Praktikum „Integritätsbedingungen“, Phase 3

Henning Basold Igor Zerr

27. Juni 2011

1 Generierte Funktionen und Trigger

1.1 HOUSE_HOUSE_DISJOINT

```
CREATE ASSERTION HOUSE_HOUSE_DISJOINT CHECK (
    NOT EXISTS ( SELECT * FROM
        Haus AS h1, Haus AS h2
        WHERE h1.id <> h2.id and h1.umriss && h2.umriss
    )
);

CREATE FUNCTION check_house_house_disjoint() RETURNS trigger
    LANGUAGE plpgsql
    AS $$Declare res RECORD;
BEGIN
  SELECT INTO res COUNT(*) AS num
  FROM TestSysRel
  WHERE NOT (
    NOT EXISTS ( SELECT * FROM
        Haus AS h1, Haus AS h2
        WHERE h1.id <> h2.id and h1.umriss && h2.umriss
    )
  )
;  IF (res.num > 0)
  THEN RAISE EXCEPTION
    'ASSERTION_CHECK_HOUSE_HOUSE_DISJOINT_violated!';
  END IF;
  RETURN NEW;
END;$$;

CREATE TRIGGER check_house_house_disjoint_haus
    AFTER INSERT OR DELETE OR UPDATE ON haus
    FOR EACH ROW
    EXECUTE PROCEDURE check_house_house_disjoint();

CREATE TRIGGER check_house_lake_disjoint_haus
    AFTER INSERT OR DELETE OR UPDATE ON haus
    FOR EACH ROW
    EXECUTE PROCEDURE check_house_lake_disjoint();
```

1.2 NO_STANDALONE_STOP

```
CREATE ASSERTION NO_STANDALONE_STOP CHECK (
     NOT EXISTS ( SELECT * FROM
          Haltestelle AS b
          WHERE NOT EXISTS ( SELECT * FROM
               Strasse AS s
               WHERE approx_circle(b.position, 1) ?# s.verlauf
          )
     )
);

CREATE FUNCTION check_no_standalone_stop() RETURNS trigger
     LANGUAGE plpgsql
     AS $$Declare res RECORD;
BEGIN
  SELECT INTO res COUNT(*) AS num
  FROM TestSysRel
  WHERE NOT (
     NOT EXISTS ( SELECT * FROM
          Haltestelle AS b
          WHERE NOT EXISTS ( SELECT * FROM
               Strasse AS s
               WHERE approx_circle(b.position, 1) ?# s.verlauf
          )
     )
   )
;  IF (res.num > 0)
  THEN RAISE EXCEPTION
     'ASSERTION_CHECK_NO_STANDALONE_STOP_violated!';
  END IF;
  RETURN NEW;
END;$$;

CREATE TRIGGER check_no_standalone_stop_haltestelle
     AFTER INSERT OR DELETE OR UPDATE ON haltestelle
     FOR EACH ROW
     EXECUTE PROCEDURE check_no_standalone_stop();

CREATE TRIGGER check_no_standalone_stop_strasse
     AFTER INSERT OR DELETE OR UPDATE ON strasse
     FOR EACH ROW
     EXECUTE PROCEDURE check_no_standalone_stop();
```

2 Datenimport

2.1 Übersicht

```
Inserted Tuples:
    Tunnel: 48
    Park: 30
    Strassenbahn: 55
    Spielplatz: 24
    Strasse: 1150
    Fluss: 7
    Parkplatz: 22
    Eisenbahn: 55
    See: 5
    Bruecke: 47
    Haus: 2205
    Landnutzung: 70
Total: 3718

Violated Assertions:
    CHECK_STREET_PLAYGROUND_DISJOINT: 1
    CHECK_PARKING_REACHABLE: 43
    CHECK_STREET_LANDUSE_DISJOINT: 74
    CHECK_NO_STANDALONE_STOP: 167
    CHECK_HOUSE_STREET_DISJOINT: 11
    CHECK_TRAFFIC_LIGHT_AT_STREET: 223
    CHECK_HOUSE_HOUSE_DISJOINT: 1604
    CHECK_HOUSE_LAKE_DISJOINT: 0
```

2.2 Nicht verletzte Assertions

2.2.1 HOUSE_STREET_DISJOINT

SQL Verletzung:

```sql
INSERT INTO Haus (id,umriss, hausnummer, strasse, ort, plz, nutzung)
    values('3457842','((52.2658021,10.5088862),(52.2658321,10.5082349))
    ','42','Kasernenstrasse','Braunschweig','38102',NULL)
```

2.2.2 HOUSE_LAKE_DISJOINT

SQL Verletzung:

```sql
INSERT INTO Haus (id,umriss, hausnummer, strasse, ort, plz, nutzung)
    values('3457843','((52.2593691,10.530547),(52.2593475,10.5305934)
    ,(52.2593108,10.530598),(52.2592873,10.5305653)
    ,(52.2592827,10.5305139),(52.2592984,10.5304732)
    ,(52.2593168,10.5304591),(52.2593452,10.5304646)
    ,(52.2593669,10.5305019),(52.2593691,10.530547))','42','
    Kasernenstrasse','Braunschweig','38102',NULL)
```

2.2.3 STREET_LAKE_DISJOINT

SQL Verletzung:

```sql
INSERT INTO Strasse (id,verlauf,strassentyp,name,maxV) values('
    23251939','[(52.2601605,10.5313293),(52.2601389,10.5313757)
    ,(52.2601022,10.5313803),(52.2600787,10.5313476)
    ,(52.2600742,10.5312962),(52.2600898,10.5312555)
    ,(52.2601082,10.5312414),(52.2601366,10.5312469)
    ,(52.2601583,10.5312842),(52.2601605,10.5313293)]','residential','
    Neue␣Knochenhauerstrasse','0')
```

2.2.4 WATER_CROSSING

```sql
INSERT INTO Bruecke (id,verlauf,name,maxV) values('3457844','
    [(52.2674258,10.5246387),(52.2669683,10.5250233)]','Leonhardstrasse
    ','30')
```

3 Bestimmen der betroffenen Relationen

```
Set<String>
getAffectedTables(Connection sql, Assertion a) throws SQLException {
    Pattern tableExtraction
        = Pattern.compile(".*␣Scan.*␣on␣(\\w+)␣.*");

    Set<String> affectedTables
        = new CopyOnWriteArraySet<String>();

    Statement check = sql.createStatement();
    ResultSet pred
        = check.executeQuery(
            "EXPLAIN␣SELECT␣*␣FROM␣TestSysRel␣WHERE␣" + a.predicate);
    while(pred.next()){
        Matcher match
            = tableExtraction.matcher(pred.getString("QUERY␣PLAN"));
        if(match.find()){
            String table = match.group(1);
            if(!table.equalsIgnoreCase("TestSysRel")){
                affectedTables.add(table);
            }
        }
    }

    return affectedTables;
}
```