

# Creating and Tuning Indexes

CQG Ukraine Internship Program 2010



# Индексы – Теоретические основы

- Кучи и Индексы
- Кластерный индекс
- Не кластерный индекс
- Составной ключ
- Уникальные индексы
- Доступ к записям при наличии или отсутствии индексов
- Статистика и выбор индексов

# Кучи и Индексы

ROMEY		ANTON		MAISO		BOTTM		PERIC	
MORGK		FAMIA		BERGS		LINOD		BLAUS	
ANATR		SPLIR		LACOR		LONEP		SEVES	
TRADH		QUEDE		FISSA		CENTC		ISLAT	
GOURL		FRANR		SPEC D		BLONP		TRAIH	
EASTC		LILAS		GALED		PICCO		OTTIC	
LAMAI		HILAA		CONSH		MAGAA		QUICK	
Page 10		Page 11		Page 12		Page 13		Page 14	

`SELECT * FROM Customers WHERE CustomerID = "ROMEY"`  
(table scan – сканирование таблицы)

# Достоинства и недостатки индексов

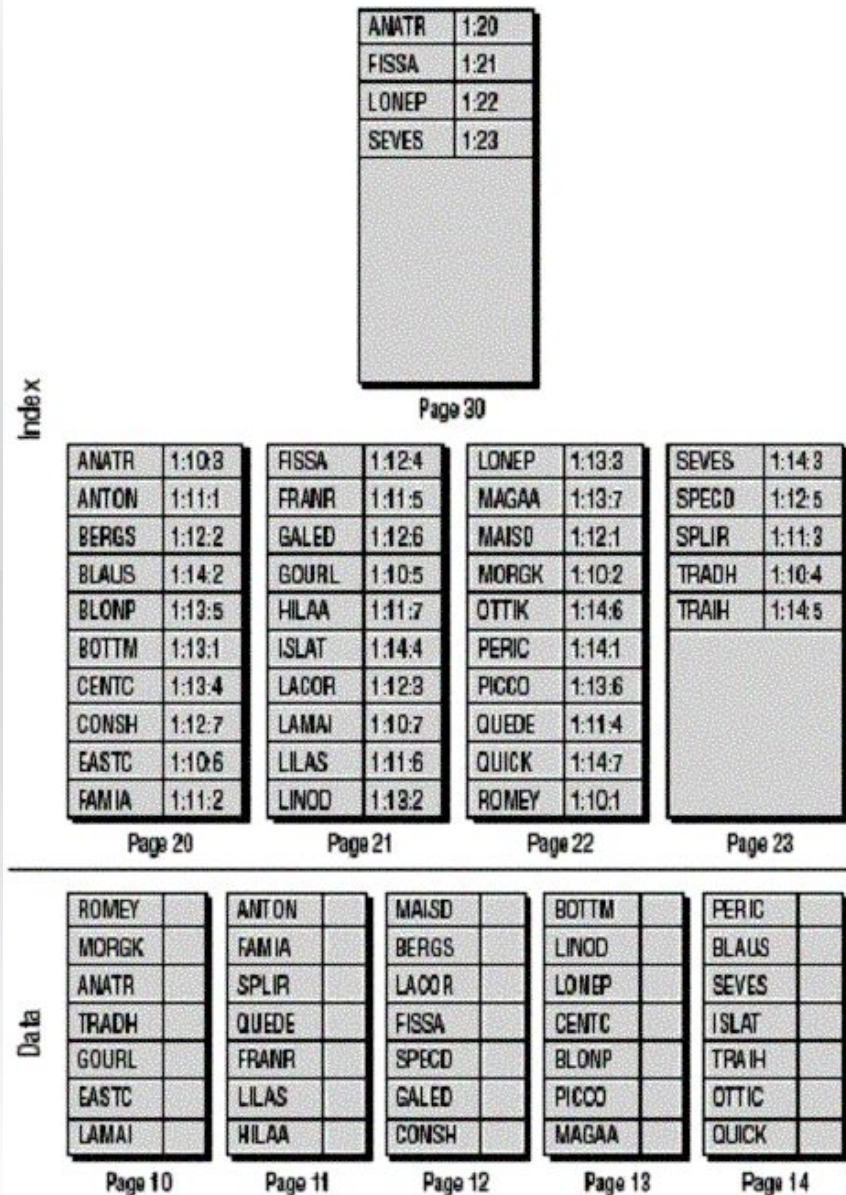
- **Функции**
  - *увеличение скорости доступа к данным*
  - *поддержка уникальности данных*
- **Недостатки**
  - *занимают дополнительное место*
  - *замедляются операции вставки, обновления и удаления записей*

