ESOF 326, Software Maintenance

Stored Procedures

Stored Procedures

Stored Procedure – subroutine available to applications that access a relational database management system (RDMS), can take inputs and the results are always tables

3 types of DB stored programs:

1. *Stored procedures*
2. Stored functions – like a stored procedure but can return a single value, can be used within a stored procedure
3. Triggers - stored programs that are activated in response to an activity within the database, typically, in response to an INSERT, UPDATE, DELETE

Advantages of Stored Procedures

Advantages of stored procedures include:

- Separation of logic
- Improved productivity because statements in a stored procedure are only written once
- Permissions – can just grant permission to execute the procedure, not on the tables
- Optimization – it is easier for a DBA to optimize the SQL and tune the database when stored procedures are used (it is easier to find missing indexes and such)
- Reduce network traffic (see next)
- Caching query plan – the first time the stored procedure is executed, an execution plan is created , which is cached for reuse (see next)

(Stackoverflow, from a response to discussion on next slide)

Advantages

Historical performance benefit:

- Reduced network latency
- Pre-parsed SQL
- Pre-generated query execution plan

"Are Stored Procedures more efficient, in general, than inline statements on modern RDBMS's" Stackoverflow https://stackoverflow.com/questions/59880/are-stored-procedures-more-efficient-in-general-than-inline-statements-on-mode (2008)

Reduced network latency

Sending a simple “exec foobar” can be much quicker than sending an large query. Also more can be done in one stored procedure, so fewer calls to DBMS.

Still an advantage?

Modern Ethernets are fast but why shove megabytes of data back and forth for no good reason?

Pre-parsed SQL

Similar benefits to compiled vs. interpreted code

Still an advantage?

Not very noticeable on the modern CPU

Pre-generated query execution plan

Permutations of many JOINs can be large and the optimizer needs to figure out the "near best" execution plan. Stored procedures store plan in memory so the overhead is avoided.

Still an advantage?

Modern DBMS's cache query plans, reducing the benefit.

However, some cached query plans are sub-optimal.

Example Stored Procedure

```
DELIMITER //

DROP PROCEDURE IF EXISTS list_semesters_in_interval;

/**
 * list_semesters semesters takes the id of a semester interval  and returns the names of those
* semesters within that interval  inclusive. This procedure assumes that the end semester of
 * the interval is not null.
 *    inputs: id of a semester interval (end semester cannot be null)
 *    outputs: list of names of all semesters within that semester
 *             interval, inclusive
*/
CREATE PROCEDURE list_semesters_in_interval(IN interval _id INT)
BEGIN
        DECLARE begin_id INT;
        DECLARE end_id INT;

        SELECT begin, end INTO begin_id, end_id
        FROM intervals
        WHERE id = interval _id;

        SELECT name
        FROM semesters
        WHERE id between begin_id and end_id;

END //
```

Example Stored Procedure Call

```
CALL list_semesters_in_interval(1);
```