


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

7

Informační systémy

2008/2009


Radim Farana



1

Obsah

- Jazyk SQL,
 - Tvorba pohledů, parametrizace, realizace
- MS SQL Server 2005 (2000)
- Doporučená literatura:
 - Gruber, M. *Mistrovství v SQL*. Svazek 1. Praha : SoftPress s.r.o., 2004. ISBN 80-86497-62-3




Informační systémy

2

Pohledy

- Vytváření virtuálních tabulek.
- Možnost pohledu spojená data z více bázových tabulek.
- Možnost lepší správy přístupových práv.



Informační systémy

3

Vytvoření pohledu

- CREATE VIEW viewname
AS sql_dotaz;
- CREATE VIEW viewname
AS
SELECT title, au_ord, au_lname, price,
ytd_sales, pub_id
FROM authors AS a
JOIN titleauthor AS ta ON (a.au_id =
ta.au_id)
JOIN titles AS t ON (t.title_id = ta.title_id);



Práce s pohledem

- SELECT * FROM viewname WHERE a=1;
 Zobrazení dat
- UPDATE viewname SET a=1 WHERE b=1;
 Modifikace dat



Realizace složitých požadavků

- CREATE VIEW pocet_prace AS
SELECT p_datum, SUM(p_kusu) AS poc,
COUNT(*) AS ks FROM prace;
- SELECT * FROM pocet_prace
WHERE poc > 10 AND ks < 5;



Poddotazy, vícenásobné dotazy

- Pohled může realizovat jakýkoliv SQL dotaz typu SELECT.
- Poddotazy (vnořené dotazy jsou povoleny).
- Vícenásobné dotazy jsou povoleny (např. UNION).



Nepovolené operace

- Pohledy specifikují data, ne jejich uspořádání.
- Klauzule ORDER BY není přípustná.



Aktualizace dat přes pohled

- Pohled nesmí obsahovat
 - Data z více tabulek,
 - Agregáční funkce,
 - Klauzuli DISTINCT,
 - Klauzule HAVING a GROUP BY,
 - Výstupní sloupce nesmí obsahovat výrazy.



Problémy vkládání přes pohledy

- Vložení dat nezobrazitelných pohledem
- CREATE VIEW pohled AS
SELECT p_typ, p_kusu FROM prace
WHERE p_kusu>20;
- INSERT INTO pohled
VALUES ('svícení', 3);
 - Pohled je aktualizovatelný,
 - vložený záznam nebude v pohledu zobrazen.



Odstranění problému vkládání

- CREATE VIEW pohled AS
SELECT p_typ, p_kusu
WHERE p_kusu>20
WITH CHECK OPTION ;
- Pozor na vyloučení klíčových sloupců z
pohledu a nemožnost dodržení požadavků
referenční integrity!



Vkládaný SQL

- Komunikace s hostitelským jazykem přes
proměnné
- Definice proměnné podle požadavků
hostitelského jazyka
 - @name – Transact SQL
 - :name – Standard SQL



Uložení hodnot proměnných

- EXEC SQL
INSERT INTO data
VALUES (:my_ID, :my_Name, ...);



Převzetí hodnot do proměnných

- EXEC SQL
SELECT ID, name, ...
INSERT INTO :my_ID, :my_Name, ...
FROM data
WHERE ID=1;
- Dotaz nesmí vracet více než jeden záznam, jinak není proveden.



Kurzor

- Nástroj pro práci s více záznamy zdroje.
- Deklarace kurzoru bez jeho naplnění:
- EXEC SQL DECLARE CURSOR curname
FOR
SELECT ID, name, ...
FROM data;
- Je vhodné definovat výstupní položky výčtem, aby byl jednoznačný jejich počet.



Práce s kurzorem

- Vytvoření kurzoru a naplnění daty
- EXEC SQL OPEN CURSOR curname;
- Převzetí obsahu jednoho záznamu
- EXEC SQL FETCH curname INTO (:my_ID, :my_name, ...);
- Uzavření kurzoru (bez ohledu na využití)
- EXEC SQL CLOSE CURSOR curname;



Využití SQLSTATE a SQLCODE

- Zjištění výsledku SQL dotazu
- EXEC SQL OPEN CURSOR curname;
- EXEC SQL FETCH curname INTO (:my_ID, :my_name, ...);
- while SQLSTATE='00000'
 ...
 EXEC SQL FETCH curname INTO (:my_ID, :my_name, ...);
- loop
 EXEC SQL CLOSE CURSOR curname;



Hlášení SQLSTATE a SQLCODE

- Výsledky
- SQLSTATE SQLCODE
- xxxxx – -n – chyba (záporné číslo)
- 00xxx – 0 – úspěšné ukončení
- 01xxx – 1 – warning
- 02xxx – 100 – provedeno, žádná data
- SQLSTATE má lépe definovaný výstup, první dva znaky třída, další tři podtřída



