

Proseduurit, funktiot ja herättimet

- esimerkkeinä Oracle, SQL Server, MySQL ja OCELOT

Jouni Huotari

S2008



JYVÄSKYLÄN
AMMATTIKORKEAKOULU

Osaaminen kilpailukyvyksi

Proseduurit

- Ohjelmamoduuleita, jotka voidaan tallettaa tietokantaan (DBMS:n tietohakemistoon)
- Bisneslogiikka lähemmäksi käytettävää tietokantaa
 - Tehokkuus: dataintensiivinen ohjelmalogiikka tietokantapalvelimen muistissa
 - Monikäyttöisyys: samaa ohjelmamoduulia voidaan kutsua monesta eri sovelluksesta (esim. päätelaitteesta) käsin
- Proseduraalisten kielet ovat toimittajakohtaisia:
 - Oraclen PL/SQL (PL = Procedural Language)
 - SQL Serverin Transact SQL
 - ANSI-standardi proseduureille: SQL/PLM (SQL'99)

Perusasioita proseduureista

- Muuttujat määritetään DECLARE-osassa
- Varsinainen ohjelmalogiikka on BEGIN-END-lohkon sisällä
- Tuotekohtaisia eroja ehto- ja toistorakenteiden toteutuksessa, yleensä löytyy kuitenkin IF-THEN-ELSE ja WHILE-luuppi
- SELECTillä haetun tietojoukon käsittely tapahtuu kursorin avulla

OCELOT, proseduurit

- Proseduuri luodaan käskyllä **CREATE PROCEDURE**
 - esim. proseduuri, joka lisää tauluun taulu1 parametrina annettavan kokonaisluvun:

```
CREATE PROCEDURE p1 (x int) MODIFIES SQL DATA  
INSERT INTO taulu1 VALUES (x)
```

- Proseduuria kutsutaan (eli se suoritetaan) **CALL**-käskyllä, esim. lisätään taulu1:een arvo 10:

```
CALL p1(10)
```

Proseduurin esimerkkirunko (OCELOT)

```
CREATE PROCEDURE OmaProseduuri (parametri tietotyyppi)
  MODIFIES SQL DATA
BEGIN
  DECLARE omaMuuttuja1 tietotyyppi;
  DECLARE omaMuuttuja2 tietotyyppi;
  DECLARE omaKursori CURSOR FOR
    SELECT sarake1, sarake2 FROM taulu1 WHERE sarake 3 = parametri;
  OPEN omaKursori;
  FETCH FROM omaKursori INTO omaMuuttuja1, omaMuuttuja2;
  INSERT INTO taulu2 VALUES (omaMuuttuja1, omaMuuttuja2);
  CLOSE omaKursori;
END;
```

Esimerkki Oraclen proseduurista

-- Tulostaa näytölle kaikki työntekijät (EMP-taulusta)

-- määritellään ensin kursori

DECLARE

 cursor employee_cursor is select * from emp;

-- sitten varsinainen ohjelmalohko, joka käy tietojoukon läpi rivi kerrallaan

BEGIN

 for EmpRec in employee_cursor loop

 DBMS_OUTPUT.put_line(EmpRec.empno || ' ' || EmpRec.ename);

 end loop;

END;

/

Funktiot ja herättimet (triggers)

- Funktiot ja herättimet ovat proseduurien erikoistapauksia
- Funktio on proseduuri, joka palauttaa jonkin arvon
- Herätin käynnistyy automaattisesti DML-tietokantaoperaatioista (INSERT, UPDATE tai DELETE)
 - Eheyden valvontaan ja monimutkaisiin tarkistuksiin
 - Muuttuneiden, poistuneiden ja uusien tietojen tallettamiseen toiseen tauluun => loki
 - Tietojen toisteisuuden hallintaan (eheystarkistukset, joita ei ole tehty viite-eheyssäännöillä)

