کوئری‌نویسی پیشرفته

Microsoft® SQL Server®

مدرس: مهدی شیشه‌بری

nikamooz;
آموزش برنامه نویسی و اجرای پروژه

جلسه ششم

معرفی مهدی شیشه‌بری

۱. مدرس و مشاور SQL Server

۲. همکاری با سازمان‌ها و شرکت‌های دولتی و خصوصی

۳. مدرس دوره‌های SQL Server در نیک‌آموز

مطالب آموزشی
جلسه ششم

عناوین موضوعات

- معرفی و بررسی Order Set Function
- ساختارهای ذخیره‌سازی در SQL Server
- بررسی ساختار Heap
- بررسی یک سناریو کاربردی

Order Set Function

Order Set Function

- یکی از انواع Aggregate Function
- شباهت: عملکرد محاسباتی
- تفاوت: متناسب با Ordering

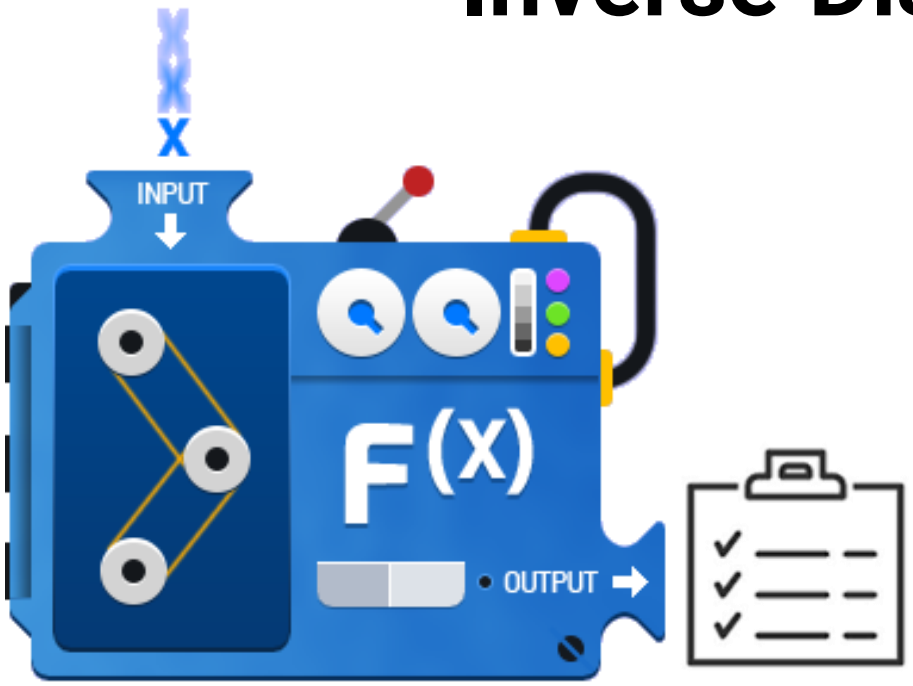
Order Set Function

- از نگاه استاندارد SQL

```
<ordered set function> WITHIN GROUP ( ORDER BY <sort specification list> )
```

Order Set Function

- Hypothetical Function
- Inverse Distribution Function



Hypothetical Function

- عملیات براساس مقادیر فرضی ورودی
- عدم پیاده‌سازی در SQL Server

Inverse Distribution Function

- پیاده‌سازی در SQL Server 2012
- صرفا برای Window Function

معرفی تابع STRING_AGG

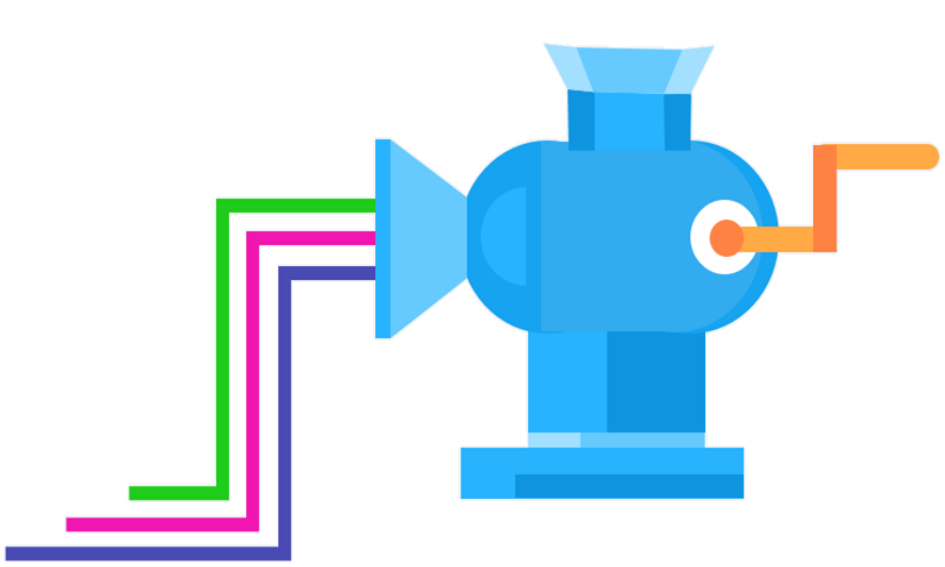
- ادغام مقادیر عبارات به‌همراه جداکننده در یک عبارت رشته‌ای
- استفاده مستقیم در گروه‌بندی

