DAT151-OBLIG3

Triggers, Procedures, database optimization

Adrian Mortensen
https://github.com/H571531/DAT151/tree/master/Oblig3

Contents

All kode / output kan også finnes på https://github.com/H571531/DAT151/tree/master/Oblig3

Task 1: Triggers

Code:

```
delimiter $ --Setter en egen delimiter siden ';' Brukes i triggerne.
CREATE TRIGGER Task1Insert AFTER INSERT ON TheTable
    FOR EACH ROW
    BEGIN
        INSERT INTO LogTable (changeTimeStamp,action,tableId,name,note)
        VALUES(NOW(),'INSERT',NEW.id,NEW.name,NEW.note);
    END$

CREATE TRIGGER TASK1UPDATEa AFTER UPDATE ON TheTable
    FOR EACH ROW
    BEGIN
        INSERT INTO LogTable (changeTimeStamp,action,tableId,name,note)
        VALUES(NOW(),'UPDATEb',OLD.id,OLD.name,OLD.note);
    END$

CREATE TRIGGER TASK1UPDATEb AFTER UPDATE ON TheTable
    FOR EACH ROW
    BEGIN
        INSERT INTO LogTable (changeTimeStamp,action,tableId,name,note)
        VALUES(NOW(),'UPDATEa',NEW.id,NEW.name,NEW.note);
    END$

CREATE TRIGGER TASK1DELETE after DELETE ON TheTable
    FOR EACH ROW
    BEGIN
        INSERT INTO LogTable (changeTimeStamp,action,tableId,name,note)
        VALUES(NOW(),'DELETE',OLD.id,OLD.name,OLD.note);
    END$
delimiter ; -- Setter delimiter tilbake til ';'
```

Output:

```
> DELETE FROM TheTable WHERE id=4;
Query OK, 1 row affected (0.004 sec)

MariaDB [privBase]> select * from LogTable;
+----+-----------------+--------+---------+------+-------------+
| id | changeTimeStamp | action | tableId | name | note        |
+----+-----------------+--------+---------+------+-------------+
|  1 | 2020-02-04      | DELETE |       4 | Hege | Ikke Adrian |
+----+-----------------+--------+---------+------+-------------+
1 row in set (0.000 sec)

> INSERT INTO TheTable (name,note) VALUES ('Hege','Hege');
Query OK, 1 row affected (0.009 sec)

> UPDATE TheTable SET note = 'Ikke Adrian' where id=5;
Query OK, 1 row affected (0.009 sec)
Rows matched: 1  Changed: 1  Warnings: 0

> select * from LogTable;
+----+-----------------+--------+---------+------+-------------+
| id | changeTimeStamp | action | tableId | name | note        |
+----+-----------------+--------+---------+------+-------------+
|  1 | 2020-02-04      | DELETE |       4 | Hege | Ikke Adrian |
|  2 | 2020-02-04      | INSERT |       5 | Hege | Hege        |
|  3 | 2020-02-04      | UPDATE |       5 | Hege | Hege        |
|  4 | 2020-02-04      | UPDATE |       5 | Hege | Ikke Adrian |
+----+-----------------+--------+---------+------+-------------+
4 rows in set (0.000 sec)
```

Task 2: Temporal database

Code:

```
INSERT INTO AnotherTable (name,note)
VALUES  ('1','2'),
        ('2','3'),
        ('4','5'),
        ('Adrian','Adrian');

Select *, row_start,row_end FROM AnotherTable;

UPDATE AnotherTable SET note='Fortsatt Adrian' Where ID=4;

Select *, row_start,row_end FROM AnotherTable;

DELETE FROM AnotherTable WHERE id=2;

Select *, row_start,row_end FROM AnotherTable;

INSERT INTO AnotherTable (name,note)
VALUES  ('Hege','Ikke Adrian');

SELECT *, row_start,row_end FROM AnotherTable;
```

Output:

```
MariaDB [privBase]> source /home/admo/git/DAT151/Oblig3/Task2/Task2.sql
Query OK, 4 rows affected (0.009 sec)
Records: 4  Duplicates: 0  Warnings: 0

+----+--------+--------+----------------------------+----------------------
------+
| id | name   | note   | row_start                  | row_end
|
+----+--------+--------+----------------------------+----------------------
------+
|  1 | 1      | 2      | 2020-02-04 13:32:19.924915 | 2038-01-18
22:14:07.999999 |
|  2 | 2      | 3      | 2020-02-04 13:32:19.924915 | 2038-01-18
22:14:07.999999 |
|  3 | 4      | 5      | 2020-02-04 13:32:19.924915 | 2038-01-18
22:14:07.999999 |
|  4 | Adrian | Adrian | 2020-02-04 13:32:19.924915 | 2038-01-18
22:14:07.999999 |
+----+--------+--------+----------------------------+----------------------
------+
4 rows in set (0.000 sec)
```

