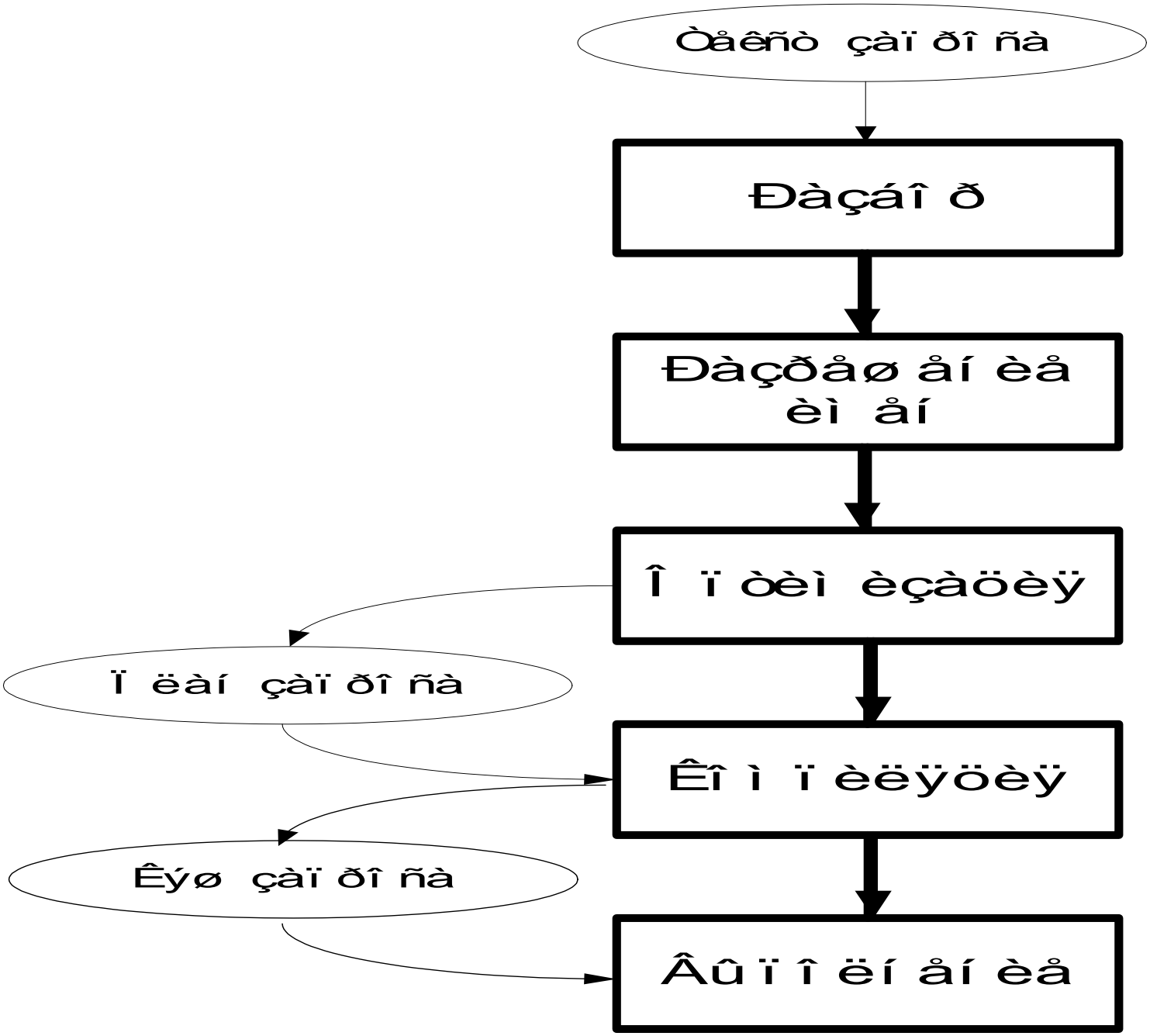
БАЗЫ ДАННЫХ

Лекция 9

Индексы


Òåêñò çàï ðî ñà
Ðàçáî ð
Ðàçðåø åí èå
èì åí
Î ï òèì èçàöèÿ
Ï ëàí çàï ðî ñà
Êî ì ï èëÿöèÿ
Êýø çàï ðî ñà
Âû ï î ëí åí èå