سناریوهای کاربردی

Max Concurrency Interval

Max Concurrency Interval

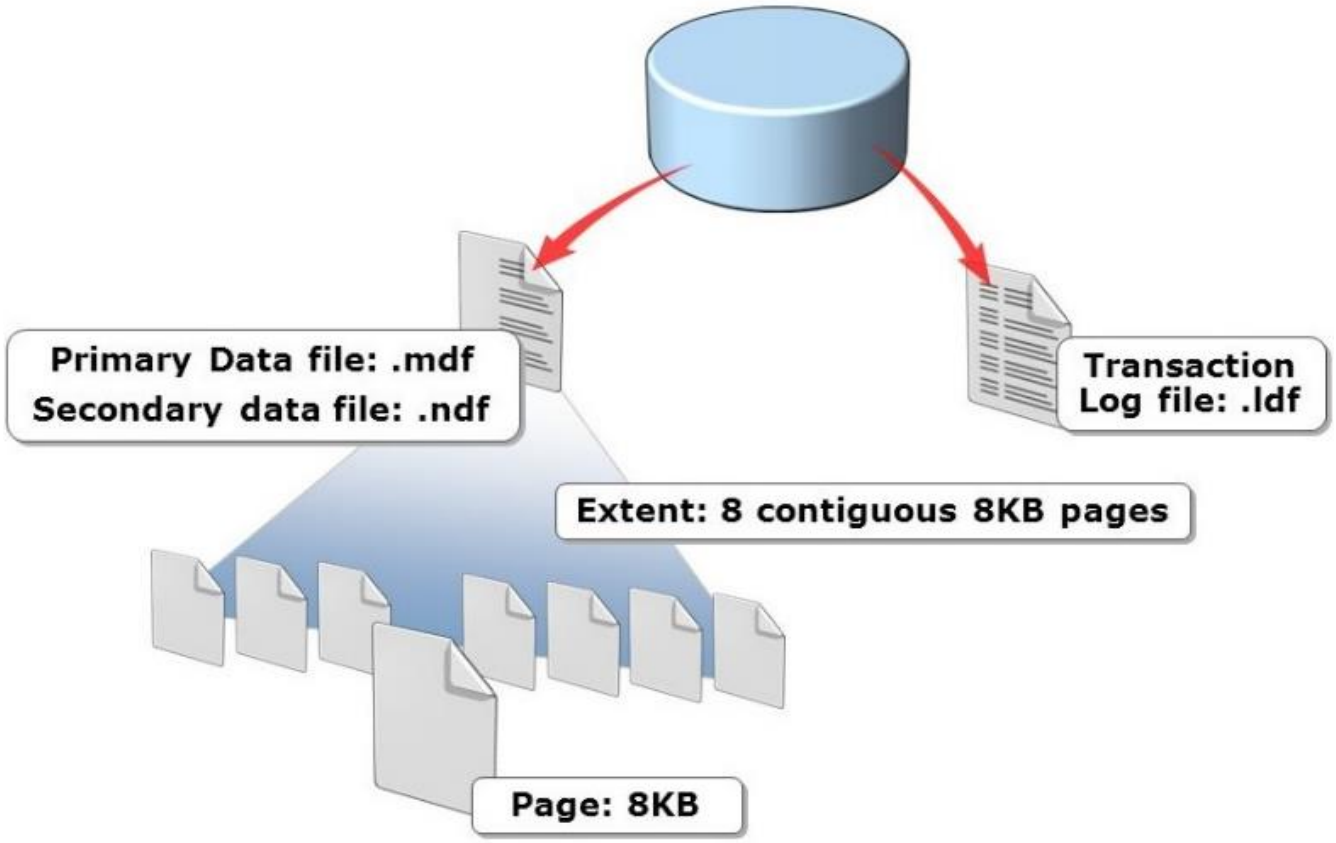
- شناسایی بیشترین فعالیت‌های همزمان



معماری دیتابیس
در
SQL Server

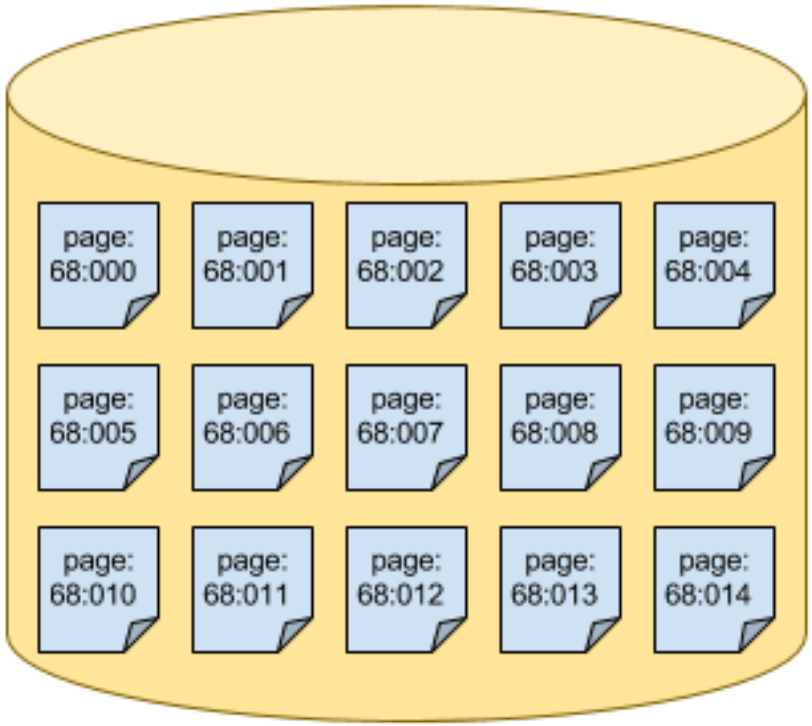
معماری دیتابیس

- Data File
- Log File



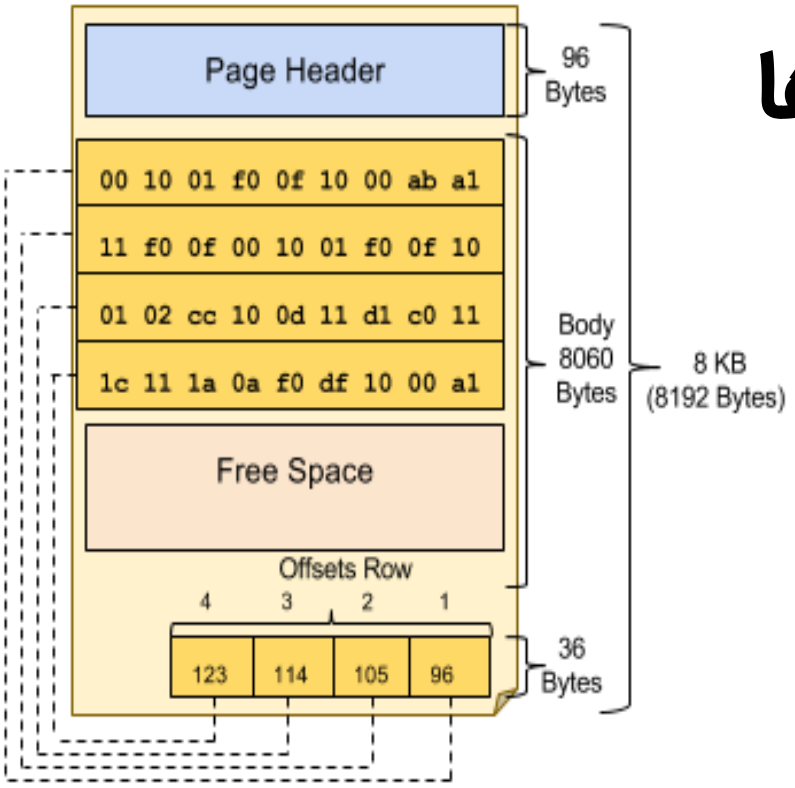
