

## مقدمه

مقدار null به معنی پوچ و هیچ می‌باشد اما زمانی که در مقدار دهی جداول از آن استفاده می‌نمایم با توجه به نوع آن ستون فضای متفاوتی اشغال می‌نماید. شاید در پایگاه داده‌های کوچک زیاد مطرح نباشد اما زمانی که حداقل چند گیگ حجم آن باشد و فرضاً 20 تا 30 درصد آن از مقادیر null پر شده باشد فضای زیادی از پوچ گرفته شده است این در حالی است که خیلی از توسعه دهندگان اصلاً به اهمیت استفاده از null توجهی نمی‌کنند و از مقادیری غیر معتبری مثل 0 یا 1- در آن ستون به جای null استفاده می‌کنند.

## SQL Server Sparce Columns

sparse column یا ستون‌های تنک قابلیتی از که از SQL Server 2008 اضافه شده و به ستون‌های عادی امکان استفاده بهینه از فضای ذخیره شده برای مقادیر null را می‌دهد. در واقع sparse column فضای مورد نیاز برای مقادیر null نسبت به مقادیر غیر null را کاهش می‌دهد. با استفاده از sparse column فضای ذخیره شده حداقل 20 تا 40 درصد کمتر خواهد شد.

## ویژگی‌های Sparse Columns

SQL Server Database Engine از کلمه کلیدی SPARSE برای تعریف یک ستون که مقادیر آن می‌بایست بهینه شود استفاده می‌نماید.

نمای Catalog جداول با ستون sparse شبیه جداول معمولی می‌باشد. مقدار برگشتی از تابع COLUMNS\_UPDATED با ستون sparse متفاوت از ستون معمولی است. در نوع داده‌های زیر امکان استفاده از sparse columns را ندارند:

|                         |           |
|-------------------------|-----------|
| text                    | geography |
| timestamp               | geometry  |
| user-defined data types | image     |
|                         | ntext     |

sparse column فضای بیشتری برای ذخیره داده‌های غیر null نسبت به داده‌های نشانه گذاری نشده با SPARSE لازم دارد و این فضا 4 بایت بیشتر از ستون معمولی است. برآورد فضای ذخیره شده براساس نوع داده با طول ثابت در جدول زیر آورده شده است:

| نوع داده   | بایت بدون sparse | بایت sparse | درصد null |
|------------|------------------|-------------|-----------|
| bit        | 0.125            | 5           | 98%       |
| tinyint    | 1                | 5           | 86%       |
| smallint   | 2                | 6           | 76%       |
| int        | 4                | 8           | 64%       |
| bigint     | 8                | 12          | 52%       |
| real       | 4                | 8           | 64%       |
| float      | 8                | 12          | 52%       |
| smallmoney | 4                | 8           | 64%       |
| money      | 8                | 12          | 52%       |

| نوع داده         | بایت بدون sparse | بایت sparse | درصد null |
|------------------|------------------|-------------|-----------|
| smalldatetime    | 4                | 8           | 64%       |
| datetime         | 8                | 12          | 52%       |
| uniqueidentifier | 16               | 20          | 43%       |
| date             | 3                | 7           | 69%       |

نوع داده با دقت - وابسته به طول

| نوع داده              | بایت بدون sparse | بایت sparse | درصد null |
|-----------------------|------------------|-------------|-----------|
| (datetime(2           | 6                | 10          | 57%       |
| (datetime(2           | 8                | 12          | 52%       |
| (time(0               | 3                | 7           | 69%       |
| (time(7               | 5                | 9           | 60%       |
| (datetimeoffset(0     | 8                | 12          | 52%       |
| (datetimeoffset (7    | 10               | 14          | 49%       |
| decimal/numeric(1,s)  | 5                | 9           | 60%       |
| decimal/numeric(38,s) | 17               | 21          | 42%       |
| vardecimal(p,s)       |                  |             |           |

نوع داده - داده وابسته به طول

| نوع داده            | بایت بدون sparse | بایت sparse | درصد null |
|---------------------|------------------|-------------|-----------|
| sql_variant         | 2*               | 2*          | 60%       |
| varchar or char     | 2*               | 4*+         | 60%       |
| nvarchar or nchar   | 2*               | 4*          | 60%       |
| varbinary or binary | 2*               | 4*          | 60%       |
| xml                 | 2*               | 4*          | 60%       |
| hierarchyid         | 2*               | 4*          | 60%       |

#### محدودیت‌های استفاده از Sparse columns

- sparse column می‌بایست nullable باشد و نمی‌تواند ROWGUIDCOL یا IDENTITY باشد.
- sparse column مقدار پیش فرض نمی‌تواند داشته باشد
- ستون محاسبه ای نمی‌تواند sparse باشد
- sparse column نمی‌تواند بخشی از clustered index یا unique primary key باشد
- sparse column نمی‌تواند بخشی از user-defined table باشد

مثالی از کاربرد Sparse columns

```
CREATE TABLE Employees_sparse (
    EMP_ID INT IDENTITY(5001,1) PRIMARY KEY,
    SSN CHAR(9) NOT NULL,
    TITLE CHAR(10) SPARSE NULL,
```

```

FIRSTNAME VARCHAR(50) NOT NULL,
MIDDLEINIT CHAR(1) SPARSE NULL,
LASTNAME VARCHAR(50) NOT NULL,
EMAIL CHAR(50) SPARSE NULL)
GO

```

```

CREATE TABLE Employees (
EMP_ID INT IDENTITY(5001,1) PRIMARY KEY,
SSN CHAR(9) NOT NULL,
TITLE CHAR(10) NULL,
FIRSTNAME VARCHAR(50) NOT NULL,
MIDDLEINIT CHAR(1) NULL,
LASTNAME VARCHAR(50) NOT NULL,
EMAIL CHAR(50) NULL)
GO

```

در این دو جدول یکی با سه ستون Sparse و دیگری بدون Sparse ایجاد شده و با 50000 ردیف داده پر شده است حال با رویه ذخیره شده sp\_spaceused می‌توان فضای ذخیره شده دو جدول را باهم مقایسه نمود.

```

sp_spaceused 'Employees'
GO
sp_spaceused 'Employees_sparse'

```

| Results Messages |                  |       |          |         |            |        |
|------------------|------------------|-------|----------|---------|------------|--------|
|                  | name             | rows  | reserved | data    | index_size | unused |
| 1                | Employees        | 50000 | 5064 KB  | 5008 KB | 32 KB      | 24 KB  |
|                  | name             | rows  | reserved | data    | index_size | unused |
| 1                | Employees_sparse | 50000 | 3080 KB  | 3064 KB | 16 KB      | 0 KB   |

البته ذکر این نکته گفتی است که بهتر است از این تکنیک برای جداولی که تعداد زیادی ستون null دارند استفاده شود.