



درس : آز پایگاه داده

استاد: ادهم

دانشجو: مهدی منصوری خواه

شماره دانشجویی: ۹۹۳۱۰۵۶

گزارش پروژه

سعی شده است در این گزارش هریک از موارد به همراه دلیل استفاده از آن ذکر شود.

```
use LinkShorteningSystem

CREATE TABLE tblLinks (
    LinkID INT IDENTITY(1,1) PRIMARY KEY,
    OriginalURL NVARCHAR(MAX) NOT NULL,
    ShortURL NVARCHAR(50) UNIQUE NOT NULL,
    CreationDate DATETIME DEFAULT GETDATE(),
    ExpirationDate DATETIME NOT NULL,
    IsExpired BIT DEFAULT 0,
    IsDeleted bit Default 0
);

CREATE TABLE tblStatistics (
    StatID INT IDENTITY(1,1) PRIMARY KEY,
    ShortURL NVARCHAR(50) NOT NULL FOREIGN KEY REFERENCES tblLinks(ShortURL),
    AccessDate DATETIME DEFAULT GETDATE()
);
```

یک دیتابیس به اسم LinkShorteningSystem ساخته شده است که دارای دو جدول tblLinks و tblStatistics است.

همانطور که مشخص است، جدول tblLinks دارای 7 فیلد می باشد که از اسم هر فیلد میتوان به کاربرد و استفاده آن پی برد.

جدول tblStatistics برای نگه داشتن آمار هایی همچون تعداد لینک های اضافه شده در روز و بیشترین لینک هایی که بهشون ارجاع داده شده است است.

در ادامه این جداول بیشتر توضیح داده می شوند.

- ثبت یک لینک جدید

```
CREATE PROCEDURE sp_SaveLink
    @OriginalUrl Nvarchar(max),
    @ShortUrl Nvarchar(50) output
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS(select 1 from tblLinks where OriginalURL = @OriginalUrl)
    BEGIN
        select @ShortUrl = ShortUrl From tblLinks where OriginalUrl = @OriginalUrl;
        return 0;
    END

    DECLARE @Chars NVARCHAR(62) = 'abcdefghijklmnopqrstuvwxyz0123456789';
    DECLARE @RandomString NVARCHAR(6);
    DECLARE @i INT = 1;

    SET @RandomString = '';

    WHILE @i <= 6
    BEGIN
        SET @RandomString = @RandomString + SUBSTRING(@Chars, ABS(CHECKSUM(NEWID())) % LEN(@Chars) + 1, 1);
        SET @i = @i + 1;
    END

    INSERT INTO tblLinks (OriginalURL, ShortURL, ExpirationDate)
    VALUES (@OriginalURL, @RandomString, DATEADD(DAY, 7, GETDATE()));

    set @ShortUrl = @RandomString
    return 0;
END
```

برای ثبت یک لینک جدید از store procedure استفاده کردم به این دلیل که یکبار تعریف می شود و هر جا لازم بشه میتوان ازش استفاده کرد و همچنین عملیات های پیچیده را میتوان به راحتی در sp انجام داد و چند ورودی و خروجی داشته داشت.

کد به این صورت کار میکند که اگر لینک کوتاه سازی شده بود و در جدول وجود داشت مقدار کوتاه شده لینک را برمی گرداند در غیراین صورت یک رشته تصادفی 6 کاراکتر متشکل از حروف کوچک و اعداد تولید می کند و در جدول tblLinks ذخیره می کند.

- ارجاع به یک لینک کوتاه شده

```
sqlCommand.sql - D...J.master (ma (54))* X
52
53 Create PROCEDURE sp_GetOriginalURL
54     @ShortURL NVARCHAR(50),
55     @OriginalURL NVARCHAR(MAX) OUTPUT
56 AS
57 BEGIN
58     SET NOCOUNT ON;
59
60     BEGIN TRANSACTION;
61
62     IF EXISTS (SELECT 1 FROM tblLinks WHERE ShortURL = @ShortURL)
63     BEGIN
64
65         IF EXISTS (
66             SELECT 1
67             FROM tblLinks
68             WHERE ShortURL = @ShortURL
69                 AND ExpirationDate > GETDATE()
70                 AND IsExpired = 0
71         )
72         BEGIN
73
74             SELECT @OriginalURL = OriginalURL
75             FROM tblLinks
76             WHERE ShortURL = @ShortURL;
77
78             INSERT INTO tblStatistics (ShortURL)
79             VALUES (@ShortURL);
80             -- trigger call here
81         END
82     ELSE
83     BEGIN
84
85         ROLLBACK TRANSACTION;
86         RAISERROR ('The link is expired...', 16, 1);
87         RETURN;
88     END
89 END
90 ELSE
91 BEGIN
92     ROLLBACK TRANSACTION;
93     RAISERROR ('The link does not exist...', 16, 1);
94     RETURN;
95 END
96
97 COMMIT TRANSACTION;
98 END;
99
```