ساختار Data File

- File ID
- Page Number

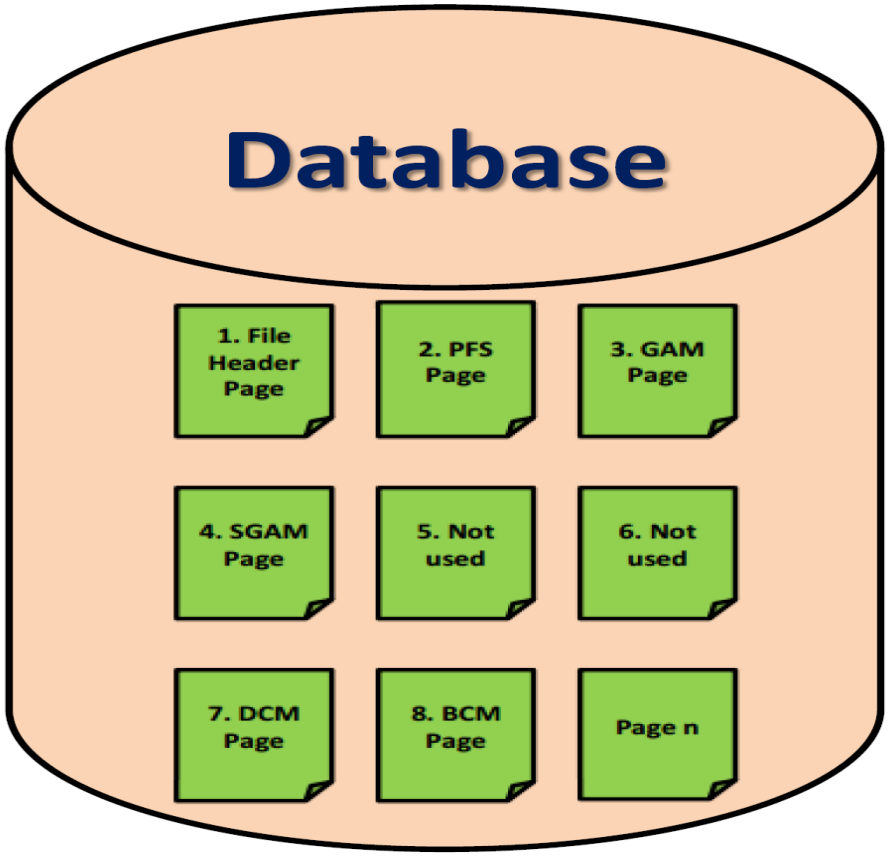


Page چیست؟

- محل ذخیره‌سازی Objectها
- محل ذخیره‌سازی دیتاها
- عملیات I/O



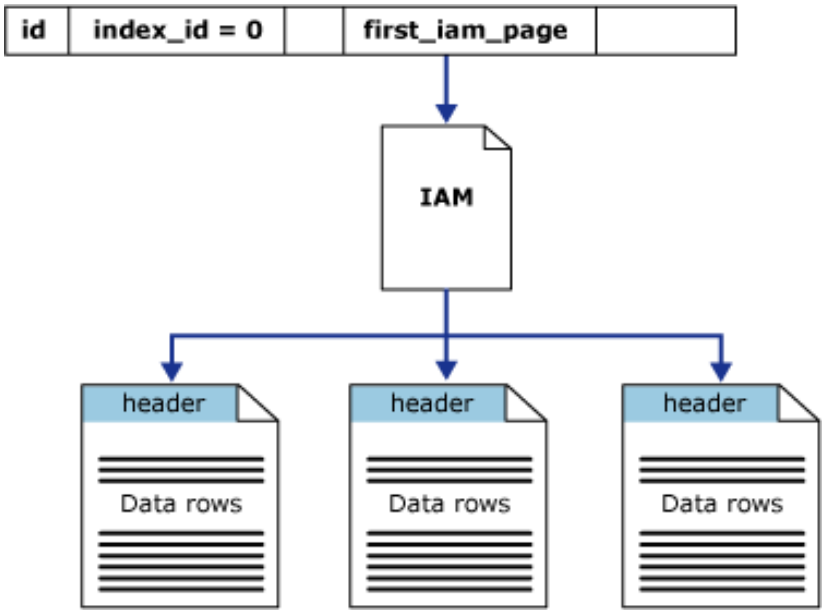
انواع Page در SQL Server



- Data Page
- Index Page
- Text / Image Page
- GAM Page