Task 3: Integrity Constraints

Code:

```
CREATE TABLE teacher(
    id SMALLINT(1) UNSIGNED NOT NULL AUTO_INCREMENT,
    name VARCHAR(80),
    salary DOUBLE NOT NULL CHECK(salary BETWEEN 1000 AND 100000), --CHECK
vil gi en feilmelding om det er feile verdier som kommer inn.
    bonus DOUBLE,
    total DOUBLE as (bonus+salary), -- MariaDB syntax, I en annen
databasetype ville det kanskje vært gjort med triggers.
    CONSTRAINT teacherPK PRIMARY KEY (id)
);

-- total kunne også vært gjort med trigger (before) insert og update
```

Output:

```
> INSERT INTO teacher (name,salary,bonus) values ('Adrian',100,200);
ERROR 4025 (23000): CONSTRAINT `teacher.salary` failed for
`privBase`.`teacher`
> INSERT INTO teacher (name,salary,bonus) values ('Adrian',1001,200);
Query OK, 1 row affected (0.004 sec)

MariaDB [privBase]> select * from teacher;
+----+--------+--------+-------+-------+
| id | name   | salary | bonus | total |
+----+--------+--------+-------+-------+
|  1 | Adrian |   1001 |   200 |  1201 |
+----+--------+--------+-------+-------+
1 row in set (0.000 sec)
```

Task4: Order of triggers

```
--a
--Finnes i ../Tables.sql
--b
delimiter $
CREATE TRIGGER tr12 BEFORE INSERT ON t1
    FOR EACH ROW
    BEGIN
        INSERT INTO t2 (note)
        VALUES('tr12');
    END$

CREATE TRIGGER tr23 AFTER INSERT ON t2
    FOR EACH ROW
    BEGIN
        INSERT INTO t3 (note)
        VALUES('tr23');
    END$

CREATE TRIGGER tr13 AFTER INSERT ON t1
    FOR EACH ROW
    BEGIN
        INSERT INTO t3 (note)
        VALUES('tr13');
    END$
delimiter ;
INSERT INTO t1 (note)
values ('Start');

select * from t1;
select * from t2;
select * from t3;
```

Output:

```
> select * from t1;
+----+---------------------+-------+
| id | fyllt               | note  |
+----+---------------------+-------+
|  1 | 2020-02-04 14:17:08 | Start |
+----+---------------------+-------+
1 row in set (0.000 sec)

> select * from t2;
+----+---------------------+------+
| id | fyllt               | note |
+----+---------------------+------+
|  1 | 2020-02-04 14:17:08 | tr12 |
+----+---------------------+------+
1 row in set (0.000 sec)

> select * from t3;
+----+---------------------+------+
| id | fyllt               | note |
+----+---------------------+------+
|  1 | 2020-02-04 14:17:08 | tr23 |
|  2 | 2020-02-04 14:17:08 | tr13 |
+----+---------------------+------+
2 rows in set (0.000 sec)
```

Som vi ser kommer tr23 (t2 først inn til t3) Altså tr12 fyres først og så tr23

Task 5: Pendant delete

Code:

```
INSERT INTO Parent (name) VALUES
    ('Parent1'),
    ('Parent2'),
    ('Parent3');

INSERT INTO Child (name, parent) VALUES
    ('Child1',1),
    ('Child2',1),
    ('Child3',1),
    ('Child4',2),
    ('Child5',2),
    ('Child6',3);

Select * from Parent;
Select * from Child;

-- Pendant Delete trigger
delimiter $
CREATE TRIGGER Pendant_Delete AFTER DELETE ON Child
FOR EACH ROW
BEGIN
    IF (Select count(*) from Child where parent=OLD.parent) < 1 THEN
        DELETE FROM Parent WHERE id=OLD.parent;
    END IF;
End$

delimiter ;
```

Output:

```
> source /home/admo/git/DAT151/Oblig3/Task5/Task5.sql
Query OK, 3 rows affected (0.004 sec)
Records: 3  Duplicates: 0  Warnings: 0

Query OK, 6 rows affected (0.014 sec)
Records: 6  Duplicates: 0  Warnings: 0

+----+---------+
| id | name    |
+----+---------+
|  1 | Parent1 |
|  2 | Parent2 |
|  3 | Parent3 |
+----+---------+
3 rows in set (0.000 sec)

+----+--------+--------+
| id | name   | parent |
+----+--------+--------+
|  1 | Child1 |      1 |
|  2 | Child2 |      1 |
|  3 | Child3 |      1 |
|  4 | Child4 |      2 |
|  5 | Child5 |      2 |
|  6 | Child6 |      3 |
+----+--------+--------+
6 rows in set (0.000 sec)

Query OK, 0 rows affected (0.009 sec)

> DELETE FROM Child WHERE id=6;
Query OK, 1 row affected (0.005 sec)

> select * from Parent;
+----+---------+
| id | name    |
+----+---------+
|  1 | Parent1 |
|  2 | Parent2 |
+----+---------+
2 rows in set (0.000 sec)
```

Task 6: Concurrency

For mange filer med kode. Se samling av kommandoer og filer som kjøres på github:

https://github.com/H571531/DAT151/tree/master/Oblig3/Task6