GOTO a WHENEVER

- Možnost při zjištění chyby přenést řízení:
- If `SQLSTATE>'20000'` GOTO `err_hand`
- Globální deklarace
- EXEC SQL WHENEVER SQLERROR GOTO `err_hand;` (CONTINUE)
 - SQLERROR – chyba – `SQLCODE<0, SQLSTATE >='30000'`
 - NOT FOUND – žádná data – `SQLCODE=100, SQLSTATE` třída '02' nebo '03'
 - SQLWARNING – upozornění – `SQLCODE=1, SQLSTATE` třída '02'



Aktualizace kurzorů

- Změna dat – UPDATE
- EXEC SQL OPEN CURSOR `curname;`
- EXEC SQL FETCH `curname` INTO `(:my_ID, :my_name, ...);`
- EXEC SQL UPDATE data SET `adata = adata - 1` WHERE CURRENT OF `curname;`
- ...



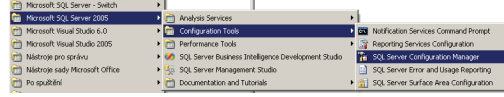
Aktualizace kurzorů

- Odstranění dat – DELETE
- EXEC SQL OPEN CURSOR `curname;`
- EXEC SQL FETCH `curname` INTO `(:my_ID, :my_name, ...);`
- EXEC SQL DELETE data WHERE CURRENT OF `curname;`
- Změna jen na aktuální pozici kurzoru.

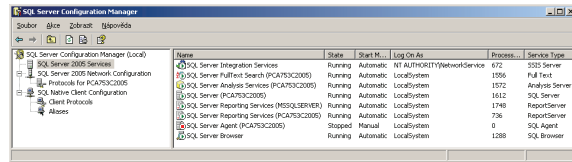


MS SQL Server – nástroje

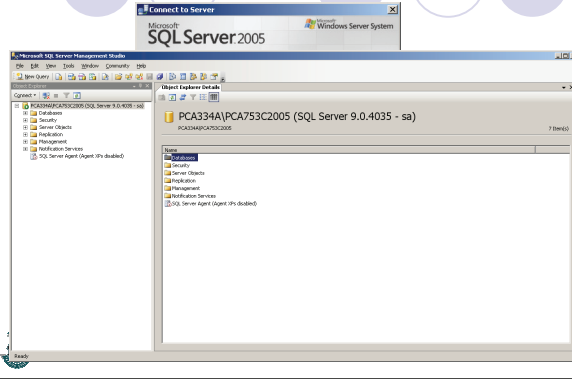
Dvě verze SQL serveru na počítači (2000)