Применение индексов

- Индекс занимает определенный объем дискового пространства
- Для вставки и удаления данных необходимо обслуживание индекса
- Чем больше индексов имеет таблица, тем больше объем работы по их реорганизации
- **Правила:**
  - Выбирать индексы для частых запросов
  - Затем оценивать их использование
  - Не индексировать столбцы, по которым нет поиска

Оптимизация запроса

- анализ запроса
- выбор индекса
- выбор порядка выполнения операций соединения
- выбор метода выполнения операций соединения

Селективность и плотность

- Селективность запроса – соотношение количества строк, удовлетворяющих условию, к общему количеству строк в таблице
  - Индекс успешно работает при <= 5%.
  - Не нужен индекс при 80% или более
- Плотность запроса – количество возвращаемых строк запроса

Анализ запроса

- Наличие аргументов поиска
- Использование оператора OR
- Существование критериев соединения

Анализ запроса

- Аргумент поиска — это часть запроса, которая ограничивает промежуточный результирующий набор запроса:
- name = ‘Иванов С.П.'
- capacity >= 100
- name = ‘Иванов С.П.‘ AND idgroup = 512

Анализ запроса

- Нельзя использовать в качестве аргументов поиска:
  - Выражение с оператором отрицания NOT
  - <>
  - Выражение

- NOT IN('d1', 'd2')
- aud_no <> 9031
- capacity * 0.6 > 100

Создание индексов

- SELECT … WHERE column_name – для этого столбца следует создать индекс
- SELECT … WHERE column_name1 AND column_name2 – создать составной индекс по всем столбцам

Индексы при соединении

- Для каждого соединяемого столбца
- Некластеризованный индекс для столбца внешнего ключа

Выбор индексов

- Оптимизатор проверяет селективность выражения с индексированным столбцом, используя статистические данные
- AUTO_CREATE_STATISTICS ON(OFF)

