Praktikum 4.4

Praktikum Basis Data
Program Studi Manajemen Informatika
FMIPA, Universitas Riau

May 27, 2022

1 Pengantar

Pada praktikum 4.4 ini, kita akan mengenal dan mempelajari tentang konsep *View* di dalam SQL (MySQL).

2 Pengenalan View

3 Create View

Statemen `CREATE VIEW` membuat view baru di dalam database. Berikut ini adalah sintaks dasar statemen `CREATE VIEW`:

```
CREATE [OR REPLACE] VIEW [db_name.]view_name [(column_list)]
AS
  select-statement;
```

4 Drop Views

5 Show Views

6 Rename Views

Behavior	ON DELETE	ON UPDATE
RESTRICT	Ditolak	Ditolak
CASCADE	Data akan dihapus	Data akan ikut diubah
NO ACTION	Data dibiarkan	Data dibiarkan
SET NULL	Diubah jadi NULL	Diubah jadi NULL

1. Restrict

Restrict dalam bahasa indonesia adalah membatasi, maksudnya adalah data pada tabel induk tidak bisa di *delete* atau di *update* (reference key nya) bila data tersebut memiliki relasi pada tabel lain.

```
CREATE TABLE products (
    id INT NOT NULL,
    name VARCHAR(100) NOT NULL,
    price INT(11),
    PRIMARY KEY(id)
) ENGINE=INNODB;

CREATE TABLE customers (
    id INT NOT NULL,
    name VARCHAR(100) NOT NULL,
    PRIMARY KEY (id)
) ENGINE=INNODB;

CREATE TABLE product_order (
    id INT NOT NULL AUTO_INCREMENT,
    product_id INT NOT NULL,
    customer_id INT NOT NULL,
    PRIMARY KEY(id)
) ENGINE=INNODB;

ALTER TABLE product_order
        ADD CONSTRAINT fk_po_customers
        FOREIGN KEY (customer_id) REFERENCES customers(id)
        ON DELETE RESTRICT ON UPDATE RESTRICT;
ALTER TABLE product_order
        ADD CONSTRAINT fk_po_products
        FOREIGN KEY (product_id) REFERENCES products(id)
        ON DELETE RESTRICT ON UPDATE RESTRICT;
```

Restrict adalah sintak yang digunakan untuk menambahkan *refrential integrity constraint* dengan tipe *restrict*. Sekarang kita coba *insert* data kedalam tabel tersebut

```
INSERT INTO products(id,name,price) VALUES (1,'sunlight'
    ,15000);
INSERT INTO customers(id,name) VALUES (1,'vivianika');
INSERT INTO product_order(product_id,customer_id) VALUES(1,1);
```

Kemudian kita akan coba *update* (bukan update barisnya ya, tapi *primary key reference*) dan *delete* data yang ada pada tabel parent (*products* dan *customers*). Hasilnya:

```
MariaDB [testing]> UPDATE products SET id=5 WHERE id=1;
ERROR 1451 (23000): Cannot delete or update a parent row: a
    foreign key constraint fails (`testing`.`product_order`,
    CONSTRAINT `product_order_ibfk_1` FOREIGN KEY (`product_id
    `) REFERENCES `products` (`id`))

MariaDB [testing]> DELETE FROM customers WHERE id=1;
ERROR 1451 (23000): Cannot delete or update a parent row: a
    foreign key constraint fails (`testing`.`product_order`,
    CONSTRAINT `product_order_ibfk_2` FOREIGN KEY (`
    customer_id`) REFERENCES `customers` (`id`))
```

2. Cascade

Cascade dalam bahasa Indonesia adalah bertingkat, maksudnya adalah bila data pada tabel induk di *delete* atau di *update* maka secara otomatis data pada tabel lain yang `memiliki relasi` akan di *delete* atau di *update* juga.

Sekarang kita ganti behavior *update* dan *delete* dari table sebelumnya menjadi `Cascade`

```
ALTER TABLE product_order DROP FOREIGN KEY fk_po_customers;
ALTER TABLE product_order DROP FOREIGN KEY fk_po_products;

ALTER TABLE product_order
        ADD CONSTRAINT fk_po_customers
        FOREIGN KEY (customer_id) REFERENCES customers(id)
        ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE product_order
        ADD CONSTRAINT fk_po_products
        FOREIGN KEY (product_id) REFERENCES products(id)
        ON DELETE CASCADE ON UPDATE CASCADE;
```

