Wprowadzenie do języka T-SQL

Michał Bleja

Składnia polecenia SELECT

```
SELECT ALL|DISTINCT lista_select
FROM lista_from
WHERE warunki_selekcji_dla_wierszy
GROUP BY wyrażenie
HAVING warunki_selekcji_dla_grup
ORDER BY wyrażenie ASC|DSC
```

Przykłady poleceń SELECT

- W ramach wykładu użyto bazy AdventureWorks firmy Microsoft
- `SELECT * FROM Person.Person`
-
```
SELECT FirstName, LastName, Title
FROM Person.Person
```
-
```
SELECT FirstName, LastName, Title
FROM Person.Person
WHERE LastName LIKE '_o%' AND Title IS NOT NULL
```
- _ - dowolny jeden znak, % - 0 lub więcej znaków

Złączenia (JOINS)

```
--iloczyn kartezjański (cross join)
SELECT *
FROM Person.Person, HumanResources.Employee

--złączenie wewnętrzne (inner join)
SELECT p.FirstName, p.LastName, e.JobTitle,
        e.HireDate
  FROM Person.Person p JOIN
        HumanResources.Employee e
      ON p.BusinessEntityID=e.BusinessEntityID

--lewe złączenie (left outer join), uwzględnia
rekordy z lewej tabeli nie mające odpowiedników w
prawej tabeli
SELECT p.FirstName, p.LastName, e.JobTitle,
        e.HireDate
  FROM Person.Person p LEFT JOIN
        HumanResources.Employee e
```

Klauzula ORDER BY

```
.--sortowanie w porządku rosnącym (asc, ascending
order)
SELECT FirstName, LastName, Title
FROM Person.Person
ORDER BY LastName
.--sortowanie w porządku malejącym (desc,
descending order)
SELECT FirstName, LastName, Title
FROM Person.Person
WHERE Title IS NOT NULL
```

Funkcje agregujące

- Zwracają wiersz podsumowania dla każdej grupy wierszy
- Stosowane często z klauzulą GROUP BY. Bez tej klauzuli wszystkie wiersze stanowią jedną grupę i funkcja zwraca jeden wiersz podsumowania
- AVG – zwraca wartość średnią z danego zbioru
- MIN, MAX – zwraca odpowiednio wartość najmniejszą i największą z danego zbioru
- SUM - oblicza sumę elementów z danego zbioru
- COUNT – zwraca liczbę elemntów w zbiorze

Funkcje agregujące

```
SELECT COUNT(*)
FROM Person.Person
SELECT COUNT(distinct FirstName)
FROM Person.Person
SELECT AVG(Rate),SUM(Rate)
FROM HumanResources.EmployeePayHistory
SELECT MIN(HireDate), MAX(HireDate)
FROM HumanResources.Employee
WHERE gender = 'M'
```

Klauzula GROUP BY

```
--GROUP BY pozwala podzielić wiersze zwrócone
przez zapytanie na grupy, w ramach każdej grupy
liczona jest wówczas funkcja agregująca
SELECT DepartmentID, COUNT(*)
FROM HumanResources.EmployeeDepartmentHistory
WHERE EndDate IS NULL
GROUP BY DepartmentID
```

- **UWAGA** – wszystkie wyrażenia po SELECT z wyjątkiem funkcji agregujących muszą zawsze wystąpić po GROUP BY

Klauzula HAVING

```
--HAVING pozwala dokonać operacji selekcji na grupach
SELECT DepartmentID, COUNT(*)
FROM HumanResources.EmployeeDepartmentHistory
WHERE EndDate IS NULL
GROUP BY DepartmentID
HAVING COUNT(*)>100
```

Zapytania zagnieżdżone (nested queries)

- --zapytanie zagnieżdżone to instrukcja select, która zawiera w sobie inną instrukcję select (zwaną podzapytaniem)