برای ارجاع به یک لینک کوتاه شده از store procedure استفاده کردم به این دلیل که به راحتی میتوان خطاها را هندل کرد(در تابع نمیتوان خطاها را هندل کرد)، همچنین یکبار تعریف می شود و هر جا لازم بشه میتوان ازش استفاده کرد و همچنین عملیات های پیچیده را میتوان به راحتی در sp انجام داد و چند ورودی و خروجی داشته داشت.

کد به این صورت عمل میکند که یک shortUrl دریافت می کند و اگر در دیتابیس وجود داشت و همچنین زمان انقضای آن نگذشته بود، مقدار originalUrl یا همان لینک کوتاه نشده را برمی گرداند. در صورتی که زمان انقضای آن گذشته بود یا در دیتابیس وجود نداشت، خطای مربوطه را به کاربر نشان میدهد.

```
100  /* After insert data to tblStatistics check expire time for link */
101  CREATE TRIGGER trg_LogAccessAndExpire
102  ON tblStatistics
103  AFTER INSERT
104  AS
105  BEGIN
106      SET NOCOUNT ON;
107
108      UPDATE tblLinks
109      SET IsExpired = 1
110      WHERE ExpirationDate <= GETDATE()
111
112  END;
113
114
115
```

به دلیل این نکته در دستور کار "جهت کاهش حجم اطلاعات ذخیره شده، هر لینکی که به مدت یک هفته مورد ارجاع قرار نگرفت باید منقضی شود." از یک trigger استفاده شده است به این علت که زمان انقضا باید به صورت اتوماتیک چک شود و trigger این کار را انجام میدهد. به این صورت به ازای هر insert که در جدول tblStatistics رخ میدهد این فرایند چک کردن تاریخ انقضا با استفاده از trigger انجام می شود.

- اطلاعات داشبورد

- نمایش نمودارهای تعداد لینک جدید ثبت شده در روز و تعداد رجوع به لینک های کوتاه شده در روز

```
Create Function f_CountRegisterdLinkToday()
RETURNS int
AS
BEGIN
    DECLARE @numberOfRegisterdLinkToday int;
    SELECT @numberOfRegisterdLinkToday = Count(LinkID)
    FROM dbo.tblLinks t
    WHERE CAST(t.CreationDate AS DATE) = CAST(GETDATE() AS DATE) and t.IsDeleted = 0;
    RETURN @numberOfRegisterdLinkToday;
END;

SELECT dbo.f_CountRegisterdLinkToday();

Create Function f_CountReferToshortenedlinksToday()
RETURNS int
AS
BEGIN
    DECLARE @numberOfReferToshortenedlinksToday int;
    SELECT @numberOfReferToshortenedlinksToday = Count(StatID)
    FROM dbo.tblStatistics t
    WHERE CAST(t.AccessDate AS DATE) = CAST(GETDATE() AS DATE);
    RETURN @numberOfReferToshortenedlinksToday;
END;

SELECT dbo.f_CountReferToshortenedlinksToday();
```

در این دو تابع دو عدد تعداد لینک جدید ثبت شده در روز و تعداد رجوع به لینک های کوتاه شده در روز محاسبه می شود به این صورت که با استفاده از آیدی هر دو جدول و تاریخ هر لینک به راحتی می توان این دو منطق را پیاده سازی کرد.

دلیل استفاده از تابع این است که منطق پیچیده ای ندارد و از select استفاده شده است و لازم نیست که چند مقدار را برگردانیم که مجبور باشیم از store procedure استفاده کنیم.(فقط یک عدد برگردانده می شود). به همین دلیل از تابع استفاده کردم.

- نمایش 3 لینکی که بیشتر از بقیه مورد ارجاع قرار گرفته اند.

```
CREATE PROCEDURE sp_GetTop3Links
AS
BEGIN
    SET NOCOUNT ON;

    SELECT TOP 3
        l.ShortURL,
        l.OriginalURL,
        COUNT(s.StatID) AS TotalAccesses
    FROM tblStatistics s
    INNER JOIN tblLinks l ON s.ShortURL = l.ShortURL
    GROUP BY l.ShortURL, l.OriginalURL
    ORDER BY TotalAccesses DESC;
END;

EXEC sp_GetTop3Links;
```

برای پیاده سازی این قسمت از store procedure استفاده کردم.(میتوان در این قسمت هم از تابع استفاده کرد، تفاوتی

ایجاد نمی کند.) به این صورت که یک inner join بین دو جدول خورده است زیرا به دنبال اسم لینک ها در جدول tblLinks و تعداد های ارجاعات آن در جدول tblStatistics هستیم و در آخر سه تا از بیشترین ارجاعات را برمی گردانیم.

