



Databáze zvířat

Dokumentace temporalních dotazů

2010/2011

Řešitelé: Ondřej Beneš – xbenes00

Tomáš Ižák – xizakt00

Pavel Srnec – xsrnec01

Jazyk T2SQL

Pro zjednodušení zápisu temporálních dotazů v tomto projektu byl vytvořen zjednodušený jazyk obdobný syntaxi TimeDB2 (byl nazván TimeToSQL, zkráceně T2SQL). Vztahuje se na dotazy nad tabulkou s prostorovými daty, která byla vybrána jako temporální. Využití v praxi by bylo sledování výskytu jednotlivých zvířat v určitém časovém období (okamžiku). Vzhledem k použití projektu lze vkládat, upravovat a mazat data nejen k aktuálnímu okamžiku, ale i k zadaným časům.

Použití T2SQL je následující:

- VALIDTIME PERIOD [<Date>,<Date>) <SQL>
pracuje nad daty v zadaném časovém intervalu
- VALIDTIME DATE <Date> <SQL>
pracuje nad daty platnými v zadaném časovém okamžiku
- NONSEQUENCED VALIDTIME <SQL>
pracuje nad všemi daty – platnými i neplatnými
- <SQL>
pracuje pouze nad aktuálně platnými daty

<Date> je informace o datu ve formátu yyyy/MM/dd

<SQL> je obyčejný SQL dotaz nad tabulkou prostorových a temporálních dat

Příklady temporálních dotazů

SELECT

- Nad 1 tabulkou

TSQL2

```
SELECT * FROM animal_movement WHERE animal_id = 1
```

T2SQL

```
SELECT move_id, animal_id, geometry  
FROM animal_movement  
WHERE animal_id = 1
```

SQL

Automaticky přeložený SQL dotaz

```
SELECT move_id, animal_id, geometry  
FROM animal_movement  
WHERE (valid_from <= sysdate OR valid_from is NULL) AND (valid_to > sysdate OR valid_to is  
NULL) AND animal_id = 1
```

- Nad více tabulkami

TSQL2

```
SELECT DISTINCT a.animal_id,a.genus,a.species
FROM animal a, animal_movement am WHERE am.animal_id<>3 AND
a.animal_id=am.animal_id AND
SDO_GEOM.SDO_INTERSECTION(am.geometry,SDO_GEOMETRY(...),1) IS NOT NULL
VALID PERIOD [2010-01-01 – 2010-12-31]
TODO -?
```

T2SQL

```
VALIDTIME PERIOD [2010-01-01,2010-12-31)
SELECT DISTINCT a.animal_id, a.genus, a.species
FROM animal a, animal_movement am
WHERE am.animal_id<>3 AND a.animal_id=am.animal_id AND
SDO_GEOM.SDO_INTERSECTION(am.geometry,SDO_GEOMETRY(...),1) IS NOT NULL
```

SQL

Automaticky přeložený SQL dotaz

```
SELECT DISTINCT a.animal_id, a.genus, a.species
FROM animal a, animal_movement am
WHERE (((valid_from <= DATE '2010-01-01' OR valid_from is NULL) AND (valid_to > DATE
'2010-01-01' OR valid_to is NULL)) OR ((valid_from <= DATE '2010-12-31' OR valid_from is
NULL) AND (valid_to > DATE '2010-12-31' OR valid_to is NULL))) AND am.animal_id<>3
AND a.animal_id=am.animal_id AND SDO_GEOM.SDO_INTERSECTION(am.geometry,
SDO_GEOMETRY(...), 1) IS NOT NULL
```

- Nad více tabulkami (2)

TSQL2

```
SELECT SUM(SDO_GEOM.SDO_AREA(SDO_GEOM.SDO_INTERSECTION(a.geometry,
b.geometry,0.1),0.1,'unit=SQ_KM')) AS area FROM animal_movement a, animal_movement b
WHERE a.animal_id=1 AND b.animal_id=1 AND a.move_id<b.move_id
```