- Zapytania niezależne (nieskorelowane) – zapytanie wewnętrzne nie odwołuje się do zapytania zewnętrznego

```
SELECT p.FirstName, p.LastName
FROM Person.Person p JOIN
        HumanResources.Employee e
      ON p.BusinessEntityID=e.BusinessEntityID
  WHERE  e.BirthDate =
                  (SELECT MIN(BirthDate)
                   FROM HumanResources.Employee)
```

Zapytania zagnieżdżone (nested queries)

- Zapytania skorelowane – zapytanie wewnętrzne odwołuje się do zapytania zewnętrznego

```
SELECT p.FirstName, p.LastName
FROM Person.Person p JOIN
        HumanResources.Employee e
      ON p.BusinessEntityID=e.BusinessEntityID
  WHERE  e.BirthDate =
                  (SELECT MIN(BirthDate)
                   FROM HumanResources.Employee
                   WHERE gender=e.gender)
```

Podstawowe typy danych

Znakowe typy danych
CHAR(rozmiar), NCHAR(rozmiar)
VARCHAR(rozmiar), NVARCHAR(rozmiar)

Liczbowe typy danych
INT, SMALLINT, FLOAT, REAL, NUMERIC(p, [s])

Typy pieniężne
MONEY, SMALLMONEY

Typy dla daty i czasu
DATE, TIME, DATETIME

Instrukcja - CREATE TABLE

```
CREATE TABLE Employee(
  Emp_id int IDENTITY(1,1)
              CONSTRAINT PK_Emp_id PRIMARY KEY,
  Code char(5) CONSTRAINT U_Code UNIQUE,
  FirstName varchar(25) NOT NULL,
  LastName varchar(30) NOT NULL,
  BirthDate date NOT NULL,
  Gender char(1) CONSTRAINT CH_Gen
          CHECK(gender='M' OR gender='F'),
  Manager_id int,
   CONSTRAINT FK_Mgr_id FOREIGN KEY(manager_id)
        REFERENCES Employee(emp_id))
```

Ograniczenia integralnościowe

- *PRIMARY KEY* – wartości wstawiane do kolumny muszą być unikalne, niedozowolone są wartości NULL

```
Emp_id int IDENTITY(1,1)
                    CONSTRAINT PK_Emp_id PRIMARY KEY
```

- *FOREIGN KEY* – wartości wstawiane do kolumny muszą być zgodne z wartościami kolumny klucza głównego, do którego klucz obcy się odnosi

```
CONSTRAINT FK_Mgr_id FOREIGN KEY(manager_id)
          REFERENCES Employee(emp_id)
```

- *UNIQUE* - wartości wstawiane do kolumny muszą być unikalne

```
Code char(5) CONSTRAINT U_Code UNIQUE
```

- *CHECK* – wartości wstawiane do kolumny muszą spełniać określone warunki

```
Gender char(1) CONSTRAINT CH_Gen
             CHECK(gender='M' OR gender='F')
```

- *NOT NULL* –wartości NULL są niedozwolone

ALTER TABLE – modyfikacja tabel

```
CREATE TABLE Phone(
phone_id int IDENTITY(1,1) NOT NULL,
number varchar(20) NOT NULL)

--Dodanie kolumny
ALTER TABLE Phone
ADD type varchar(2)

--Modyfikacja kolumny
ALTER TABLE Phone
ALTER COLUMN type char(1) NOT NULL

--Dodanie ograniczenia
ALTER TABLE Phone
ADD CONSTRAINT PK_APhone_id
                    PRIMARY KEY(phone_id)
```

Tworzenie baz danych w SQL Server

Michał Bleja

Definicja bazy danych

Baza danych – zbiór plików przechowywanych na dysku.