- SGAM Page
- PFS Page
- IAM Page
- BCM Page
- DCM Page

IAM Page چیست؟

- Index Allocation Map
- ردیابی Pageهای مرتبط با یک جدول



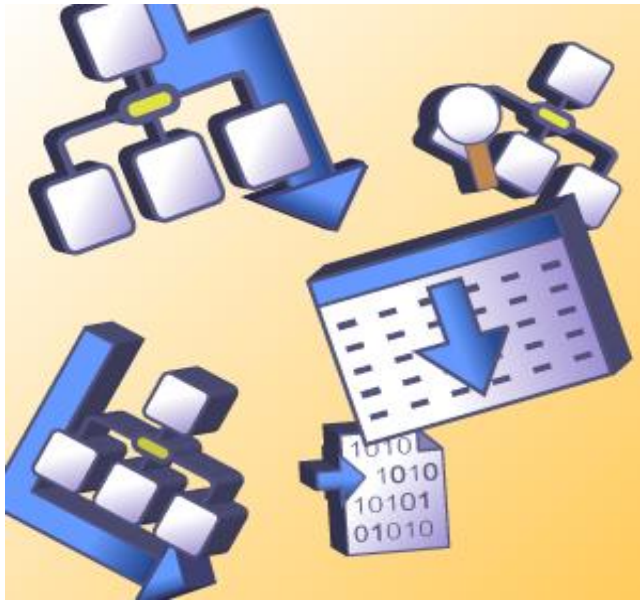
Index
در
SQL Server

Index چیست؟

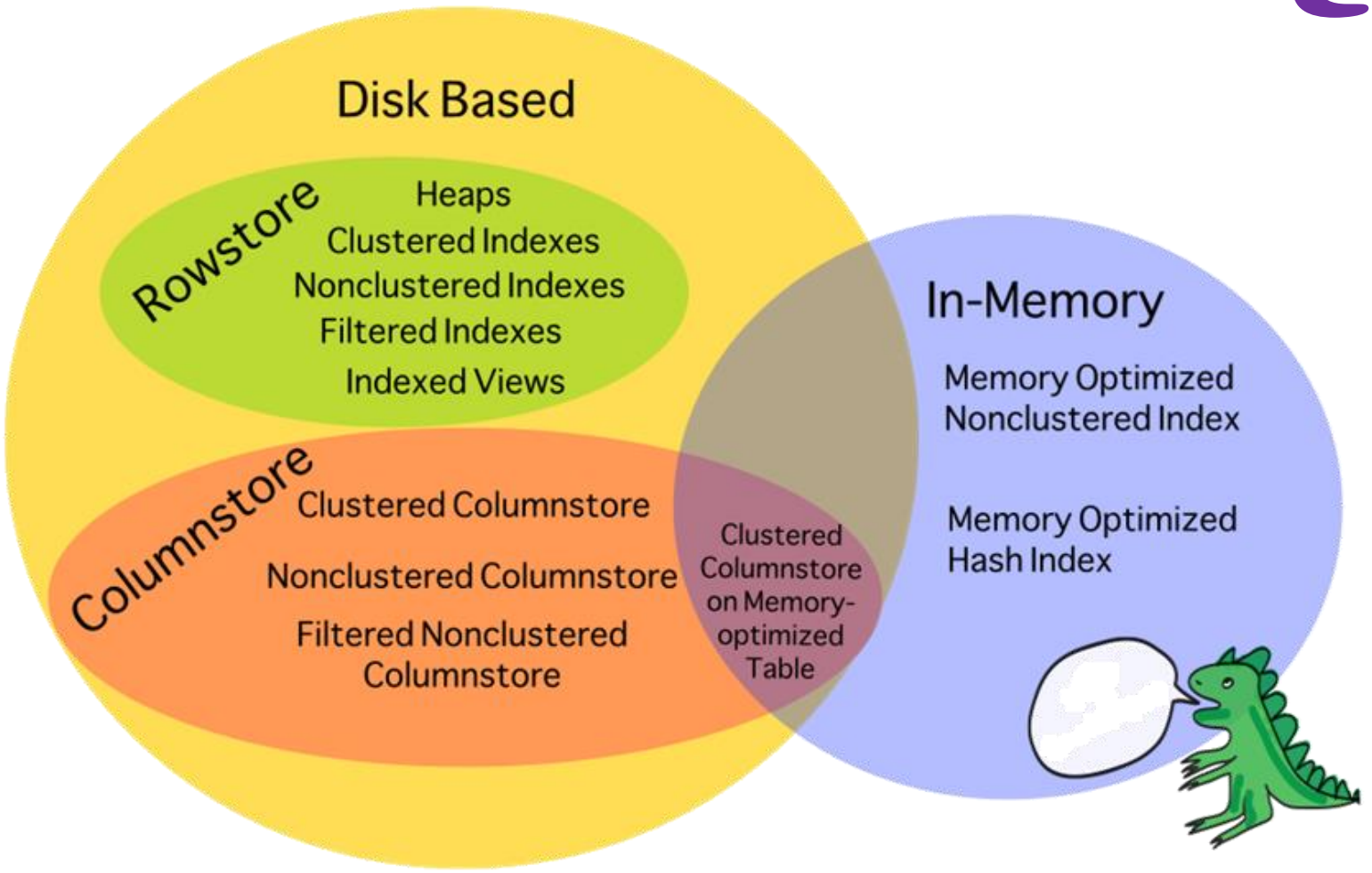
- شاخص/ نمایه
- لیست مرتب‌شده‌ای از اطلاعات