SQL

```
SELECT SUM(SDO_GEOM.SDO_AREA(SDO_GEOM.SDO_INTERSECTION(a.geometry,
b.geometry,0.1),0.1,'unit=SQ_KM')) AS area
FROM animal_movement a, animal_movement b
WHERE a.animal_id=1 AND b.animal_id=1 AND a.move_id<b.move_id AND
(((a.valid_from <= sysdate OR a.valid_from is NULL) AND (a.valid_to > sysdate OR a.valid_to is
NULL)) OR ((a.valid_from <= sysdate OR a.valid_from is NULL) AND (a.valid_to > sysdate OR
a.valid_to is NULL))) AND
(((b.valid_from <= sysdate OR b.valid_from is NULL) AND (b.valid_to > sysdate OR b.valid_to is
NULL)) OR ((b.valid_from <= sysdate OR b.valid_from is NULL) AND (b.valid_to > sysdate OR
b.valid_to is NULL)))
```

UPDATE

TSQL2

```
UPDATE animal_movement  
SET geometry = SDO_GEOMETRY(...) VALID PERIOD [2010-06-06 – 2010-06-08]  
WHERE move_id = 10
```

T2SQL

```
VALIDTIME PERIOD [2010-06-06,2010-06-08) UPDATE animal_movement SET geometry =  
SDO_GEOMETRY(...) WHERE move_id = 10
```

SQL

Je provedena uloženou procedurou animal_movement_update, která v tomto případě pracuje následovně. Napsáno pseudokódem.

```
IF new_from<=old_from AND new_to>old_from THEN
```

```
    IF new_to>=old_to THEN
```

```
        UPDATE animal_movement SET valid_from=new_from, valid_to=new_to,  
geometry=new_move WHERE move_id=old_move_id;
```

```
    ELSE
```

```
        UPDATE animal_movement SET valid_from=new_to WHERE move_id=old_move_id;
```

```
        INSERT INTO animal_movement (animal_id,valid_from, valid_to, geometry)
```

```
        VALUES (old_animal_id, new_from, new_to,new_move);
```

```
    END IF;
```

```
ELSIF new_to>=old_to AND new_from<old_to THEN
```

```
    IF new_from<=old_from THEN
```

```
        UPDATE animal_movement SET valid_from=new_from, valid_to=new_to,  
geometry=new_move WHERE move_id=old_move_id;
```

```
    ELSE UPDATE animal_movement SET valid_to=new_from WHERE move_id=old_move_id;
```

```
        INSERT INTO animal_movement (animal_id,valid_from, valid_to, geometry)
```

```
        VALUES (old_animal_id, new_from, new_to,new_move);
```

```
    END IF;
```

```
ELSE
```

```
    INSERT INTO animal_movement (animal_id,valid_from, valid_to, geometry)
```

```
    VALUES (old_animal_id, new_to, old_to,old_move);
```

```
    UPDATE animal_movement SET valid_to=new_from WHERE move_id=old_move_id;
```

```
    INSERT INTO animal_movement (animal_id,valid_from, valid_to, geometry)
```

```
    VALUES (old_animal_id, new_from, new_to,new_move);
```

```
END IF;
```

DELETE

TSQL2

```
DELETE FROM animal_movement  
WHERE move_id = 10 VALID PERIOD [2010-06-06 - FOREVER]
```

T2SQL

```
VALIDTIME DATE 2010-06-06 DELETE FROM animal_movement WHERE move_id = 10
```

SQL

Je provedena uloženou procedurou animal_movement_delete, která v tomto případě pracuje následovně. Napsáno pseudokódem.

```
IF new_from<=old_from AND new_to>old_from THEN  
    IF new_to>=old_to THEN  
        DELETE FROM animal_movement WHERE move_id=old_move_id;  
    ELSE  
        UPDATE animal_movement SET valid_from=new_to WHERE move_id=old_move_id;  
    END IF;  
ELSIF new_to>=old_to AND new_from<old_to THEN  
    IF new_from<=old_from THEN  
        DELETE FROM animal_movement WHERE move_id=old_move_id;  
    ELSE  
        UPDATE animal_movement SET valid_to=new_from-1 WHERE move_id=old_move_id;  
    END IF;  
ELSE  
    INSERT INTO animal_movement (animal_id,valid_from, valid_to, geometry)  
    VALUES (old_animal_id, new_to, old_to,old_move);  
    UPDATE animal_movement SET valid_to=new_from-1 WHERE move_id=old_move_id;  
END IF;
```