‌ها بر عملکرد

یکی از جنبه‌های طراحی پایگاه داده که به‌ویژه در پایگاه‌های داده با حجم زیاد داده‌ها اهمیت دارد، فشرده‌سازی داده‌ها است . MS SQL Server امکان فشرده‌سازی جداول و ایندکس‌ها را فراهم می‌کند که می‌تواند به کاهش فضای ذخیره‌سازی مورد نیاز و افزایش کارایی کمک کند. با این حال، فشرده‌سازی ممکن است باعث کاهش عملکرد در هنگام نوشتن داده‌ها شود، زیرا فرآیند فشرده‌سازی و بازفشرده‌سازی داده‌ها به پردازش اضافی نیاز دارد. در طراحی پایگاه داده، باید تعادلی بین کاهش فضای ذخیره‌سازی و هزینه‌های پردازشی فشرده‌سازی برقرار شود.

## فشرده سازی داده ها :

```
ALTER TABLE MyTable REBUILD WITH (DATA_COMPRESSION = ROW);  
ALTER TABLE MyTable REBUILD WITH (DATA_COMPRESSION = PAGE);
```

## پارتیشن بندی جداول :

```
CREATE PARTITION FUNCTION MyPartitionFunction (int)
AS RANGE LEFT FOR VALUES (10, 20, 30);
```

```
CREATE PARTITION SCHEME MyPartitionScheme
AS PARTITION MyPartitionFunction
ALL TO ([PRIMARY]);
```

```
CREATE TABLE MyPartitionedTable (
    ID INT,
    Data NVARCHAR(100)
) ON MyPartitionScheme(ID);
```

## Partition Switching برای جداول بزرگ :

```
CREATE TABLE OrdersPartitioned (
    OrderID INT,
    OrderDate DATE
) ON MyPartitionScheme(OrderDate);
```

```
ALTER TABLE OrdersPartitioned SWITCH PARTITION 1 TO OrdersArchive;
```

## ۱۴. بهینه‌سازی برای دسترسی به داده‌ها در شرایط بالا (High Availability)

در برنامه‌های کاربردی حساس به زمان که نیاز به دسترسی مداوم به داده‌ها دارند، دسترسی‌پذیری بالا (High Availability) یکی از اصول کلیدی در طراحی پایگاه داده است. MS SQL Server ابزارهایی همچون Always On Availability Groups و Failover Clustering را ارائه می‌دهد که می‌تواند باعث افزایش قابلیت اطمینان و در دسترس بودن داده‌ها در مواقع خرابی سرور شود. طراحی پایگاه داده باید به گونه‌ای باشد که نه تنها از نظر عملکرد بهینه باشد بلکه در برابر خطاها و خرابی‌ها نیز مقاوم باشد.



## ۱۵. پشتیبانی از تحلیل‌های پیچیده و پردازش‌های تحلیلی (OLAP)

پایگاه‌های داده که برای پردازش‌های تحلیلی و تصمیم‌گیری طراحی می‌شوند مثلاً OLAP یا Data Warehousing نیازمند طراحی خاصی هستند تا بتوانند حجم بالای داده‌ها را به‌طور کارآمد پردازش کنند MS SQL Server. از ابزارهایی مانند SQL Analysis Services (SSAS) Server برای پشتیبانی از تحلیل‌های پیچیده استفاده می‌کند. طراحی مناسب برای این نوع پایگاه داده‌ها شامل استفاده از Data Cubes، Aggregations و Partitioning می‌شود که می‌توانند سرعت تحلیل‌ها و پردازش‌های تحلیلی را به‌شدت افزایش دهند.

```
SELECT ProductID, Region, SUM(SalesAmount)
FROM Sales
GROUP BY CUBE (ProductID, Region);
```

## ۱۶. مدیریت تغییرات داده و نگهداری تاریخچه (Temporal Data)

در برخی از برنامه‌ها، نگهداری تاریخچه تغییرات داده‌ها برای پیگیری تغییرات در طول زمان ضروری است. مدیریت داده‌های زمانی (Temporal Data) در MS SQL Server می‌تواند به کمک جداول System-Versioned انجام شود. این ویژگی به شما این امکان را می‌دهد که علاوه بر ذخیره داده‌های جاری، تاریخچه تغییرات آن‌ها را نیز ذخیره کنید. طراحی صحیح این نوع پایگاه داده می‌تواند به‌ویژه در برنامه‌های مالی، حسابداری و مدیریت مستندات کمک‌کننده باشد.

## ۱۷. آزمایش و نظارت بر عملکرد پایگاه داده

یکی از جنبه‌های حیاتی در طراحی پایگاه داده، آزمایش و نظارت مستمر بر عملکرد آن است MS SQL Server. ابزارهایی مانند SQL Server Profiler و Dynamic Management Views (DMVs) را برای تحلیل عملکرد پایگاه داده ارائه می‌دهد. استفاده از این ابزارها می‌تواند به شناسایی گلوگاه‌های عملکردی و بهبود آن‌ها کمک کند. همچنین، نظارت مستمر بر عملکرد می‌تواند به شناسایی تغییرات غیرمنتظره در عملکرد سیستم و رفع سریع مشکلات کمک کند.