OCELOT, funktiot

- Funktio luodaan käskyllä **CREATE FUNCTION**
 - Esim. funktio, joka palauttaa merkkijonon 'Suuri' tai 'Pieni':
CREATE FUNCTION f1 (p1 INT) RETURNS CHAR(5)
BEGIN
 IF p1 > 15 THEN
 RETURN 'Suuri ';
 ELSE
 RETURN 'Pieni';
 END IF;
END;
- Funktiota käytetään osana SELECTiä, esim.:
SELECT f1(17) FROM t1;

OCELOT, herättimet

- Herätin luodaan käskyllä **CREATE TRIGGER**
 - Esim. herätin, joka tarkistaa tiedon ennen tallennusta tauluun koe
- ```
CREATE TRIGGER trig1
BEFORE UPDATE ON koe
BEGIN
 IF p1 > 15 THEN
 ROLLBACK;
 ELSE
 COMMIT;
 END IF;
END;
```
- Herätintä ei siis tarvitse erikseen kutsua

# SQL Server, esimerkki

SQL Query Analyzer - [Query - INSTALL00\INSTALL01.Northwind.INSTALL00\Administrator - Untitled5\*]

Object Browser

- INSTALL00\INSTALL01(INSTALL00\Administratc
  - master
  - model
  - msdb
  - Northwind
    - User Tables
      - dbo.Categories
        - Columns
        - Indexes
        - Constraints
        - Dependencies
        - Triggers
      - dbo.CustomerCustomerDemo
      - dbo.CustomerDemographics
      - dbo.Customers
      - dbo.Employees
      - dbo.EmployeeTerritories
      - dbo.Order Details
      - dbo.Orders
      - dbo.Products
      - dbo.Region
      - dbo.Shippers
      - dbo.Suppliers
      - dbo.Territories
    - System Tables
    - Views
    - Stored Procedures
      - dbo.CustOrderHist
      - dbo.CustOrdersDetail
      - dbo.CustOrdersOrders
      - dbo.Employee Sales by Country
      - dbo.Sales by Year
      - dbo.SalesByCategory
      - dbo.Ten Most Expensive Products
    - Functions
    - User Defined Data Types
  - pubs

```
DECLARE @RC int
DECLARE @CustomerID nchar(5)
-- Set parameter values
EXECUTE = [Northwind].[dbo].[CustOrderHist] 'ALFKI'
```