SQL Server wymaga co najmniej dwóch plików:

- Pliku danych (ang. primary data file, *.mdf) – przechowuje informacje startowe dla bazy, dane i obiekty użytkownika, wskazuje pozostałe pliki danych (ang. secondonary data files, *.ndf)
- Pliku logu (ang. log file, *.ldf) – rejestruje wszystkie zmiany na bazie danych - może rosnąć do nieskończoności

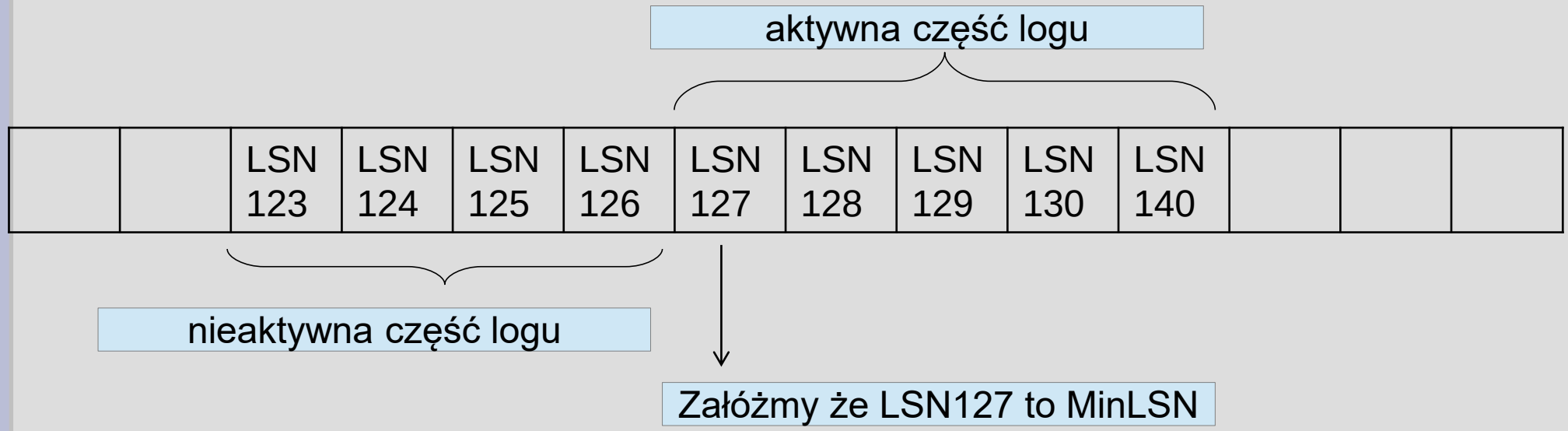
Strony i ekstenty

- Strona ma rozmiar 8kB i jest podstawową jednostką alokacji (SQL Server odczytuje i zapisuje całe strony)
- Pliki danych zawierają strony ponumerowane od *0* do *n*
- Ekstent składa się z 8 ciągłych stron (ma więc rozmiar 64kB)
- Eksteny są stosowane w celu ułatwienia zarządzania stronami
- Ekstent może składować dane jego obiektu
- Każda strona ekstentu może również składować dane innego obiektu

Plik logu

Plik logu jest zapisywany w porządku chronologicznym (w sposób kolisty). Każdy wpis w logu ma swój unikalny numer LSN (ang. log sequence number). Numery nadawane są w porządku rosnącym.

MinLSN – numer najstarszej aktywnej transakcji



Pliki danych (zawiera dane składowane w bazie)

Zalety stosowania kilku plików danych:

- Pliki można przechowywać na klilku dyskach celem przyspieszenia operacji I/O
- Można utworzyć macierz RAID i zlecić macirzy optymalizację I/O
- Dwa pliki wymuszają dwa wątki, co implikuje użycie dwóch rdzeni procesora
- Łatwiej zarządzać kopiami zapasowymi

Grupy plików

- Grupa plików – logiczna struktura, która obejmuje zestaw plików
- W bazie zawsze istnieje grupa PRIMARY (zawiera główny plik bazy danych, wszystkie strony systemowych obiektów)

Można tworzyć dodatkowe grupy plików.

