

راهنمای اجرای Query در sql server

در طول دوره ای که **DBA** بودم، اشتباهات متداولی را در هنگام مرور **query** های **SQL** مشاهده کردم که برخلاف سیستم هایی که تنظیم می کردم، اجرا می شدند. با این تجربه دریافتم که دستورالعمل های کلی وجود دارند که باید در هنگام نوشتن **query** ها و همچنین طراحی یک نمودار دیتابیس دنبال شوند. در این آموزش نگاهی به بخش های مختلفی که این اشتباهات متداول اتفاق می افتند و راه هایی که می توان آنها را برطرف کرد، خواهیم داشت.

این بخش ها عبارتند از:

- **Query writing** (نوشتن query)
- **Indexing** (نمایه سازی)
- **Schema design** (طراحی نمودار)

توضیح:

در هر بخش آموزش مثال های خاصی را بررسی خواهیم کرد که مواردی را توضیح می دهند که در هنگام اجرا در **SQL Server** نباید اتفاق بیفتند. برای هر کدام از آیتم ها یک راه حل یا یک انتخاب ارائه خواهم داد که اجرای بهتری را نمایش می دهند. لطفاً به یاد داشته باشید که این ها روش های کلی بوده و استثنائاتی هم وجود دارند، اما به طور کل دنبال کردن این اصول اولیه ممکن است شما را به یک اجرای سریع و عاقلانه برساند.

موضوعات خاصی که بررسی خواهند شد، عبارتند از:

- نوشتن **query**:
- چگونه ترتیب اتصال می تواند بر روی **Query Plan** تاثیر بگذارد.
- **Function Calls** را از لیست **SELECT** حذف کنید.
- از استفاده از \diamond در عبارت **WHERE** جلوگیری کنید.
- از استفاده از عملکردها در عبارت **WHERE** جلوگیری کنید.
- از کاراکترهای **Wildcard** برای شروع عبارت جستجو جلوگیری کنید.
- از یک جدول مشتق به جای **IN Predicate** با مجموع عملکردها استفاده کنید.
- نمایه سازی (**Indexing**):
- اطمینان حاصل کنید که همه ی ستون های متصل ایندکس شده اند.
- در هنگام ایجاد ایندکس ها از ترتیب **WHERE**، **JOIN**، **ORDER BY**، **SELECT Column** استفاده کنید.
- مطمئن شوید که همه ی جدول ها دارای یک **Clustered Index** تعریف شده می باشند.
- طراحی نمودار:

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

- از گزینه ی **DELETE CASCADE** برای رسیدگی به **Child Key Removal** در ارتباطات کلید خارجی (**Foreign Key**) استفاده کنید.
- لغو نرمالسازی (Denormalize) برای اجرا جدول نمونه و راه اندازی داده (**Sample Table and Data Set up**)

بنابراین می توانید از طریق این مثال ها در محیط خود کار کنید. در زیر نموداری با تست داده مشاهده می کنید که می توانید برای تست هر موضوعی از ان استفاده کنید.

-- NOTE: You have to recreate these tables after each section of the tutorial

-- table creation logic

-- parent table

```
CREATE TABLE [dbo].[Parent](
  [ParentID] [bigint] NOT NULL,
  [IntDataColumn] [bigint] NULL,
  [VarcharDataColumn] [varchar](1000) NULL,
  [DateDataColumn] [datetime] NULL,
  CONSTRAINT [PK_Parent] PRIMARY KEY CLUSTERED
  ([ParentID] ASC)
)
GO
```

-- child table

```
CREATE TABLE [dbo].[Child](
  [ChildID] [bigint] NOT NULL,
  [ParentID] [bigint] NULL,
  [IntDataColumn] [bigint] NULL,
  [VarcharDataColumn] [varchar](10) NULL,
  CONSTRAINT [PK_Child] PRIMARY KEY CLUSTERED
  ([ChildID] ASC)
)
GO
```

-- foreign key constraint

```
ALTER TABLE [dbo].[Child] WITH CHECK
ADD CONSTRAINT [FK_Child_Parent] FOREIGN KEY([ParentID])
REFERENCES [dbo].[Parent] ([ParentID])
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

```
ON DELETE CASCADE
GO
```

```
-- child detail table
```

```
CREATE TABLE [dbo].[ChildDetail](
  [ChildDetailID] [bigint] NOT NULL,
  [ChildID] [bigint] NOT NULL,
  [ExtraDataColumn] [bigint] NULL,
  CONSTRAINT [PK_ChildDetail] PRIMARY KEY CLUSTERED
    ([ChildDetailID],[ChildID] ASC)
)
GO
```

```
-- foreign key constraint
```

```
ALTER TABLE [dbo].[ChildDetail] WITH CHECK
ADD CONSTRAINT [FK_ChildDetail_Child] FOREIGN KEY([ChildID])
REFERENCES [dbo].[Child] ([ChildID])
ON DELETE CASCADE
GO
```

```
-- data load
```

```
DECLARE @val BIGINT
DECLARE @val2 BIGINT
SELECT @val=1
WHILE @val < 100000
BEGIN
  INSERT INTO dbo.[Parent] VALUES(@val,@val,'TEST'+CAST(@val
AS VARCHAR),getdate()-(@val/24.0))

  SELECT @val2=1
  WHILE @val2 < 20
  BEGIN
    INSERT INTO dbo.[Child] VALUES ((@val*100000)+@val2,@val,@val,'TEST'
+CAST(@val AS VARCHAR))
    INSERT INTO dbo.[ChildDetail] VALUES (1,(@val*100000)+@val2,9999)
    INSERT INTO dbo.[ChildDetail] VALUES (2,(@val*100000)+@val2,1111)
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>

```

INSERT INTO dbo.[ChildDetail] VALUES (3,(@val*100000)+@val2,3333)
INSERT INTO dbo.[ChildDetail] VALUES (4,(@val*100000)+@val2,7777)
SELECT @val2=@val2+1
END
SELECT @val=@val+1

END
GO

```

-- small table for joins

```

CREATE TABLE [dbo].[Small](
  [SmallID] [bigint] NOT NULL,
  [IntDataColumn] [bigint] NULL,
  [VarcharDataColumn] [varchar](100) NULL,
  CONSTRAINT [PK_Small] PRIMARY KEY CLUSTERED
  ([SmallID] ASC)
)
GO

```

-- data load

```

INSERT INTO dbo.[Small] VALUES (50,80,'TEST5080')
INSERT INTO dbo.[Small] VALUES (510,810,'TEST510810')
INSERT INTO dbo.[Small] VALUES (7001,9030,'TEST70019030')
INSERT INTO dbo.[Small] VALUES (12093,10093,'TEST1209310093')
INSERT INTO dbo.[Small] VALUES (48756,39843,'TEST48756,39843')
INSERT INTO dbo.[Small] VALUES (829870,57463,'TEST82987057463')
GO

```

-- cleanup statements

```

--DROP TABLE [dbo].[Small]
--DROP TABLE [dbo].[ChildDetail]
--DROP TABLE [dbo].[Child]
--DROP TABLE [dbo].[Parent]

```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>



آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>