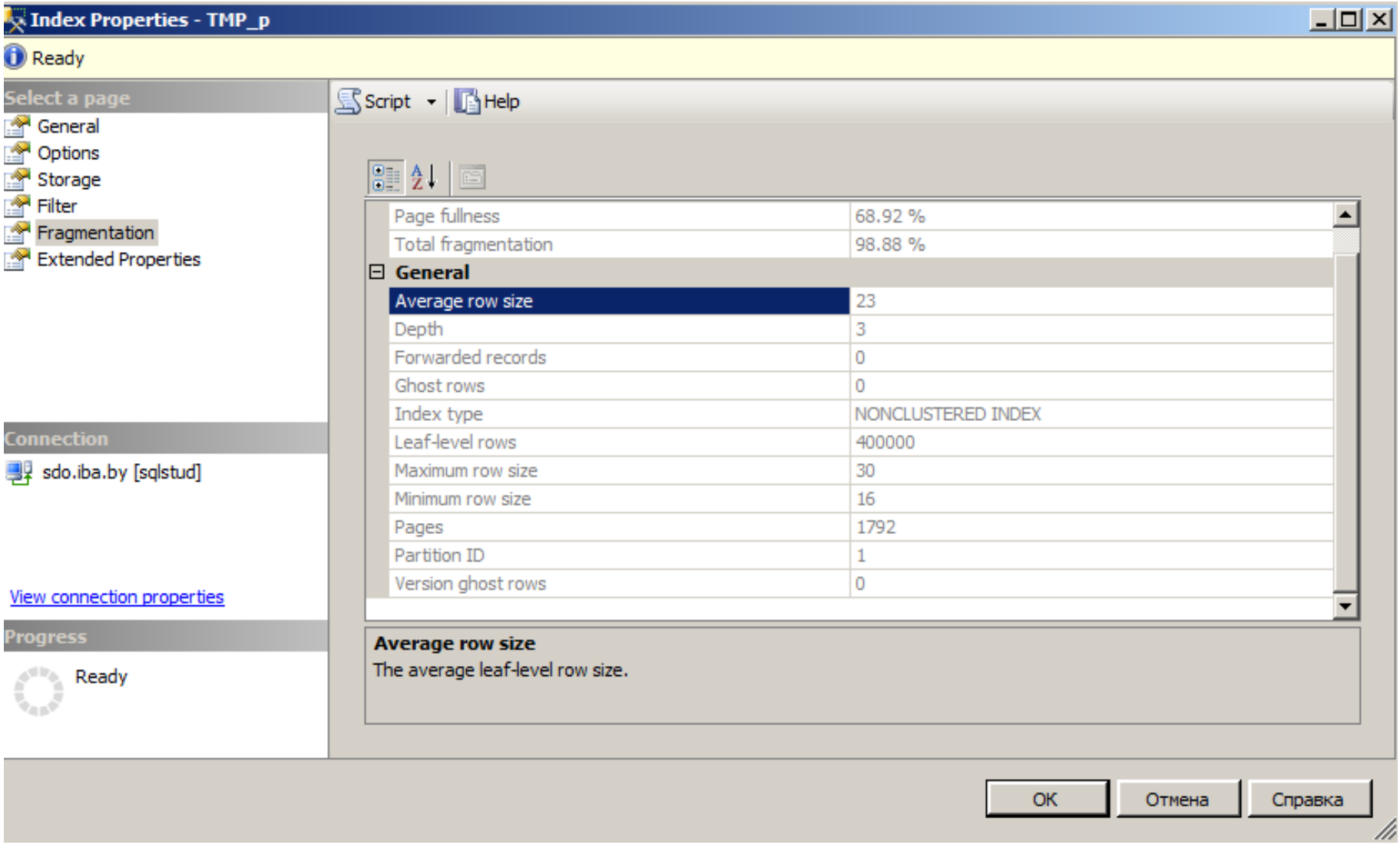
Фрагментация индекса

- Внутренняя фрагментация - объем данных, хранящихся в каждой странице
- Внешняя фрагментация - нарушение логического порядка страниц

```
CREATE TABLE test (p int);

SET NOCOUNT ON;
DECLARE @I INT = 0;
WHILE (@I<200000)
    BEGIN
    INSERT test (p) VALUES (FLOOR(30000*rand()));
    SET @I=@I+1;
    END;
```

```
CREATE NONCLUSTERED INDEX TMP_p ON test (p)
```



```sql
SELECT  a.index_id,
        name,
        avg_fragmentation_in_percent,
        fragment_count,
        avg_fragment_size_in_pages
FROM sys.dm_db_index_physical_stats (DB_ID('Test1'),
        OBJECT_ID('TMP_p'), NULL, NULL, NULL) AS a
JOIN sys.indexes AS b
ON a.object_id = b.object_id AND a.index_id = b.index_id;
```

	index_id	name	avg_fragmentation_in_percent	fragment_count	avg_fragment_size_in_pages
1	1	PK__PRODUCTS__59EE6463D5BD6110	0	1	1
2	1	PK__OFFICES__6A36566EE58513E5	0	1	1
3	1	PK__SALESREP__1D72C150D274422D	0	1	1
4	1	PK__CUSTOMER__6A01161A279E1D84	0	1	1
5	1	PK__ORDERS__6C473C958C5BCF61	0	1	1
6	0	NULL	0	0	0
7	0	NULL	19.6629213483146	38	36.9473684210526
8	2	TMP_p	98.8839285714286	1792	1

```sql
ALTER INDEX TMP_p ON test REORGANIZE;
```