## В-деревья (B-tree)

... Where CustomerID = 'ROMEY'

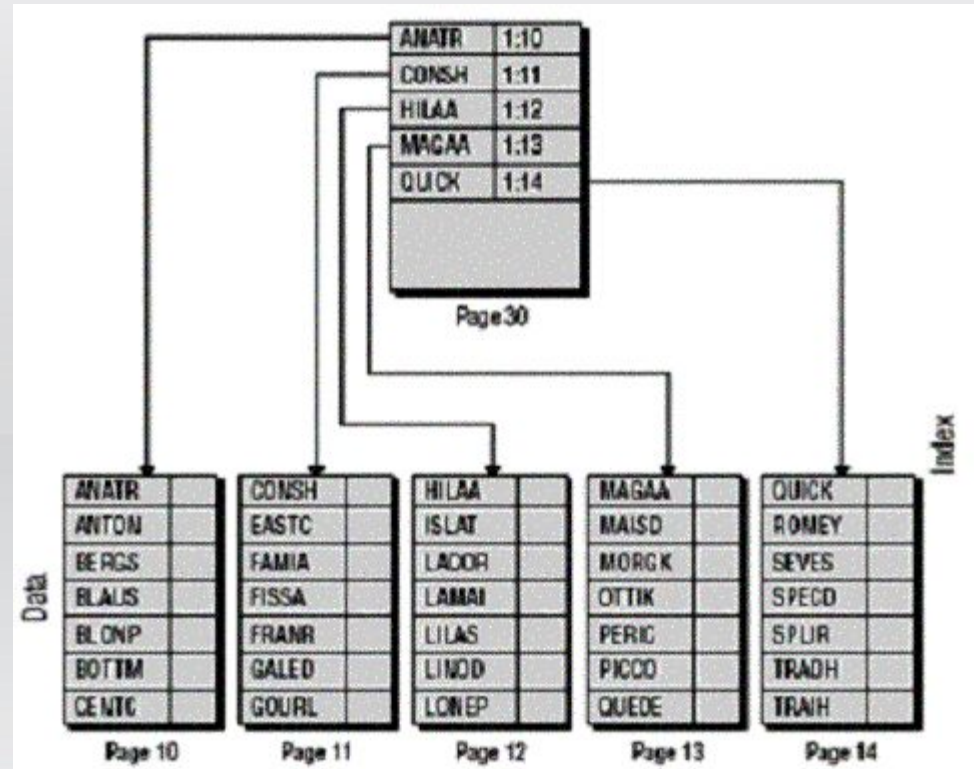
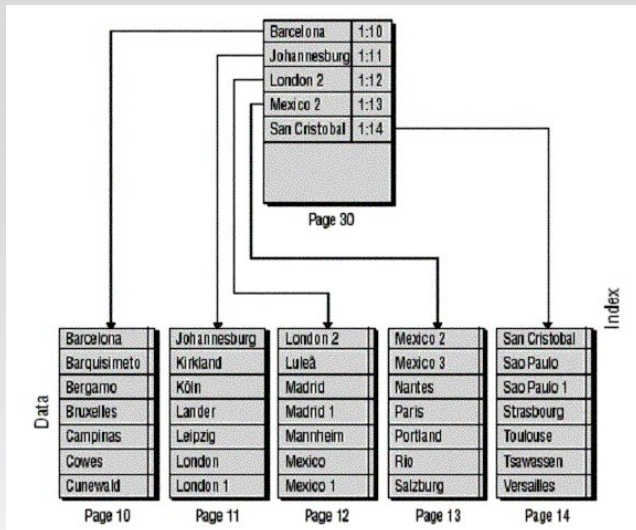
-будут прочитаны только  
страницы 30, 22 и 10 в  
указанном порядке

- Корневой уровень
- Промежуточный уровень
- Уровень листьев (leaf level) 20 -23
- Уровень данных
- ROW ID (ID 1:13:5)