## ۱۸. استراتژی‌های پشتیبان‌گیری و بازیابی (Backup and Recovery)

در هر پایگاه داده‌ای، استراتژی‌های پشتیبان‌گیری و بازیابی (Backup and Recovery) باید به گونه‌ای طراحی شوند که علاوه بر تضمین امنیت داده‌ها، بر عملکرد کلی سیستم نیز تأثیر منفی نگذارند. MS SQL Server انواع مختلفی از روش‌های پشتیبان‌گیری مانند پشتیبان‌گیری کامل (Full Backup)، پشتیبان‌گیری تفاضلی (Differential Backup) و پشتیبان‌گیری تراکنشی (Transaction Log Backup) را ارائه می‌دهد. در طراحی پایگاه داده، انتخاب بهترین استراتژی پشتیبان‌گیری بر اساس نیازهای برنامه و حجم داده‌ها اهمیت ویژه‌ای دارد.

## ۱۹. استفاده از کشینگ و حافظه‌های غیر فرار (Non-Volatile Memory)

استفاده از کش‌ها (Caching) برای ذخیره‌سازی نتایج کوئری‌ها و داده‌های پر مصرف می‌تواند زمان پاسخ‌دهی را به طور قابل توجهی کاهش دهد. علاوه بر کش کردن در RAM، MS SQL Server امکان استفاده از حافظه‌های غیر فرار (Non-Volatile Memory) مانند SSD و NVDIMM را فراهم می‌کند. این حافظه‌ها می‌توانند داده‌ها را با سرعت بسیار بالاتری نسبت به دیسک‌های سخت سنتی ذخیره و بازیابی کنند. استفاده بهینه از این منابع می‌تواند کارایی سیستم را در دسترسی به داده‌های پرتکرار و مهم به طور چشمگیری بهبود بخشد.

## مانیتورینگ عملکرد با Dynamic Management Views (DMVs)

استفاده از DMVs برای نظارت بر عملکرد ایندکس‌ها:

```
SELECT
    OBJECT_NAME(IX.OBJECT_ID) AS TableName,
    IX.name AS IndexName,
    PS.[IndexType_Desc],
    PS.[User_Seeks], PS.[User_Scans], PS.[User_Lookups], PS.[User_Updates]
FROM sys.dm_db_index_usage_stats AS PS
    INNER JOIN sys.indexes AS IX ON PS.[Object_Id] = IX.[Object_Id]
    AND PS.[Index_Id] = IX.[Index_Id]
WHERE OBJECTPROPERTY(IX.OBJECT_ID, 'IsUserTable') = 1;
```

## ۲۰. مدیریت داده‌های توزیع‌شده و جغرافیایی

در بعضی از برنامه‌ها، نیاز به ذخیره و پردازش داده‌های جغرافیایی (Geospatial Data) و توزیع‌شده در چندین مکان جغرافیایی وجود دارد MS SQL Server. از نواحی جغرافیایی و عملیات جغرافیایی (Geospatial Operations) پشتیبانی می‌کند. این ویژگی‌ها می‌توانند برای سیستم‌های مدیریت منابع، ناوبری و تحلیل‌های جغرافیایی کاربردی باشند. طراحی مناسب برای ذخیره‌سازی داده‌های جغرافیایی و توزیع‌شده می‌تواند به‌طور چشمگیری به کارایی برنامه‌هایی که به این نوع داده‌ها وابسته هستند، کمک کند.

### نتیجه‌گیری تکمیلی و نکات نهایی

در مجموع، طراحی پایگاه داده در MS SQL Server باید به‌گونه‌ای انجام شود که علاوه بر بهینه‌سازی عملکرد، از مقیاس‌پذیری، دسترس‌پذیری، امنیت، و قابلیت نگهداری مناسب نیز برخوردار باشد. استفاده از ابزارهای مختلفی مانند ایندکس‌گذاری، بهینه‌سازی کوئری‌ها، فشرده‌سازی داده‌ها، مدیریت تراکنش‌ها و به‌کارگیری ویژگی‌های جدید SQL Server می‌تواند عملکرد سیستم را به میزان قابل توجهی افزایش دهد. از آنجا که هر برنامه نیازهای خاص خود را دارد، طراحی پایگاه داده باید انعطاف‌پذیر و متناسب با نیازهای پروژه و محدودیت‌های منابع باشد.

این تحقیق به تفصیل به بررسی ابعاد مختلف تأثیر طراحی پایگاه داده بر عملکرد برنامه‌ها پرداخته و به توسعه‌دهندگان کمک می‌کند تا با بهره‌گیری از تکنیک‌های مختلف، سیستم‌هایی کارآمدتر و مقیاس‌پذیرتر بسازند.