Vysoká škola báňská – Technická univerzita Ostrava
Fakulta strojní, Katedra automatizační techniky a řízení

6

Informační systémy

2008/2009

Radim Farana

Obsah

- Jazyk SQL,
  - DML, pokročilé agregace,
  - výkonné dotazy,
  - transformace dat,
- Doporučená literatura:
  Gruber, M. *Mistrovství v SQL*. Svazek 1. Praha : SoftPress s.r.o., 2004. ISBN 80-86497-62-3

Agregace výrazů

- SUM(ks*cena) AS platba
- SUM(cena*0.19) AS DPH
- Raději
- SUM(cena)*0.19 AS DPH
- Výpočet je proveden jen jednou

Vnořování agregátů

- SELECT p_datum, MAX(SUM(p_kusu)) FROM prace GROUP BY p_datum;
- Vnoření není dovoleno, nutno změnit přístup, např:
- SELECT TOP 1 p_datum, SUM(p_kusu) FROM prace GROUP BY p_datum ORDER BY SUM(p_kusu) DESC;

Počet unikátních hodnot

- COUNT(položka)
  - Počet neprázdných hodnot položky
- COUNT(DISTINCT položka)
  - Počet různých neprázdných hodnot položky
- COUNT(*)
  - Počet záznamů – včetně prázdných

Transformace dat

- CASE – výběr hodnot,
- NULLIF – test shody hodnot,
- COALESCE – eliminace NULL,
- CAST – změna datového typu,
- Vestavěné funkce

CASE

- CASE výraz
  WHEN hodnota THEN výstup
- CASE výraz
  WHEN hodnota THEN výstup1
  ELSE výstup2

- CASE p_kusu WHEN 1 THEN 'jedna,
  WHEN 2 THEN 'dva' ELSE 'více

CASE

- CASE
  WHEN výraz=hodnota THEN výstup
- CASE
  WHEN výraz=hodnota THEN výstup1
  ELSE výstup2

- CASE WHEN p_kusu<5 THEN 'málo,
  WHEN p_kusu=5 THEN 'akorát,
  ELSE 'více'

NULLIF

- NULLIF(výraz, hodnota)
- Pokud je výraz=hodnota, vrací NULL, jinak vrací výsledek výrazu.

- NULLIF(p_kusu, 5)

COALESCE

- COLEASCE(výraz, výraz, …)
- Vrací první výraz, který není NULL

- SELECT
  COALESCE(p_pracovnik, pr_OSC), *
  FROM pracovnici LEFT JOIN prace
  ON pr_OSC=p_pracovnik;

Konverze datových typů

- CAST (výraz AS datový typ)

- CAST ('1/13/2006' AS DATETIME)
- CAST (cena*0.19 AS INTEGER)

- CONVERT(datový typ, výraz)
  - širší možnosti nad standard SQL

Vestavěné funkce

- Práce s řetězci.
- Práce s datumy a časem.
- Matematické funkce.
- Systémové funkce.
- Ostatní funkce.

Práce s řetězci

Function	Description	Example
LCASE( )[1], **LOWER( )**	Converts strings to lowercase	`SELECT UPPER(substring(lname, 1, 1)) + LOWER(substring (lname, 2, 99)) FROM employee` Displays a last name after the first character is converted to uppercase and the remaining characters to lowercase.
LTRIM( )	Removes leading spaces from a string	`SELECT stor_name, LTRIM(stor_address) FROM stores` Displays an address column after extraneous spaces are removed from the front.
SUBSTRING( )	Extracts one or more characters from a string	`SELECT SUBSTRING(phone,1,3) FROM employee` Displays the first three characters (the area code) of a phone number.
UCASE( )[1], **UPPER( )**	Converts strings to uppercase	`SELECT * FROM employee WHERE UPPER(lname) = 'SMITH'` Converts the contents of the lname column to uppercase before comparing them to a specific value (avoids mismatches if the search is case sensitive). For details about case sensitivity in SQL Server, see Query Designer Considerations .

Práce s datumy a časem

Function	Description	Example
DATEDIFF( )	Calculates an interval between two dates.	`SELECT fname, lname, hire_date FROM employee WHERE DATEDIFF(year, hire_date, getdate()) > 5` Locates all employees hired more than five years ago.
DATEPART( )	Returns the specified portion of a date or datetime column, including the day, month, or year.	`SELECT DATEPART(year, hire_date) FROM employee` Displays only the year in which an employee was hired (not the full date).
CURDATE( )[1], **GETDATE( )** or **DATE( )**	Returns the current date in datetime format. This function is useful as input for many other date functions, such as calculating an interval forward or backward from today.	`SELECT order_id FROM orders WHERE order_date = GETDATE() ` Displays orders placed today.