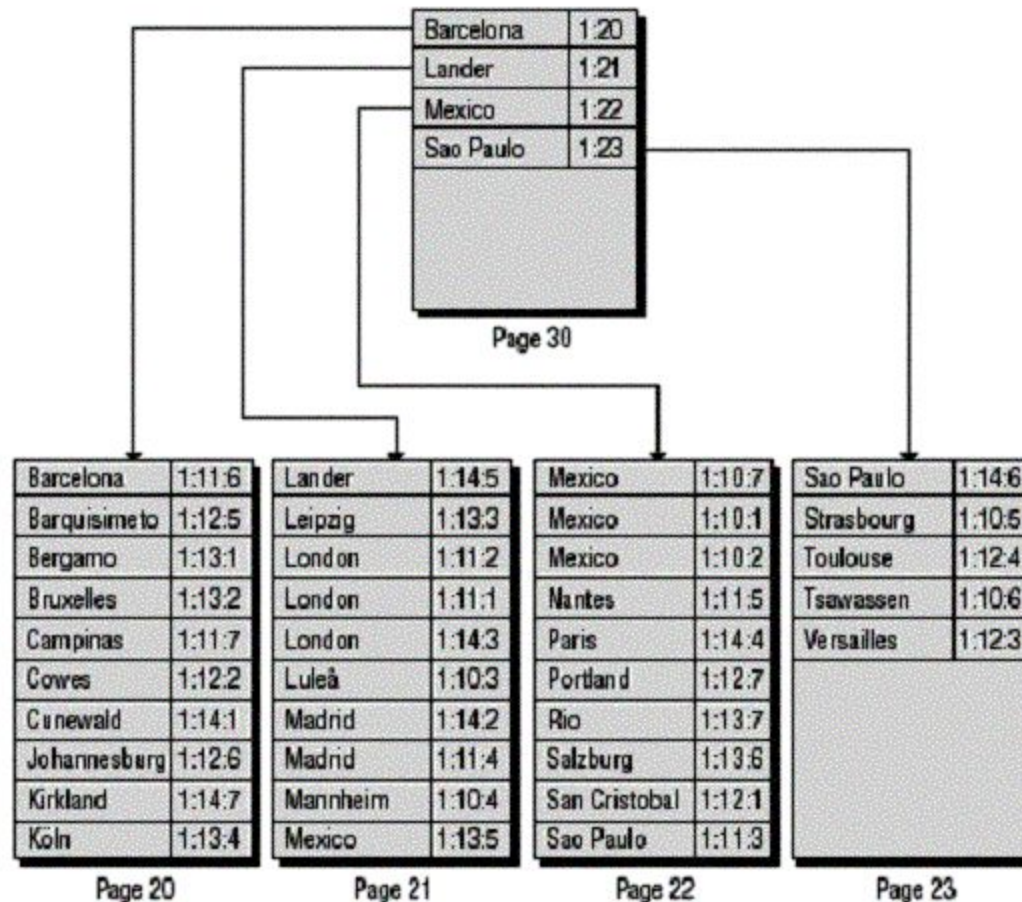
## Кластерный индекс

- leaf level этого индекса есть сами страницы таблицы с данными
- Может быть только один
- Является уникальным