Page fullness	99.45 %
Total fragmentation	1.13 %

General

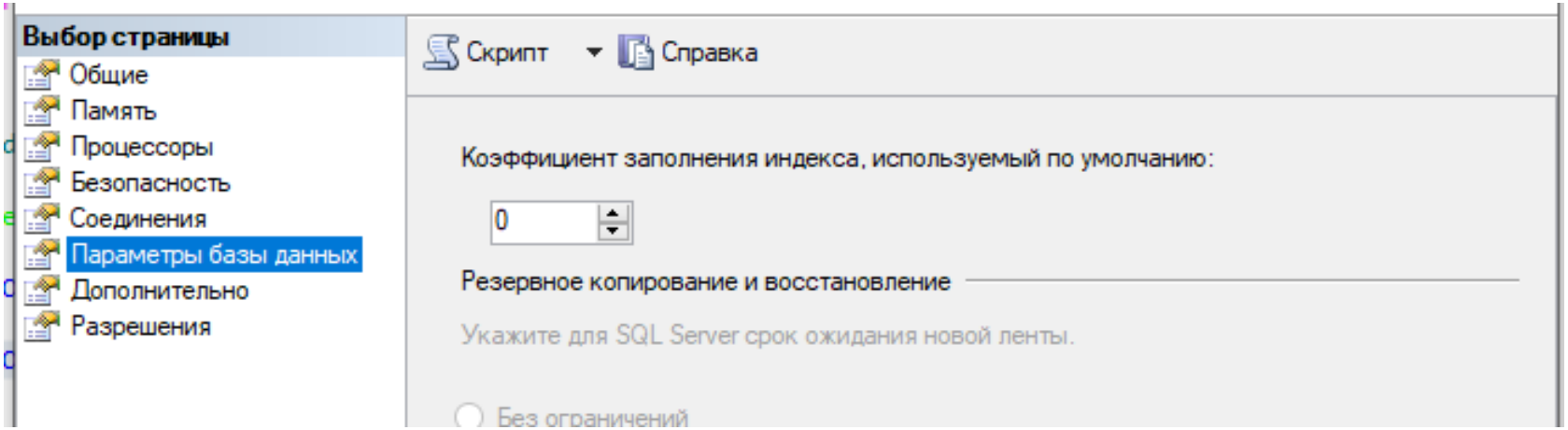
	index_id	name	avg_fragmentation_in_percent	fragment_count	avg_fragment_size_in_pages
1	1	PK__PRODUCTS__59EE6463D5BD6110	0	1	1
2	1	PK__OFFICES__6A36566EE58513E5	0	1	1
3	1	PK__SALESREP__1D72C150D274422D	0	1	1
4	1	PK__CUSTOMER__6A01161A279E1D84	0	1	1
5	1	PK__ORDERS__6C473C958C5BCF61	0	1	1
6	0	NULL	0	0	0
7	0	NULL	19.6629213483146	38	36.9473684210526
8	2	TMP_p	1.12721417069243	34	36.5294117647059

```sql
ALTER INDEX TMP_p ON test REBUILD WITH (ONLINE=OFF);
```

Page fullness	99.81 %
Total fragmentation	0.11 %

General

	index_id	name	avg_fragmentation_in_percent	fragment_count	avg_fragment_size_in_pages
1	1	PK__PRODUCTS__59EE6463D5BD6110	0	1	1
2	1	PK__OFFICES__6A36566EE58513E5	0	1	1
3	1	PK__SALESREP__1D72C150D274422D	0	1	1
4	1	PK__CUSTOMER__6A01161A279E1D84	0	1	1
5	1	PK__ORDERS__6C473C958C5BCF61	0	1	1
6	0	NULL	0	0	0
7	0	NULL	19.6629213483146	38	36.9473684210526
8	2	TMP_p	0.112233445566779	4	222.75




```sql
CREATE TABLE test (p int);

SET NOCOUNT ON;
DECLARE @I INT = 0;
WHILE (@I<20000)
    BEGIN
    INSERT test (p) VALUES (FLOOR(30000*rand()));
    SET @I=@I+1;
    END;

CREATE NONCLUSTERED INDEX ind_for_rebuild ON test(p)

SELECT  * FROM SYS.dm_db_index_physical_stats(DB_ID('Test'), OBJECT_ID('Test'), NULL,NULL,NULL)

ALTER INDEX ind_for_rebuild ON test REORGANIZE

ALTER INDEX ind_for_rebuild ON test REBUILD WITH (FILLFACTOR = 80);
```

Фрагментация	
Заполненность страниц	65,77 %
Общая фрагментация	67,75 %

Фрагментация	
Заполненность страниц	79,97 %
Общая фрагментация	0,00 %

Изменение индекса

- FillFactor указывает, сколько листовых страниц индекса заполняется
- При высоком FillFactor больше строк помещается в одну страницу данных, но может стать больше разбиений страниц
- При низком FillFactor на странице данных меньше записей, что снизит число разбиений страниц, но потребуется больше операций чтения, так как данные будут распределены по большему числу страниц

Изменение индекса

- ALTER INDEX
- ALLOW_ROW_LOCKS, ALLOW_PAGE_LOCKS, IGNORE_DUP_KEY
- REBUILD - пересоздание индекса
- REORGANIZE - реорганизация страниц листьев индекса
- DISABLE - отключение индекса

DISABLE

- DISABLE отключает указанный индекс
- Отключенный индекс недоступен, пока он не будет снова включен
- REBUILD
- При отключенном кластеризованном индексе таблицы данные этой таблицы будут недоступны, так как все страницы данных таблицы с кластеризованным индексом хранятся в его листьях дерева