- نمایش کلیه نگاشت های موجود در سیستم به همراه زمان باقیمانده تا انقضا و تعداد رجوع انجام شده

```
Alter PROCEDURE sp_DetailsOfLinks
AS
BEGIN
    SET NOCOUNT ON;

    SELECT
        l.ShortURL,
        l.OriginalURL,
        ISNULL(COUNT(s.StatID), 0) AS TotalAccesses,
        DATEDIFF(DAY, GETDATE(), l.ExpirationDate) AS RemainingExpiration
    FROM tblLinks l
    LEFT JOIN tblStatistics s ON l.ShortURL = s.ShortURL where l.IsDeleted = 0
    GROUP BY l.ShortURL, l.OriginalURL, l.ExpirationDate
    ORDER BY TotalAccesses DESC;
END;

EXEC sp_DetailsOfLinks;
```

در این قسمت مانند قسمت قبل، هم میتوان از sp و هم از تابع استفاده کرد. به این صورت که یک left join بین دو جدول خورده است زیرا به دنبال نمایش کلیه نگاشت های موجود در سیستم در جدول tblLinks و تعداد های

ارجاعات آن و مدت زمان باقی مانده تا انقضا لینک ها با جدول tblStatistics هستیم.



و در آخر روی ستون های shortUrl و originalURI در جدول tblLinks، ایندکس قرار داده شده است زیرا این ستون ها زیاد سرچ می شوند به همین دلیل ایندکس قرار داده شده است که سرعت جستجو بالاتر برود.

برای پیاده سازی این پروژه از **Razore page** در دات نت استفاده شده است که با زبان **#C** زده شده است و از **orm** مانند **Dapper** برای فراخوانی روال ها، توابع و... که در **sql server** ساختیم استفاده شده است.



در این بخش تعداد لینک های ثبت نام شده در روز و تعداد لینک های ارجاع شده به همراه نمودار آن ها رسم شده است.

Create Short URL

Enter Original URL:

Shorten URL

Retrieve Original URL

Enter Short URL:

Retrieve Original URL

Top Links

#	Short URL	Original URL	Total Accesses
1	ghub11	https://github.com	8
2	apple3	https://apple.com	4
3	msft22	https://microsoft.com	3

در این قسمت کاربر میتواند لینک ها خود برای ساختن لینک کوتاه وارد کند و همچنین با استفاده از لینک کوتاه،لینک اصلی را بازیابی کند. در صورتی که خطایی رهد،خطا از سمت Sql server به کاربر نمایش داده می شود.

در قسمت پایین تر آن، سه لینکی که بیشترین ارجاع را داشته است نمایش داده شده است.

All Links				
#	Short URL	Original URL	Total Resumes	Remaining Days
1	ghub11	https://github.com	8	12
2	apok3	https://apple.com	4	27
3	mdt22	https://microsoft.com	3	17
4	opna4	https://opnai.com	2	22
5	aman23	https://amazon.com	1	0
6	abst23	https://example.com	1	4
7	v80w2	https://www.sqlbhash.com/learn-sql-user-defined-functions/	1	4
8	08earj	https://www.aparat.com/game	1	4
9	ves5a5	mdt22	0	4
10	qjrh7	https://www.w3schools.com/sql/sql_create_index.asp	0	5
11	va8awo	mdt23	0	5
12	joni4a	AAAA	0	5
13	xar85c	https://www.hetabrah.ir/	0	7
14	0g1764	https://web.bale.ai/	0	7
15	j0g94n	https://www.aparat.com/vi/4Sp460	0	4
16	nm63i	https://www.w3schools.com/sql/sql_ref_sql_server.asp	0	4
17	c3w7zu	https://www.aparat.com/v_13018423	0	4
18	goog1	https://google.com	0	4
19	yahoo7	https://yahoo.com	0	7
20	etack9	https://etackoverflow.com	0	2
21	linkdin	https://linkedin.com	0	15
22	exp123	https://expindin.com	0	-5
23	q8acnd	https://localhost:7011/dashboard	0	4

و در آخر تمام لینک ها با جزئیات اطلاعاتشون نمایش داده شده است.

تمام موارد پیاده سازی با جزئیات آن به همراه دلیل استفاده از هر کدام ذکر شد.