انواع Index

- Clustered Index
- Nonclustered Index
- XML Index
- Spatial Index
- Columnstore Index

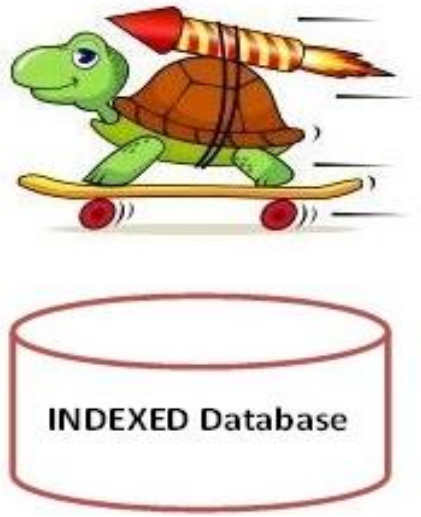


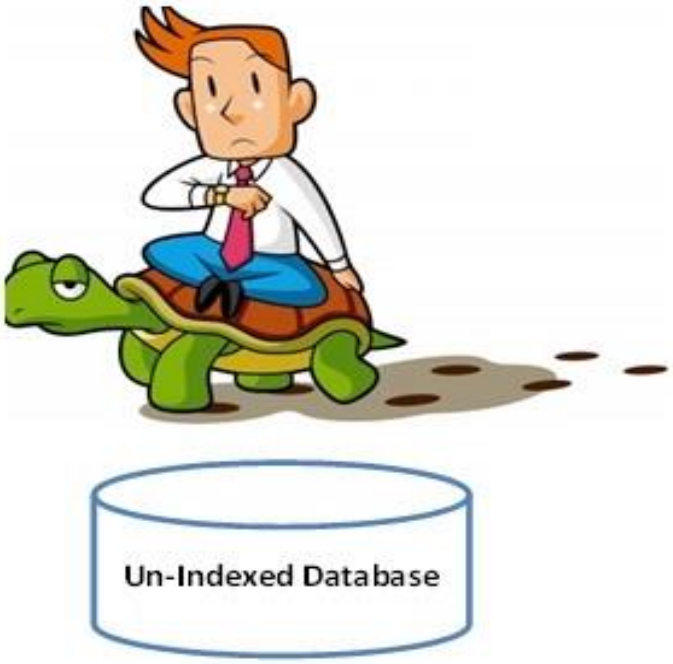
انواع Index


Disk Based
Rowstore
Heaps
Clustered Indexes
Nonclustered Indexes
Filtered Indexes
Indexed Views
Columnstore
Clustered Columnstore
Nonclustered Columnstore
Filtered Nonclustered Columnstore
Clustered Columnstore on Memory-optimized Table
In-Memory
Memory Optimized Nonclustered Index
Memory Optimized Hash Index

چرا Index؟

- افزایش سرعت





جستجوی اطلاعات
در
SQL Server

جستجوی اطلاعات

A	K
B	D
C	W
X	B
B	L
J	R

لیست مرتب نشده

A	K
B	D
C	W
X	B
B	L
J	R

بررسی لیست
(Scan)

