

PL/SQL og kveikjur

Gagnasafnsfræði 2011

Hallgrímur H. Gunnarsson

Háskóli Íslands

2011-11-17

Uppbygging stefs

```
CREATE FUNCTION identifier (arguments) RETURNS type AS '  
  DECLARE  
    declaration;  
    [...]  
  BEGIN  
    statement;  
    [...]  
  END;  
' LANGUAGE 'plpgsql';
```

Dæmi um vsk stef

```
CREATE FUNCTION mvsk (x real) RETURNS REAL AS '  
    BEGIN  
        RETURN x*0.255;  
    END;  
' LANGUAGE 'plpgsql';
```

Dæmi um factorial stef

```
CREATE FUNCTION FACT(N INTEGER)
    RETURNS INTEGER AS $FACT$
DECLARE
    S INTEGER := 1; //Ákvörðum breytur hér
BEGIN
    FOR i IN 2..N LOOP
        S := S*i;
    END LOOP;
    RETURN S;
END;
$FACT$
LANGUAGE PLPGSQL;
```

Stef sem endurnýjar reiknaðar sýndartöflur

```
CREATE FUNCTION cs_refresh_mviews () RETURNS INTEGER AS '  
DECLARE  
    mviews RECORD;  
BEGIN  
    RAISE NOTICE 'Refreshing materialized views...';  
  
    FOR mviews IN SELECT * FROM cs_materialized_views LOOP  
        -- Now "mviews" has one record from cs_materialized_views  
        RAISE NOTICE 'Refreshing materialized view %', quote_ident(mviews.mv_name);  
        EXECUTE 'TRUNCATE TABLE ' || quote_ident(mviews.mv_name);  
        EXECUTE 'INSERT INTO ' || quote_ident(mviews.mv_name) || ' ' || mviews.mv_query;  
    END LOOP;  
  
    RAISE NOTICE 'Done refreshing materialized views.'  
    RETURN 1;  
end;  
' LANGUAGE 'plpgsql';
```

Kveikjur

Kveikja er stef sem keyrir sjálfkrafa þegar tiltekinn *atburður* á sér stað.

Þrír þættir:

- Atburður (e. event) sem ræsir kveikju: insert, delete, update
- Skilyrði (e. condition) sem ákvarðar hvort kveikjan fari af stað
- Aðgerð (e. action) sem segir til um hvað á að gera þegar kveikjan er framkvæmd

Dæmi um notkun:

- Skorða leyfileg gögn í insert, t.d. kennitölutékk
- Skrá breytingar (e. logging)
- Business logík, t.d. bregðast við ef birgðastaða á lager fer undir eitthvað ákveðið lágmark

Kveikjur í SQL

SQL til að stofna kveikju:

```
CREATE TRIGGER name { BEFORE | AFTER } { event [ OR ... ] }  
    ON table [ FOR [ EACH ] { ROW | STATEMENT } ]  
    EXECUTE PROCEDURE funcname ( arguments )
```

Parameters:

- event: INSERT, UPDATE eða DELETE.
- BEFORE og AFTER: Ákvarðar hvort stefið er framkvæmt fyrir eða eftir atburðinn
- FOR EACH ROW/STATEMENT: Tilgreinir hvort það eigi að framkvæma stefið einu sinni eða fyrir hverja röð sem breyttist við framkvæmdina
- funcname: Stef sem á að framkvæma

PL/SQL og kveikjur

Stef fyrir kveikju í PL/SQL hefur aðgang að mörgum sérstökum breytum:

- OLD: Gamla röðin (RECORD tag)
- NEW: Nýja röðin (RECORD tag)
- TG_NAME: Nafn á kveikju
- TG_WHEN: Hvort kveikjan var ræst BEFORE eða AFTER
- TG_LEVEL: FOR EACH ROW eða STATEMENT kveikja
- TG_OP: Aðgerð (INSERT, DELETE eða UPDATE)
- TG_NARGS: Fjöldi viðfanga í kallið úr kveikjunni
- TG_ARGV: Fylki með viðföngum

Dæmi um trigger

```
CREATE TRIGGER audit_log AFTER UPDATE
    ON employees FOR each ROW
    EXECUTE PROCEDURE audit_logger();
```

Dæmi um trigger stef

```
CREATE FUNCTION audit_logger() RETURNS trigger AS '
BEGIN
    INSERT INTO employees_salary_changes
        (empid, old_salary, new_salary)
        VALUES
            (new.empid, old.salary, new.salary);

    RETURN new;
END
' LANGUAGE plpgsql;
```