1. Execute in new window

2. Muuta parametri(t) tarvittaessa

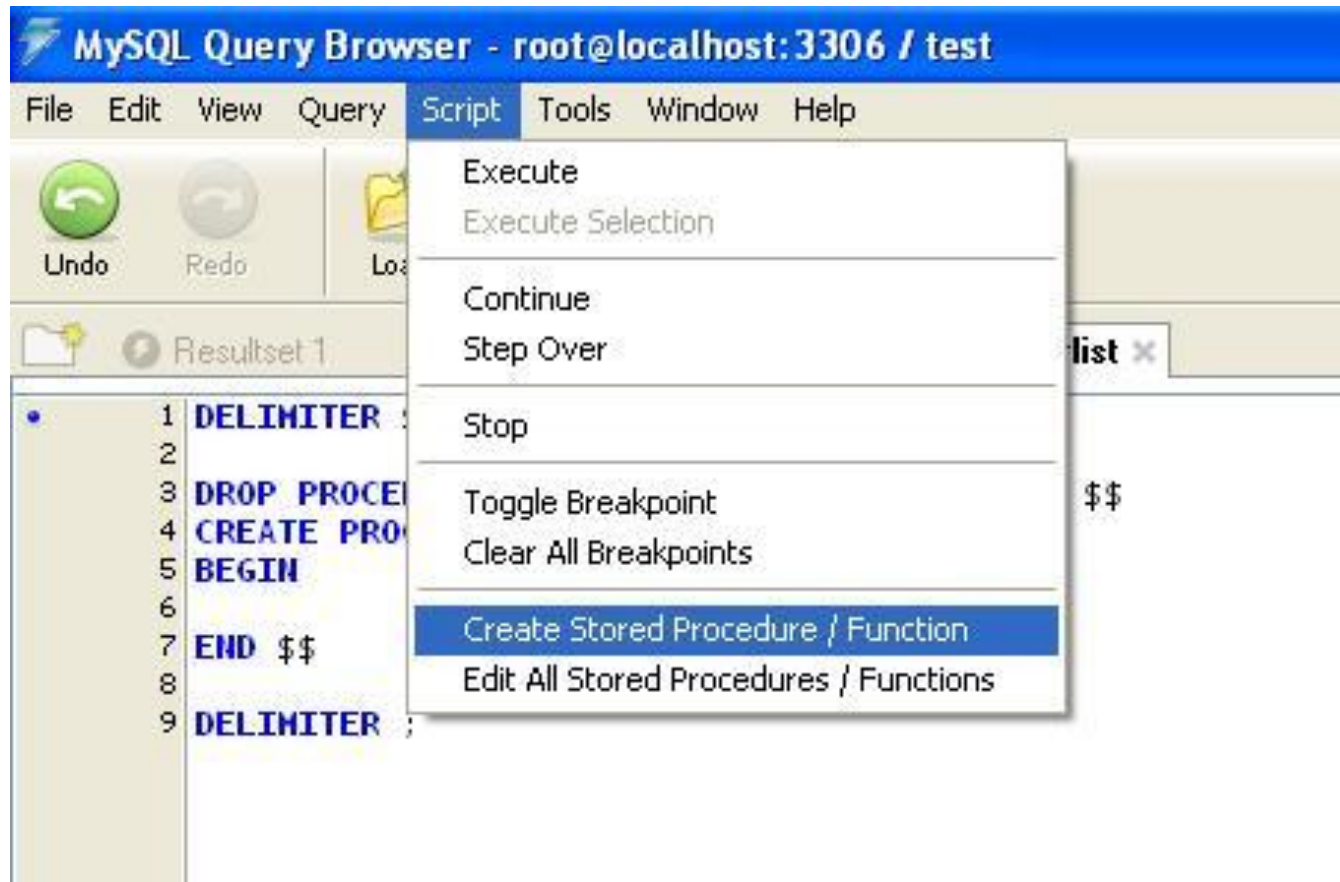
3. Paina Start-painiketta

|    | ProductName                       | Total |
|----|-----------------------------------|-------|
| 1  | Aniseed Syrup                     | 6     |
| 2  | Chartreuse verte                  | 21    |
| 3  | Escargots de Bourgogne            | 40    |
| 4  | Flotemysost                       | 20    |
| 5  | Grandma's Boysenberry Spread      | 16    |
| 6  | Lakkalikööri                      | 15    |
| 7  | Original Frankfurter grüne Soße 2 | 15    |
| 8  | Raclette Courdavault              | 15    |
| 9  | Rössle Sauerkraut                 | 17    |
| 10 | Spegesild                         | 2     |
| 11 | Veggie-spread                     | 20    |

Query batch completed. INSTALL00\INSTALL01 (8.0) INSTALL00\Administrator (61) Northwind 0:00:00 11 rows Ln 4, Col 53

Connections: 9

# *MySQL, esimerkki Query Browserin käytöstä proseduurin luonnissa*



- Herong's Tutorial Notes: [http://www.herongyang.com/sql/proc\\_sql.html](http://www.herongyang.com/sql/proc_sql.html)
- <http://dev.mysql.com/doc/refman/5.1-maria/en/create-procedure.html>

# Oracle

- Käynnistä SQL\*Plus (Worksheet)
- Luo proseduuuri (BEGIN ... END)
- Laita päälle tulostus näytölle:  
set serveroutput on
- Suorita proseduuuri esim. samalla idealla kuin teit proseduurin, ts. kutsu sitä:
  - begin ProseduurinNimi(parametrit); end;
  - tai jos olet antanut proseduurille nimen, voit kutsua sitä:  
start ProseduurinNimi
- PL/SQL-harjoitukset: erillinen tiedosto