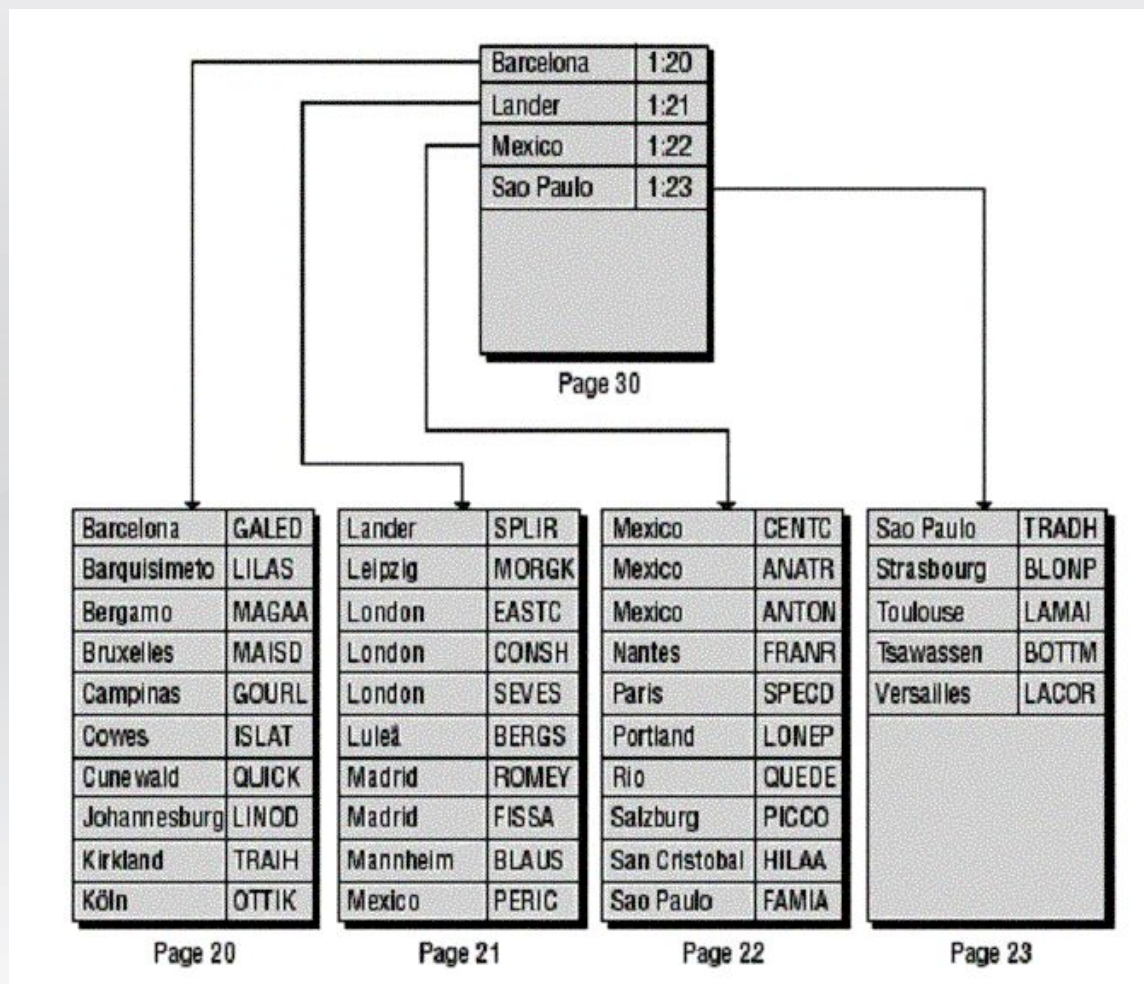




# Не кластерный индекс



# Некластерный индекс поверх кластерного





## Составной ключ

- Длина ключа индекса не должна превышать 900 байт
- 16 столбцов

## Уникальные индексы

- Unique constrain
- Primary key

# Доступ к записям при наличии или отсутствии индексов

- Сканирование таблицы.
- Выборка данных по кластерному индексу
- Выборка данных по не кластерному индексу



SELECT \* FROM Customers WHERE City = 'London'

Barcelona	1:20
Lander	1:21
Mexico	1:22
Sao Paulo	1:23

Page 30

Barcelona	1:11:6
Barquisimeto	1:12:5
Bergamo	1:13:1
Bruxelles	1:13:2
Campinas	1:11:7
Cowes	1:12:2
Cunewald	1:14:1
Johannesburg	1:12:6
Kirkland	1:14:7
Köln	1:13:4

Page 20

Lander	1:14:5
Leipzig	1:13:3
London	1:11:2
London	1:11:1
London	1:14:3
Luleå	1:10:3
Madrid	1:14:2
Madrid	1:11:4
Mannheim	1:10:4
Mexico	1:13:5

Page 21

Mexico	1:10:7
Mexico	1:10:1
Mexico	1:10:2
Nantes	1:11:5
Paris	1:14:4
Portland	1:12:7
Rio	1:13:7
Salzburg	1:13:6
San Cristobal	1:12:1
Sao Paulo	1:11:3

Page 22

Sao Paulo	1:14:6
Strasbourg	1:10:5
Toulouse	1:12:4
Tsawassen	1:10:6
Versailles	1:12:3

Page 23

Mexico	
Mexico	
Luleå	
Mannheim	
Strasbourg	
Tsawassen	
Mexico	

Page 10

London	
London	
Sao Paulo	
Madrid	
Nantes	
Barcelona	
Campinas	

Page 11

San Cristobal	
Cowes	
Versailles	
Toulouse	
Barquisimeto	
Johannesburg	
Portland	

Page 12

Bergamo	
Bruxelles	
Leipzig	
Köln	
Mexico	
Salzburg	
Rio	

Page 13

Cunewald	
Madrid	
London	
Paris	
Lander	
Sao Paulo	
Kirkland	

Page 14

SELECT \* FROM Customers WHERE City = 'London'