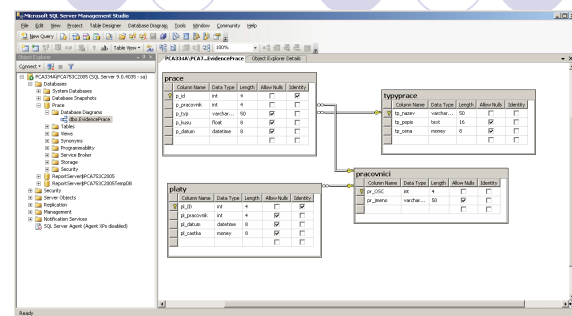
SQL Server Configuration Manager



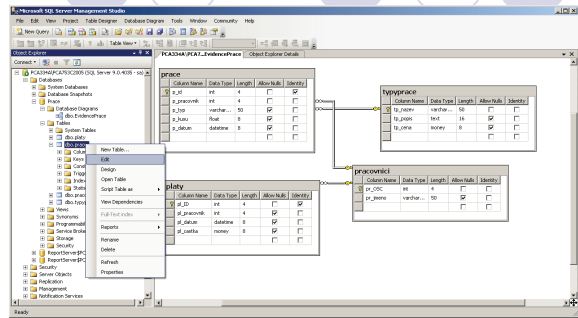
Management Studio



Management Studio – diagramy



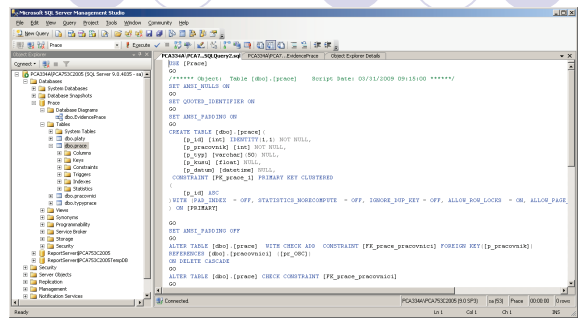
Management Studio – tabulky



Informační systémy

25

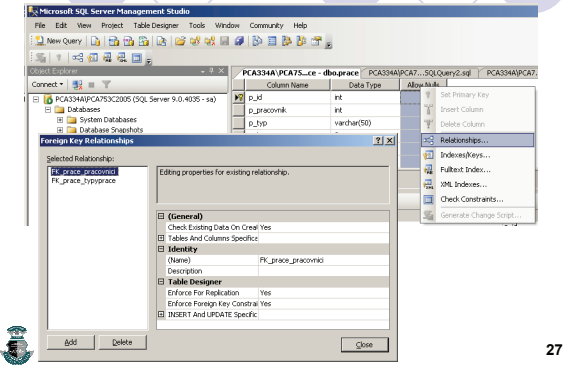
Management Studio – tabulky – Edit



Informační systémy

26

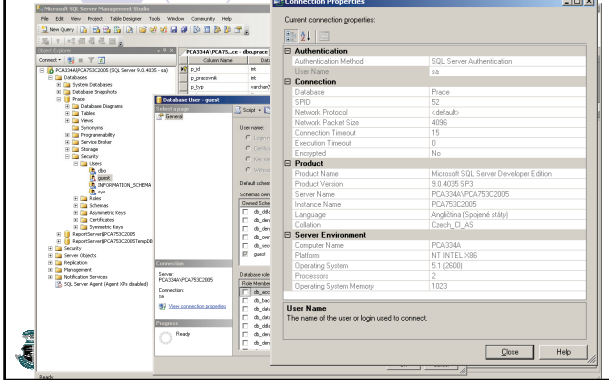
Management Studio – tabulky – Design



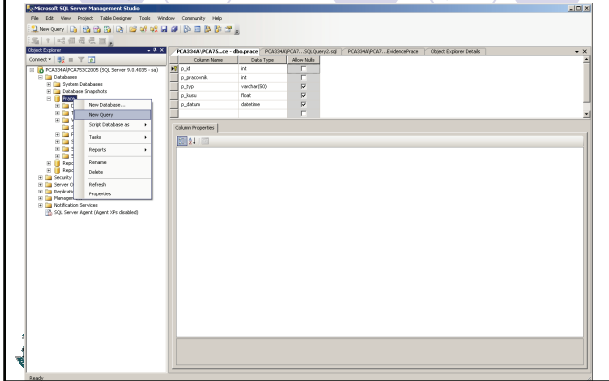
Informační systémy

27

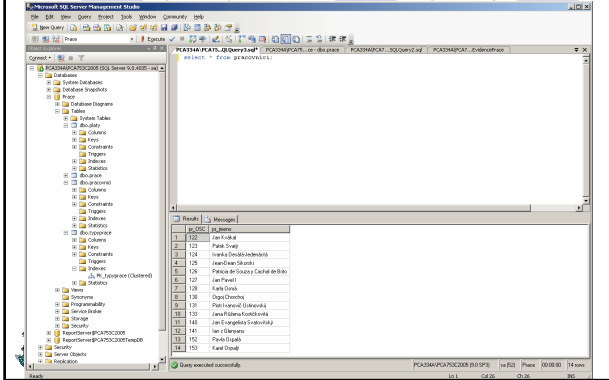
Management Studio – Security



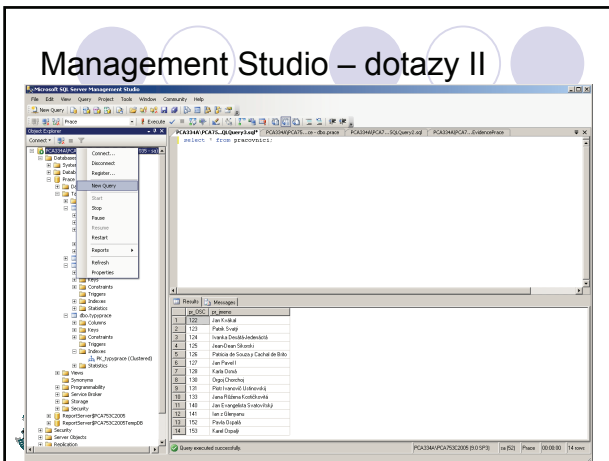
Management Studio – dotazy



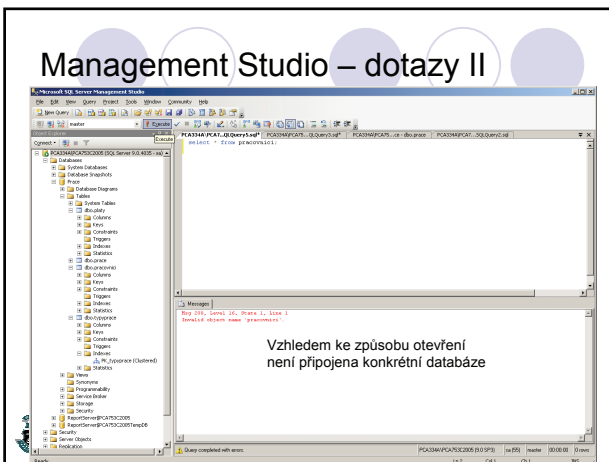
Management Studio – dotazy



Management Studio – dotazy II



Management Studio – dotazy II



Management Studio – dotazy II