Matematické funkce

Function	Description	Example
ROUND( )	Rounds a number off to the specified number of decimal places	`SELECT ROUND(qty * (price * discount), 2) FROM sales` Displays a total price based on a discount, then rounds the results off to two decimal places.
FLOOR( )	Rounds a number down to the nearest (smallest) whole number	`UPDATE titles SET price = FLOOR(price)` Rounds all prices in the titles table down to the nearest whole number.
CEILING( )	Rounds a number up to the nearest whole number	`INSERT INTO archivetitle SELECT title, CEILING(price) FROM titles` Copies the title and the price (rounded up to the nearest integer) from the titles table to the archivetitle table.

Systémové funkce

Function	Description	Example
DATALENGTH( )	Returns the number of bytes used by the specified expression	`SELECT DATALENGTH(au_lname + ', ' + au_fname) FROM authors` Lists the number of bytes required for the combination of last and first names.
USER( )[1], **USER_NAME( )**	Returns the current user name	`SELECT company_name, city, phone FROM customers WHERE salesperson = USER_NAME()` Creates a list of customers for the salesperson who runs the query.

Ostatní funkce

Function	Description	Example
CONVERT( )	Converts data from one data type into another. Useful to format data or to use the contents of a data column as an argument in a function that requires a different data type.	`SELECT 'Hired: ' + CONVERT(char (11), hire_date) FROM employee` Displays a date with a caption in front of it; the CONVERT( ) function creates a string out of the date so that it can be concatenated with a literal string.
SOUNDEX( )	Returns the Soundex code for the specified expression, which you can use to create "sounds like" searches.	`SELECT au_lname, au_fname FROM authors WHERE SOUNDEX(au_fname) = 'M240'` Searches for names that sound like "Michael".
STR( )	Converts numeric data into a character string so you can manipulate it with text operators.	`SELECT str(job_id) + ' ' + str(job_lvl) FROM employee` Displays the job_id and job_lvl columns (both numeric) in a single string.

Příklad

- Určení doby v hodinách, minutách a sekundách mezi začátkem a koncem akce
- SELECT Sessions.*,
  ROUND(CAST([SLastTime]-[SStartTime] As Float)* 24, 0, 1) AS SHours,
  ROUND((CAST([SLastTime]-[SStartTime] As Float)* 24 - ROUND(CAST([SLastTime]-[SStartTime] As Float)* 24, 0, 1))*60, 0, 1) AS SMinutes,
  (CAST([SLastTime]-[SStartTime] As Float)*24*60 - ROUND(CAST([SLastTime]-[SStartTime] As Float)*24*60, 0, 1))*60 AS SSeconds
  FROM Sessions ORDER BY SStartTime DESC;

S využitím agregace

- Určení doby v hodinách, minutách a sekundách mezi začátkya konci akcí v součtu za celý měsíc (a rok):
- SELECT Year([SStartTime]) AS SYear, Month([SStartTime]) AS SMonth,
  ROUND(Sum(CAST([SLastTime]-[SStartTime] AS Float))* 24, 0, 1) AS SHours,
  ROUND((Sum(CAST([SLastTime]-[SStartTime] AS Float))* 24 - ROUND(Sum(CAST([SLastTime]-[SStartTime] AS Float))* 24, 0, 1))*60, 0, 1) AS SMinutes,
  (Sum(CAST([SLastTime]-[SStartTime] AS Float))*24*60 - ROUND(Sum(CAST([SLastTime]-[SStartTime] AS Float))*24*60, 0, 1))*60 AS SSeconds
  FROM Sessions GROUP BY Year([SStartTime]), Month([SStartTime]) ORDER BY Year([SStartTime]) DESC , Month([SStartTime]) DESC;

Přístupová práva

- ALTER
- SELECT
- INSERT
- UPDATE
- REFERECES
- INDEX
- DROP
- ALL – všechna uvedená práva

Přidělení přístupových práv

- GRANT práva ON objekt TO uživatel;

  seznam práv
- GRANT INSERT, UPDATE, DELETE ON authors TO Mary, John, Tom;
- GRANT SELECT ON authors TO public;
- GRANT právo(položky) ON objekt TO uživatel;

  seznam položek

Přidělení přístupových práv

- Přidělení všech práv k objektu:
- GRANT ALL ON objekt TO uživatel;
  GRANT ALL PRIVILEGES ON objekt TO uživatel;
- Přidělení práv všem uživatelům:
- GRANT práva ON objekt TO PUBLIC;

Umožnění postoupit práva

- GRANT práva ON objekt TO uživatel WITH GRANT OPTION;
- Zmocněný uživatel není vlastníkem schématu, musí se na objekty odkazovat včetně názvu schématu.

