Server-side programming in modern DBMS

Maciej Grzenda
Warsaw University of Technology
Faculty of Mathematics and Information Science
M.Grzenda@mini.pw.edu.pl
http://www.mini.pw.edu.pl/~grzemand
DBMS programming

Server-side procedures

Stored procedures
- Standard SP
- System SP
- Triggers
- Extended SP

Functions
- Parametrised views
- Returning scalar values

Ms SQL Server capabilities only. Diverse capabilities can be found in different DBMS.
Stored procedures

- The procedures defined in SQL-like language, executed by DBMS
- Allows to speed up the execution of resource-intensive processing e.g. complex updates affecting thousands of documents
- System SP provide means to configure DBMS and set some of the options not available through SQL statements
Stored procedure – basic syntax

- CREATE PROCEDURE ProcedureName [@parameterName ParameterType [OUTPUT],...]
  AS
  procedureCode

- EXECUTE PROCEDURE ProcedureName [@variableName|Value,...]

• The syntax comes from Ms SQL Server
• Multiple arguments of different types can be specified
• The procedure can return multiple values through OUTPUT parameters
Sample stored procedure

CREATE PROCEDURE CalculateOrderCount @country varchar(100) 
AS
DECLARE @customer varchar(100)
DECLARE custom CURSOR LOCAL FOR SELECT CustomerId FROM Customers 
    WHERE country=@country
OPEN custom
FETCH NEXT FROM custom INTO @customer
WHILE @@FETCH_STATUS=0
BEGIN
    UPDATE Customers SET OrderCount=(SELECT COUNT(*) FROM Orders 
        WHERE CustomerId=@customer) WHERE CustomerId=@customer
    FETCH NEXT FROM custom INTO @customer
END
CLOSE custom
DEALLOCATE custom

The procedure affects multiple records
By using stored procedures we can significantly reduce network communication overhead
Database logic – pros and cons

- Database functions and stored procedures:
  - allow to create parameterised views and complex server-side processing
  - The most efficient processing – reduced client-server communication overhead
  - Execution plans created by the database server => faster than ad hoc statements
  - Unfortunately different standard exist (Transact-SQL, PL/SQL) => limited portability of procedure/function code
Issues to consider

• Processing efficiency – can we postpone the processing?

• Portability of a solution:
  – Stored procedures,
  – User-defined functions,
  – Enterprise Java Beans (EJB)
  – …

• Estimate the cost of each solution satisfying „must do” requirements
Triggers

- Procedures executed after modification or instead of modification to the database
- There are INSERT triggers, UPDATE triggers and DELETE triggers
- Thus you can define for instance AFTER INSERT trigger for ORDERS table i.e. the procedure that will be called by the DBMS every time INSERT statement has been executed
- Notice: if multiple records are affected by a single CRUD statement, the trigger is activated only once – for the whole group of records
Triggers - remarks

- Capabilities may depend on SQL implementation.
- In some implementations (Transact-SQL) triggers can be defined for views. Thus a view behaves like a table and allows to define what actions will take place when a record is inserted into a view.
- Trigger can change the content of another or even the same table and thus make DBMS fire other triggers.
CREATE TRIGGER TriggerName ON TableName
{AFTER|INSTEAD OF}
{INSERT,UPDATE,DELETE}
AS
TriggerCode

AFTER triggers will be called after an action e.g. after a record is inserted into a table.

INSTEAD OF triggers in according to their name are called instead of original actions e.g. instead of inserting the records into a table/view they are defined for.
Trigger - example

CREATE TRIGGER InsertOrder ON Orders AFTER INSERT 
AS
DECLARE @customer varchar(100)
SET @customer=(SELECT CustomerId FROM inserted)
UPDATE Customers SET OrderCount=OrderCount+1
WHERE CustomerId=@customer

1. This code assumes that one order is inserted by an INSERT statement.
2. Recall that multiple records can be inserted at the same time by using
   INSERT ..... SELECT ..... 
3. Sample code comes from Ms SQL Server
4. The trigger will be executed AFTER processing each INSERT 
   requested on Orders table
Programming constraints

- Guidelines:
  - First try to use referential integrity, define primary keys and check constraints,
  - if necessary use triggers to enforce more complex rules,
  - Triggers can roll back the whole transaction. Still it is not suggested to roll back transaction, especially in AFTER triggers.
SP, triggers and functions

- Enable the most efficient form of data processing – thus are useful when time really matters,
- Can be used by multiple applications and ensure consistent processing
- Triggers can help perform the same actions no matter if the CRUD statement was issued from one application or another
Execution plans

• While the same or similar code as in SP could be used in an ad hoc statement, SP can take advantage of execution plan

• DBMS may maintain execution plans for SP that further decrease execution time by:
  – Caching the best predicted method of accessing data and performing requested operations for an entire SP compiled and analysed beforehand.
Regular execution

• Modern DBMS run continuously
• They offer built-in capability of executing different actions on regular basis e.g. every hour
• Thus, stored procedures can be also executed automatically – for instance every night to transfer temporary data between the systems, remove out-of-date information etc.
• In Ms SQL Server, SQL Server Agent is the service responsible for managing jobs