Kemudian kita akan coba *update* (bukan update barisnya ya, tapi *primary key reference*) dan *delete* data yang ada pada tabel parent (*products* dan *customers*). Hasilnya:

```
MariaDB [testing]> UPDATE products SET id=5 WHERE id=1;
Query OK, 1 row affected (0.050 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [testing]> SELECT * FROM product_order;
+----+------------+-------------+
| id | product_id | customer_id |
+----+------------+-------------+
|  1 |          5 |           1 |
+----+------------+-------------+
1 row in set (0.001 sec)


MariaDB [testing]> DELETE FROM customers WHERE id=1;
Query OK, 1 row affected (0.048 sec)

MariaDB [testing]> SELECT * FROM product_order;
Empty set (0.001 sec)
```

3. No Action

   No Action ini setara dengan RESTRICT. MySQL menolak operasi delete atau update untuk tabel induk jika ada referensi dengan tabel lainnya.

   Sekarang kita ganti behavior *update* dan *delete* dari table sebelumnya menjadi `No Action`

```
ALTER TABLE product_order DROP FOREIGN KEY fk_po_customers;
ALTER TABLE product_order DROP FOREIGN KEY fk_po_products;

ALTER TABLE product_order
        ADD CONSTRAINT fk_po_customers
        FOREIGN KEY (customer_id) REFERENCES customers(id)
        ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE product_order
        ADD CONSTRAINT fk_po_products
```

```
        FOREIGN KEY (product_id) REFERENCES products(id)
        ON DELETE NO ACTION ON UPDATE NO ACTION;
```

Kemudian kita akan coba *update* (bukan update barisnya ya, tapi *primary key reference*) dan *delete* data yang ada pada tabel parent (*products* dan *customers*). Hasilnya:

```
MariaDB [testing]> UPDATE customers SET id=4 WHERE id=1;
ERROR 1451 (23000): Cannot delete or update a parent row: a
    foreign key constraint fails (`testing`.`product_order`,
    CONSTRAINT `fk_po_customers` FOREIGN KEY (`customer_id`)
    REFERENCES `customers` (`id`) ON DELETE NO ACTION ON
    UPDATE NO ACTION)
MariaDB [testing]> DELETE FROM products WHERE id=1;
ERROR 1451 (23000): Cannot delete or update a parent row: a
    foreign key constraint fails (`testing`.`product_order`,
    CONSTRAINT `fk_po_products` FOREIGN KEY (`product_id`)
    REFERENCES `products` (`id`) ON DELETE NO ACTION ON UPDATE
     NO ACTION)
MariaDB [testing]>
```

4. Set Null

   Set null adalah menghapus atau meng-update baris data dalam tabel induk dan men-set kolom atau beberapa kolom dalam tabel anak menjadi *null*. hal ini akan valid jika kolom *foreign key* tidak *not null*.

   Sekarang kita ganti behavior *update* dan *delete* dari table sebelumnya menjadi `SET NULL`

```
DROP TABLE product_order;

CREATE TABLE product_order (
    id INT NOT NULL AUTO_INCREMENT,
    product_id INT,
    customer_id INT,
    PRIMARY KEY(id)
) ENGINE=INNODB;

ALTER TABLE product_order
        ADD CONSTRAINT fk_po_customers
        FOREIGN KEY (customer_id) REFERENCES customers(id)
        ON DELETE SET NULL ON UPDATE SET NULL;
ALTER TABLE product_order
        ADD CONSTRAINT fk_po_products
        FOREIGN KEY (product_id) REFERENCES products(id)
        ON DELETE SET NULL ON UPDATE SET NULL;

INSERT INTO product_order(product_id,customer_id) VALUES(1,1);
```

Kemudian kita akan coba *update* (bukan update barisnya ya, tapi *primary key reference*) dan *delete* data yang ada pada tabel parent (*products* dan *customers*). Hasilnya:

```
MariaDB [testing]> SELECT * FROM product_order;
+----+------------+-------------+
| id | product_id | customer_id |
+----+------------+-------------+
|  1 |          1 |           1 |
+----+------------+-------------+
```

```
1 row in set (0.001 sec)

MariaDB [testing]> UPDATE products SET id=5 where id=1;
Query OK, 1 row affected (0.045 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [testing]> SELECT * FROM product_order;
+----+------------+-------------+
| id | product_id | customer_id |
+----+------------+-------------+
|  1 |       NULL |           1 |
+----+------------+-------------+
1 row in set (0.001 sec)

MariaDB [testing]> DELETE FROM customers WHERE id=1;
Query OK, 1 row affected (0.056 sec)

MariaDB [testing]> SELECT * FROM product_order;
+----+------------+-------------+
| id | product_id | customer_id |
+----+------------+-------------+
|  1 |       NULL |        NULL |
+----+------------+-------------+
1 row in set (0.001 sec)
```