جدول مرتب شده

A	J
B	K
B	L
B	R
C	W
D	X

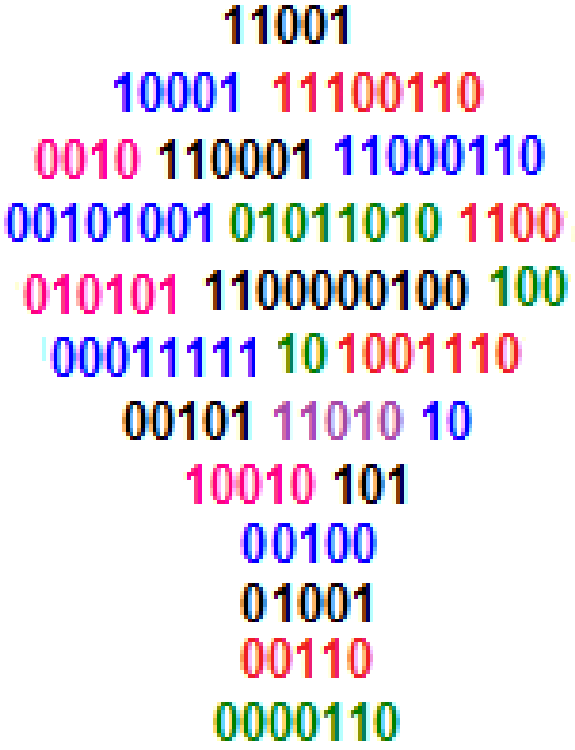
مرتب شدن
کارکترها

ایندکس/ راهنما

محل قرارگیری	ایندکس
ستون اول	A-D
ستون دوم	J-X

ساختارهای ذخیره‌سازی در SQL Server

ساختار ذخیره‌سازی دیتا در Page



- Row Based
  - Heap Table
  - Clustered Table
- Column Based

ساختار ذخیره‌سازی دیتا در Page

Table of Data

date	price	size
2011-01-20	10.1	10
2011-01-21	10.3	20
2011-01-22	10.5	40
2011-01-23	10.4	5
2011-01-24	11.2	55
2011-01-25	11.4	66
...	...	...
2013-03-31	17.3	100

Row Oriented Database

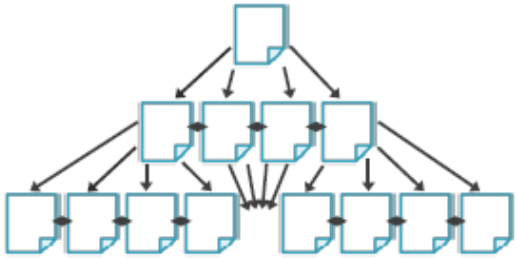
date	price	size
2011-01-20	10.1	10
2011-01-21	10.3	20
2011-01-22	10.5	40
2011-01-23	10.4	5
2011-01-24	11.2	55
2011-01-25	11.4	66
...	...	...
2013-03-31	17.3	100

Column Oriented Database

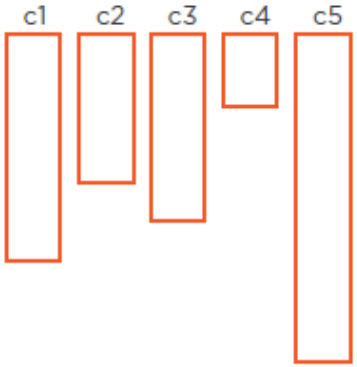
date	price	size
2011-01-20	10.1	10
2011-01-21	10.3	20
2011-01-22	10.5	40
2011-01-23	10.4	5
2011-01-24	11.2	55
2011-01-25	11.4	66
...	...	...
2013-03-31	17.3	100

ساختار ذخیره‌سازی دیتا در Page

Heap structure
(possible in all versions)



Row-based clustered index
(possible in all versions)

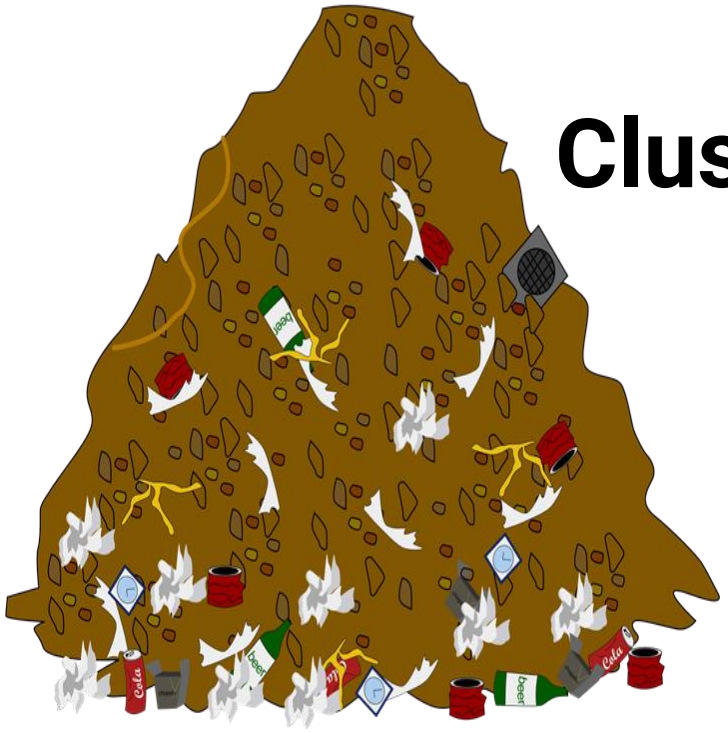


Column-based clustered index
(limited options in SQL Server 2014; more options in SQL Server 2016+)