Zalety stosowania grup plików:

- Pozwalają umieszczać obiekty i dane w odpowiednich plikach
- Ułatwiają wykonywanie kopii zapasowych i operacje odtwarzania

Tworzenie plików danych

- Należy wyspecyfikować rozmiar i nazwę pliku
- Można wskazać, że SQL Server będzie automatycznie powiększał plik o określoną wartość
- Plik zostaje przypisany do odpowiedniej grupy plików

Tworzenie bazy danych bez podania plików

- Plik danych ma rozmiar pliku danych bazy *model*
- Plik logu ma rozmiar równy max{512kB, 25%*size(plik danych)}

```
CREATE DATABASE demo1DB;


--Sprawdź nazwy i rozmiary plików

SELECT name, size*1.0/128
FROM sys.master_files
WHERE name = N'demo1DB';
GO
```

Tworzenie bazy danych

```
CREATE DATABASE demo2DB
ON
   (NAME = demo2DB,
    FILENAME = 'D:\demo2DB\data\demo2DB.mdf',
    SIZE = 3072KB , FILEGROWTH = 1024KB )
LOG ON
   (NAME = demo2DB_log,
    FILENAME = 'D:\demo2DB\log\demo2DB_log.ldf',
    SIZE = 1024KB , FILEGROWTH = 10%)
```

Tworzenie bazy danych z dwoma grupami plików

```
CREATE DATABASE demo3DB
ON
PRIMARY
 (NAME = demo3DB1,
  FILENAME = 'D:\demo3DB\data\demo3DB1.mdf',
  SIZE = 10240KB , FILEGROWTH = 0),
FILEGROUP FG2
 (NAME = demo3DB2,
  FILENAME = N'D:\demo3DB\data\demo3DB2.ndf',
  SIZE = 10240KB , FILEGROWTH = 1024KB )
LOG ON
 (NAME = demo3DB_log,
  FILENAME = 'D:\demo3DB\log\demo3DB_log.ldf',
  SIZE = 5120KB , FILEGROWTH = 10%)
```

Dodawanie plików i grup plików

```
ALTER DATABASE demo3DB
 ADD FILEGROUP FG3


ALTER DATABASE demo3DB
 ADD FILE (NAME = demoDB3,
  FILENAME = 'D:\demo3DB\data\demo3DB3.ndf' ,
  SIZE = 5120KB , FILEGROWTH = 1024KB )
  TO FILEGROUP FG3

--FG2 będzie domyślną grupą plików
ALTER DATABASE demo3DB
MODIFY FILEGROUP FG2
DEFAULT
```

Opcje bazy danych – model odzyskiwania

```
ALTER DATABASE demo3DB
SET RECOVERY FULL
```

Model odzyskiwania (ang. recovery model)

- FULL – Wszystkie zmiany są rejestrowane w logu (wmaga kopii zapasowych logu, można odtworzyć bazę do punktu w czasie, w szczególności do momentu awarii)
- SIMPLE – Wszystkie zmiany są rejestrowane w logu (CHECKPOINT usuwa nieaktywną część logu, brak możliwości wykonania kopii logu, zmiany od ostatniej pełnej kopii nie są chronione)

Opcje bazy danych – model odzyskiwania

- BULK-LOGGED – ogranicza rejestrowanie operacji masowych (stosowany np. przy dużych wsadach danych). Nie można odtworzyć bazy do punktu w czasie, w którym nastąpiło ograniczone rejstrowanie, wymagane są kopie logu

Opcje bazy danych

- PAGE_VERIFY CHECKSUM – przed zapisem strony na dysk liczona jest jej suma kontrolna i zapisywana w nagłówku. Po odczytaniu strony SQL Server ponownie oblicza sumę i porównuje z wartością zapisaną w nagłówku. Informacje o błędnych stronach są w tabeli msdb.dbo.suspect_pages.
- AUTO_CLOSE – implikuje zamknięcie bazy i zwolnienie jej zasobów po zakończeniu wszystkich połączeń
- AUTO_SHRINK – implikuje automatyczne zmniejszanie plików, w których ilość wolnej przestrzeni przekracza 25%

Opcje bazy danych

- AUTO_CREATE_STATISTICS – statystyki na kolumnach występujących w predykatach będą generowane automatycznie dla optymalizatora
- AUTO_UPDATE_STATISTICS – przestarzałe statystyki dla optymalizatora będą uaktualniane automatycznie
- AUTO_UPDATE_STATISTICS_ASYNC – określa czy statystyki są uaktualniane synchronicznie lub asynchronicznie (optymalizator nie czeka na aktualne statystyki przed kompilacją zapytania)