Odebrání práv

- REVOKE práva ON objekt FROM uživatel;
- REVOKE SELECT ON Budget_Data FROM Mary;
- REVORE ALL ON objekt FROM uživatel;
  - Odebrání všech práv.

Odebrání práv

- Současné odebrání předaných práv:
  - REVOKE GRANT OPTION FOR práva ON objekt FROM uživatel;
- Současné zrušení objektů vytvořených díky poskytnutým právům:
  - REVOKE práva ON objekt FROM uživatel CASCADE;
- Odebrání práv jen pokud nebyly díky nim vytvořeny jiné objekty (jinak chyba):
  - REVOKE práva ON objekt FROM uživatel RESTRICT;

Řízení transakcí

- Zahájením SQL relace začíná transakce
- COMMIT WORK
  - Ukončení transakce s uložením všech změn
- ROLLBACK WORK
  - Ukončení transakce se stornováním všech změn

Řízení transakcí

- BEGIN TRANSACTION name
  - Zahájení transakce
- COMMIT TRANSACTION name
  - Ukončení transakce
- ROLLBACK TRANSATION name
  - Stornování transakce
- @@TRANCOUNT
  - Systémová proměnná - počet aktivních transakcí

Příklad

```
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
GO
BEGIN TRANSACTION OuterTran -- @@TRANCOUNT set to 1.
GO
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1 -- @@TRANCOUNT set to 2.
GO
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2 -- @@TRANCOUNT set to 3.
GO
INSERT INTO TestTran VALUES (3, 'ccc')
GO
COMMIT TRANSACTION Inner2 -- Decrements @@TRANCOUNT to 2.
-- Nothing committed.
GO
COMMIT TRANSACTION Inner1 -- Decrements @@TRANCOUNT to 1.
-- Nothing committed.
GO
COMMIT TRANSACTION OuterTran -- Decrements @@TRANCOUNT to 0.
-- Commits outer transaction OuterTran.
GO
```

Složitější situace

- SAVE TRANSACTION name
  - Definice bodu návratu – je možno stornovat transakci jen k bodu návratu nebo celou

Problémy souběhu transakcí

- Ztracené aktualizace (lost or buried updates.
- Nečisté čtení (dirty reads).
- Neopakovatelné čtení (nonrepeatable reads).
- Vložení přeludu (phantom reads).

Ztracené aktualizace

Transakce A	Stav databáze	Transakce B
SELECT comm FROM data;	comm = 0.08	SELECT comm FROM data;
UPDATE data SET comm = 0.10 WHERE comm = 0.08;		
COMMIT WORK;	comm = 0.10	
		UPDATE data SET comm = 0.12;
		COMMIT WORK;
	comm = 0.12	

Aktualizace Transakce A byla ztracena,
resp. Transakce B provedla aktualizaci na základě dat před provedením Transakce A.

Nečisté čtení

Transakce A	Stav databáze	Transakce B
SELECT comm FROM data;	comm = 0.08	
UPDATE data SET comm = 0.10 WHERE comm = 0.08;		
	comm = 0.10	
		SELECT comm FROM data;
ROLLBACK WORK;	comm = 0.08	

Transakce B získá data, která jsou následně odstraněna,
což je v rozporu s předpokladem zrušení všech změn Transakce A

Neopakovatelné čtení

Transakce A	Stav databáze	Transakce B
SELECT comm FROM data;	comm = 0.08	SELECT comm FROM data;
UPDATE data SET comm = 0.10 WHERE comm = 0.08;		
COMMIT WORK;	comm = 0.10	
		SELECT comm FROM data;

Transakce B získá při dvou čteních různé hodnoty,
u většiny transakcí se očekává neměnnost dat v průběhu jejich provádění

Vložení předludu

Transakce A	Stav databáze	Transakce B
	AVG(A) = 0.20	SELECT AVG(A) FROM data;
INSERT INTO data VALUES (0.50);		
COMMIT WORK;		
	AVG(A) = 0.30	SELECT AVG(A) FROM data;

Vložením řádku v průběhu Transakce A změní výsledek stejného dotazu

Úrovně izolace

- SET TRANSACTION ISOLATION LEVEL
  { READ COMMITTED
  | READ UNCOMMITTED
  | REPEATABLE READ
  | SERIALIZABLE
  }

Úrovně izolace

	Ztracená aktualizace	Nečisté čtení	Neopakova-telné čtení	Vložení předludu
READ COMMITTED	NE	NE	ANO	ANO
READ UNCOMMITTED	NE	ANO	ANO	ANO
REPEATABLE READ	NE	NE	NE	ANO
SERIALIZABLE (implicitní)	NE	NE	NE	NE