ساختار Heap

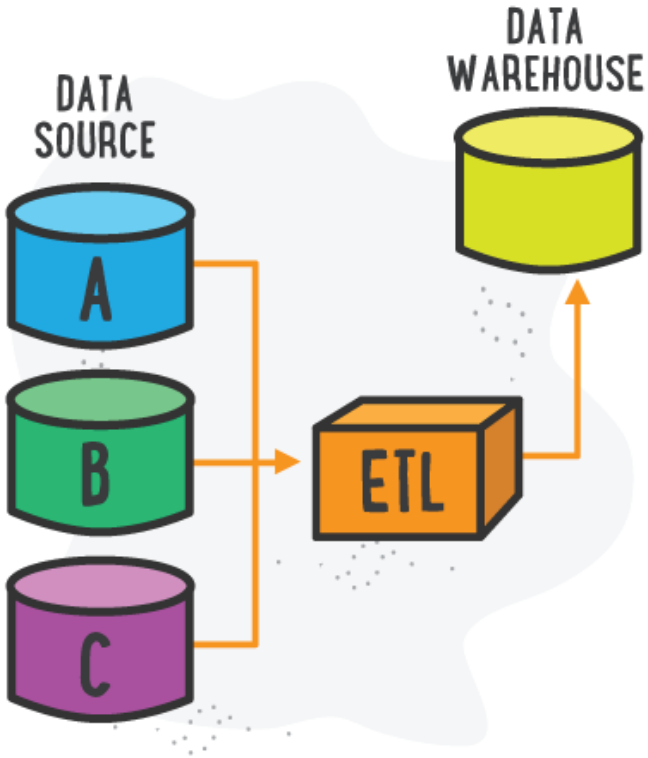
Heap

- توده، کپه و ...
- فاقد نظم و ترتیب
- جداول فاقد Clustered Index



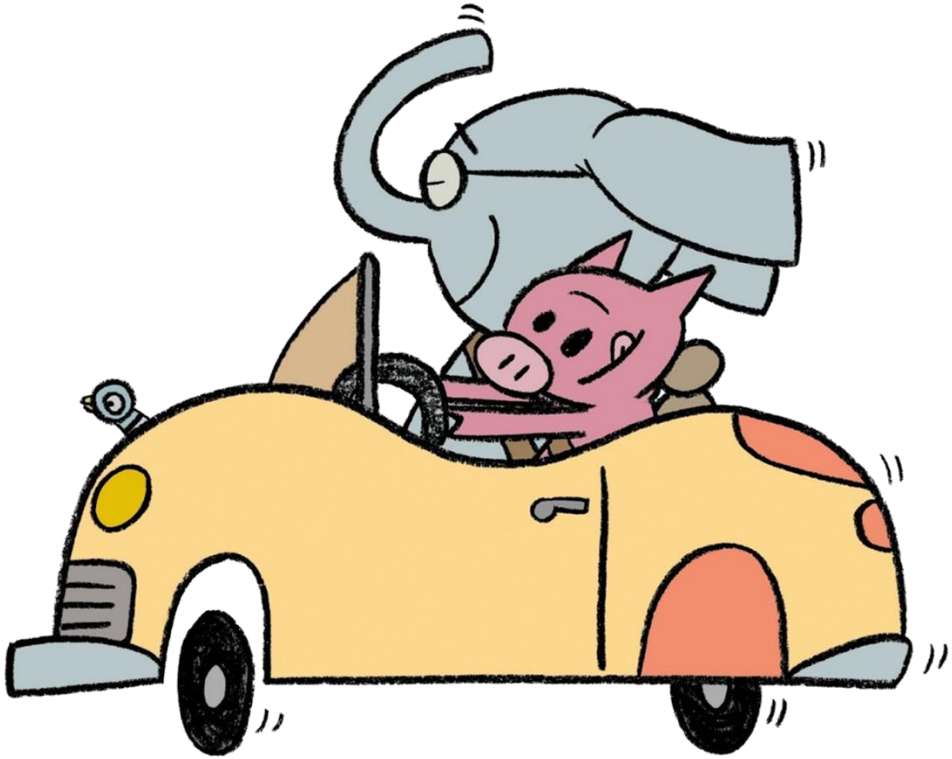
مزایای Heap

- سرعت بالای درج اطلاعات



مزایای Heap

- حجم کمتر نسبت به سایر ساختارها

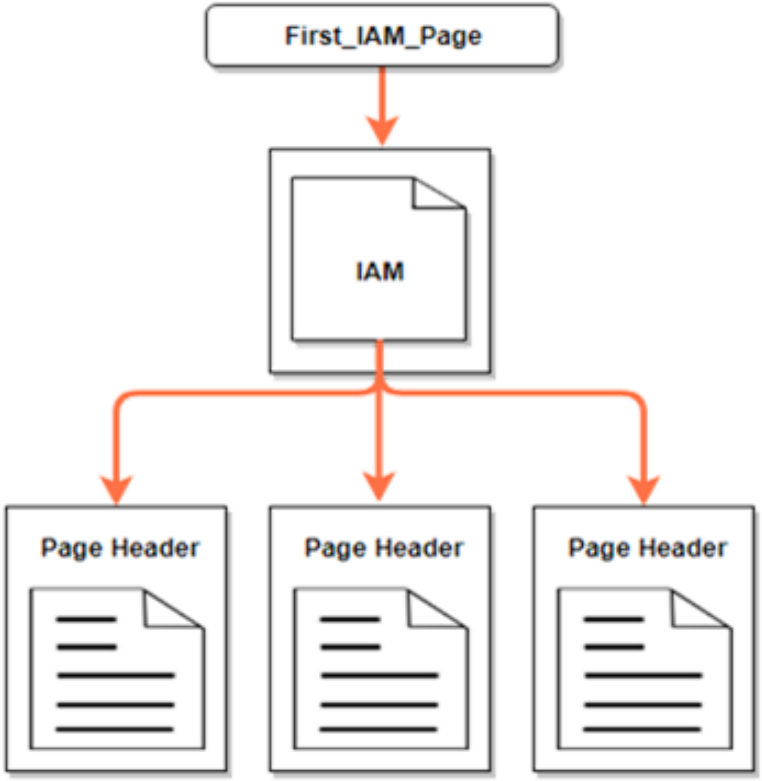


معایب Heap

- سرعت پایین هنگام بازیابی رکوردها

معایب Heap

- عدم ارتباط میان Page جدول



شبکه‌های اجتماعی نیک آموز

اطلاع رسانی سریع کارگاه‌های نسبتا رایگان،

کوپن‌های تخفیف، مقلات، فیلم و دوره‌های نیک آموز

Instagram

Telegram

در این سناریو تعدادی برنامه کاربردی داریم که هر یک در بازه‌های زمانی مختلف می‌توانند به چندین کاربر سرویس دهند.