Opcje bazy danych - Collation

- COLLATION – określa sekwencję porządkową dla bazy danych.
- Zachodzi hierarchia dziedziczenia collation: instancja-baza-tabela-kolumna.
- Przykład collection: Polish_CS_AS
- CS – case sensitive (a ≠ A)
- CI – case insensitive (a = A

Systemowe bazy danych

- **master** – przechowuje informacje systemowe niezbędne do pracy instancji SQL Server (np. informacje o innych bazach, lokalizacje ich plików, konta logowania, serwery dołączone)
- **resource** – ukryta baza tylko do odczytu, która zawiera obiekty systemowe (ukazują się one w każdej w bazie w schemie sys)
- **model** – szablon dla bazy danych tworzonych w instancji
- **msdb** – używana przez SQL Server Agent (składuje zadania, alerty, ich harmonogramy), przechowuje historię kopii zapasowych i odtwarzań
- **tempdb** – przestrzeń do składowania tymczasowych i pośrednich wyników, tabel i zmiennych tymczasowych

Przenoszenie bazy danych użytkownika

Rozważmy poniższą bazę:

```
CREATE DATABASE demoDB ON
PRIMARY
 ( NAME = demoDB1, FILENAME = 'd:\data\demoDB1.mdf'
, SIZE = 6144KB , FILEGROWTH = 1024KB ),
FILEGROUP FG1
 ( NAME = demoDB2, FILENAME = 'd:\data\demoDB2.ndf'
   , SIZE = 6144KB , FILEGROWTH = 1024KB )
LOG ON
( NAME = demoDB_log,
   FILENAME = 'd:\data\demoDB_log.ldf' ,
      SIZE = 1024KB , FILEGROWTH = 10%)
```

Przenieśmy ją w tryb OFFLINE

```
ALTER DATABASE demoDB SET OFFLINE WITH ROLLBACK
IMMEDIATE
```

Przenoszenie bazy danych użytkownika

- Przenieść pliki bazy danych do nowej lokalizacji
- Po przensiesieniu dla każdego pliku bazy danych należy wykonać poniższe polecenie:

```
ALTER DATABASE nazwa_bazy
MODIFY FILE
  ( NAME = nazwa_logiczna,
      FILENAME = 'nowa_ścieżka_do_pliku')
```

- Przenieść bazę w tryb ONLINE
- Sprawdzić czy lokalizacje plików zostały zmienione

```
SELECT name, physical_name, state_desc
FROM sys.master_files
WHERE database_id = DB_ID('nazwa_bazy')
```

Przenoszenie systemowych baz danych (nie dotyczy master i resource)

- Zatrzymać instancje - NET STOP MSSQLSERVER
- NET START MSSQLSERVER /f /T3608
- Uruchomić sqlcmd i zlokalizować pliki bazy

```
SELECT name, physical_name
   FROM sys.master_files
   WHERE database_id = DB_ID('nazwa_bazy')
```

- Dla każdego pliku bazy danych należy wykonać poniższe polecenie:

```
   ALTER DATABASE nazwa_bazy
   MODIFY FILE
  ( NAME = nazwa_logiczna,
      FILENAME = 'nowa_ścieżka_do_pliku')
```

- Wykonać shutdown instancji
- Przenieść pliki do nowych lokalizacji
- Uruchomić instancję
- Sprawdzić czy lokalizacje plików zostały zmienione

Przenoszenie bazy danych master

- Uruchomić SQL Server Configuration Manager
- Zmienić ścieżki do plików bazy master (*Properties-Advanced-Startup parameters lub Properties-Startup parameters*)
- Zatrzymać instancję SQL Server
- Przenieść pliki do nowych lokalizacji
- Uruchomić instancję
- Sprawdzić czy lokalizacje plików zostały zmienione

Systemowej bazy danych **resource** nie można przenosić. Jej pliki są składowane w ...\Microsoft SQL Server\...\MSSQL\Binn\