Barcelona	1:20
Lander	1:21
Mexico	1:22
Sao Paulo	1:23

Page 30

Barcelona	GALED
Barquisimeto	LILAS
Bergamo	MAGAA
Bruxelles	MAISD
Campinas	GOURL
Cowes	ISLAT
Cunewald	QUICK
Margaria	LINOD
Kirkland	TRAH
Köln	OTTIK

Page 20

Lander	SPLIR
Leipzig	MORGK
London	EASTC
London	CONSH
London	SEVES
Luleå	BERGS
Madrid	ROMEY
Madrid	FISSA
Mannheim	BLAUS
Mexico	PERIC

Page 21

Mexico	CENTC
Mexico	ANATR
Mexico	ANTON
Nantes	FRANR
Paris	SPECD
Portland	LONEP
Rio	QUEDE
Salzburg	PICCO
San Cristobal	HILAA
Sao Paulo	FAMIA

Page 22

Sao Paulo	TRADH
Strasbourg	BLONP
Toulouse	LAMAI
Tsawassen	BOTTM
Versailles	LACOR

Page 23

ANATR	1:10
CONSH	1:11
HILAA	1:12
MAGAA	1:13
QUICK	1:14

Page 31

ANATR	
ANTON	
BERGS	
BLAUS	
BLONP	
BOTTM	
CENTC	

Page 10

CONSH	
EASTC	
FAMIA	
FISSA	
FRANR	
GALED	
GOURL	

Page 11

HILAA	
ISLAT	
LACOR	
LAMAI	
LILAS	
LINOD	
LONEP	

Page 12

MAGAA	
MAISD	
MORGK	
OTTIK	
PERIC	
PICCO	
QUEDE	

Page 13

QUICK	
ROMEY	
SEVES	
SPLIR	
TRADH	
TRAH	

Page 14

# Создание индекса TSQL

- CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ]  
INDEX *index\_name* ON <object> ( *column* [ ASC | DESC  
][ ,...*n* ] ) [ INCLUDE ( *column\_name* [ ,...*n* ] ) ] [ WITH  
( <relational\_index\_option> [ ,...*n* ] ) ] [ ON {  
*partition\_scheme\_name* ( *column\_name* ) |  
*filegroup\_name* | default } ] [ ; ]

# Параметры

- ASC|DESC
- INCLUDE ( *column* [ ,... *n* ] ) – 1023 до 2 ГБ
- WITH
  - FILLFACTOR = *fillfactor*
  - IGNORE\_DUP\_KEY
  - SORT\_IN\_TEMPDB = { ON | OFF }
  - STATISTICS\_NORECOMPUTE = { ON | OFF }
  - DROP\_EXISTING = { ON | OFF }



# Информация об индексах

- sp\_helpindex 'Orders'
- SELECT indid, name, first, root, dpages, rowcnt FROM sysindexes WHERE id=OBJECT\_ID('Orders')

# Статистика и выбор индексов

- Что из себя представляет статистика
- `dbcc show_statistics`
- Выбор индексов
- Создание и обновление статистики

# Статистика

DBCC SHOW\_STATISTICS (N'Person', LastName)

Информация из заголовка статистик:

Name	Updated Rows	Rows Sampled	Steps	Density	Average key length	String Index
-----	-----	-----	-----	-----	-----	-----
_WA_Sys_00000002_1B29035F	Mar 25 2009 11:21AM	5 5	4	0	13.6	YES

Префикс набора полей и связанные с ним плотности и длина

All Density Average Length Columns

-----	-----	-----
0.25	13.6	LastName

Шаги гистограммы:

RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
-----	-----	-----	-----	-----

Andersen	0	2	0	0
Smith	0	1	0	1
Williams	0	1	0	1
Zhang	0	1	0	1

# работа со статистикой

- фоновое создание и обновление статистики ALTER DATABASE: AUTO\_CREATE\_STATISTICS и AUTO\_UPDATE\_STATISTICS; sp\_autostats; и опции NORECOMPUTE: CREATE STATISTICS и UPDATE STATISTICS)
- ручное управление статистикой (CREATE STATISTICS, UPDATE STATISTICS, DROP STATISTICS, CREATE INDEX, DROP INDEX)
- ручное создание статистики
- просмотр существующих объектов статистики таблицы или базы данных (sp\_helpstats, представления каталога sys.stats, sys.stats\_columns)
- включение/выключение автоматического, асинхронного обновления статистики (ALTER DATABASE, опция AUTO\_UPDATE\_STATISTICS\_ASYNC