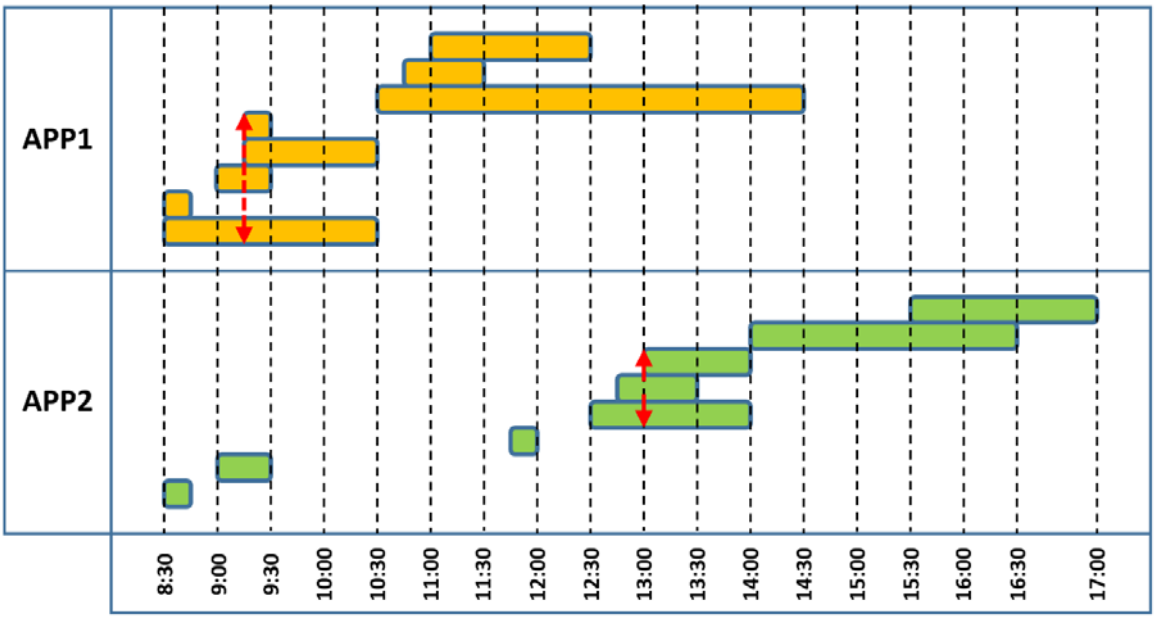
ابتدا با اجرای اسکریپت زیر، جدول Users_Apps را ایجاد کرده و رکوردهای تستی را در آن درج کنید.

```
DROP TABLE IF EXISTS dbo.Users_Apps;
GO

CREATE TABLE dbo.Users_Apps
(
  Code INT PRIMARY KEY,
  App VARCHAR(10),
  UserID VARCHAR(10),
  Start_Time DATETIME,
  End_Time DATETIME
);
GO

INSERT INTO dbo.Users_Apps(Code, App, UserID, Start_Time, End_Time)
VALUES
  (1,  'app1', 'user1', '20180212 08:30', '20180212 10:30'),
  (2,  'app1', 'user2', '20180212 08:30', '20180212 08:45'),
  (3,  'app1', 'user3', '20180212 09:00', '20180212 09:30'),
  (4,  'app1', 'user4', '20180212 09:15', '20180212 10:30'),
  (5,  'app1', 'user5', '20180212 09:15', '20180212 09:30'),
  (6,  'app1', 'user6', '20180212 10:30', '20180212 14:30'),
  (7,  'app1', 'user7', '20180212 10:45', '20180212 11:30'),
  (8,  'app1', 'user8', '20180212 11:00', '20180212 12:30'),
  (9,  'app2', 'user8', '20180212 08:30', '20180212 08:45'),
  (10,  app2', 'user7', '20180212 09:00', '20180212 09:30'),
  (11, 'app2', 'user6', '20180212 11:45', '20180212 12:00'),
  (12, 'app2', 'user5', '20180212 12:30', '20180212 14:00'),
  (13, 'app2', 'user4', '20180212 12:45', '20180212 13:30'),
  (14, 'app2', 'user3', '20180212 13:00', '20180212 14:00'),
  (15, 'app2', 'user2', '20180212 14:00', '20180212 16:30'),
  (16, 'app2', 'user1', '20180212 15:30', '20180212 17:00');
GO
```

اکنون از شما می‌خواهم کوئری‌ای بنویسید که به‌ازای هر برنامه کاربردی، ماکزیموم تعداد کاربری را که در یک لحظه در حال استفاده از آن هستند، نمایش دهد.

توجه داشته باشید که اگر به‌محض خاتمه یک Session، Session دیگری آغاز شود آن‌گاه این دو Session به‌هیچ عنوان هم‌زمان نخواهند بود.

با توجه به رکوردهای فرضی، نمودار زمانی ارائه سرویس توسط هر برنامه کاربردی به‌صورت زیر خواهد بود. در این نمودار حداکثر تعداد Sessionهای هم‌زمان با برنامه کاربردی App1 برابر با 4 و App2 برابر با 3 می‌باشد.



خروجی کوئری شما می‌بایست به‌صورت زیر باشد:

```
  App    Num_Concurrent
------- ---------------
  App1         4
  App2         3

(2 rows affected)
```