KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Type Kelompok

Post Test Minggu Ke - 9

Fitra Ilyasa, Andhika Marcelino Purwanto, Murliana, Rahma Wati
120140048, 120140187, 120140076, 120140184
fitra.120140048@student.itera.ac.id
andhika.120140187@student.itera.ac.id
murliana.120140076@student.itera.ac.id
rahma.120140184@student.itera.ac.id

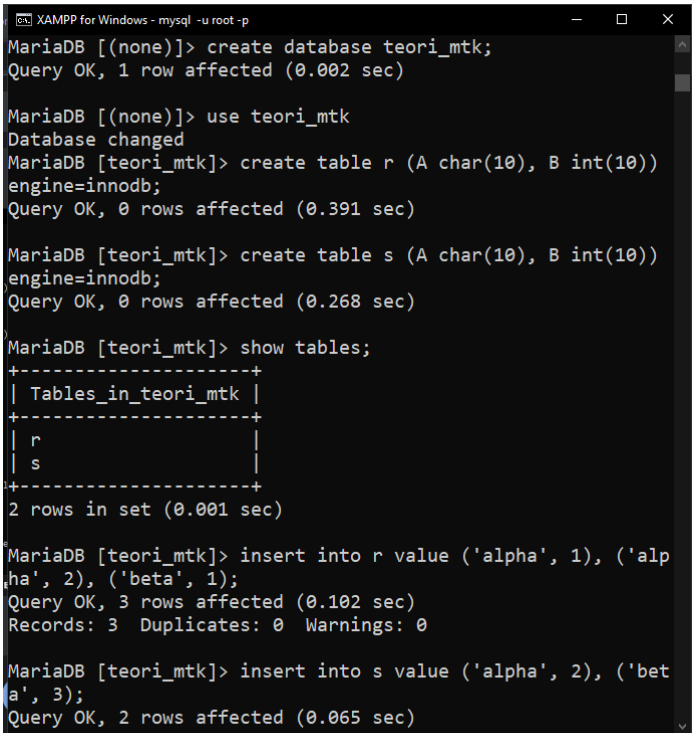
6 April 2022

1. Demonstrate

*use of the **relational algebra** operations from mathematical **set** theory (**union**, **intersection**, **difference**, and **Cartesian product**) and the **relational algebra** operations developed specifically for **relational databases** (**select** (restrict), **project**, **join**, and **division**)*

Definisi:

Relational Algebra (aljabar relasional) merupakan kumpulan operasi terhadap relasi dimana setiap operasi menggunakan satu atau lebih relasi untuk menghasilkan satu relasi yang baru. Aljabar relasional termasuk kategori prosedural dan juga menyediakan seperangkat operator untuk memanipulasi data.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Operasi UNION

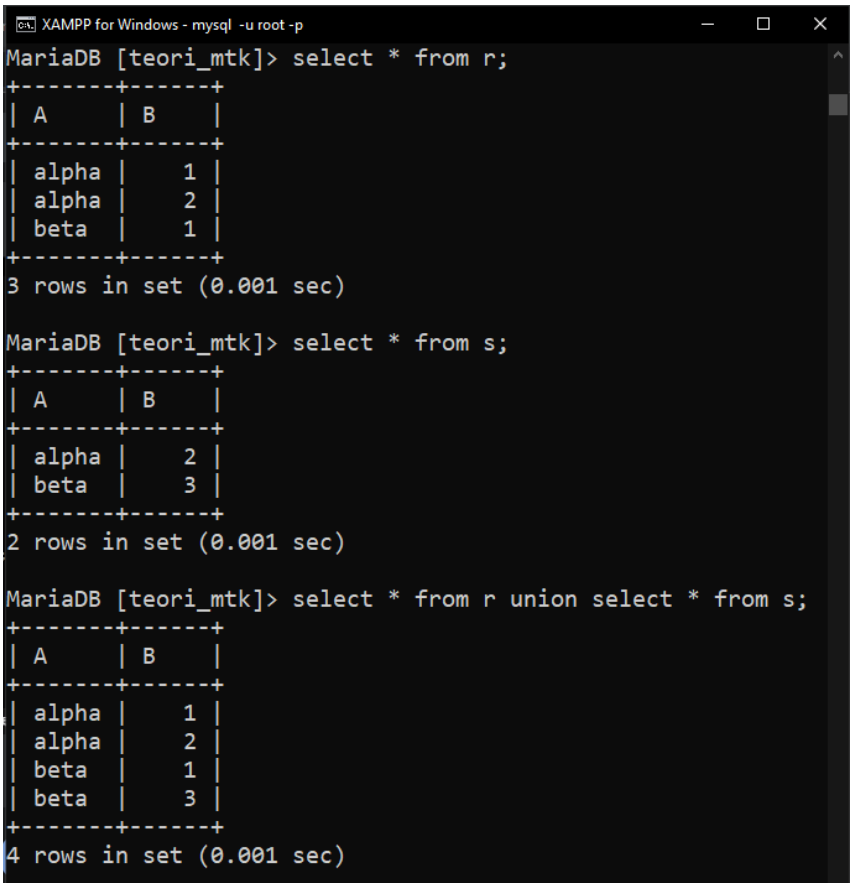
Union (∪), adalah operasi untuk menghasilkan gabungan table degan syarat kedua table memiliki atribut yangsama, yaitu domain atribut ke-i masing – masing table harus sama.

R U S = {x | x ∈ R atau x ∈ S}

Operasi ini dapat dilaksanakan apabila R dan S mempunyai atribut yang sama sehingga jumlah komponennya sama.

Notasi: R S dimana, R dan S adalah relasi,
simbol '∪' digunakan untuk menunjukkan operator Union.
Hasil operasi Union, yang dilambangkan dengan R S, adalah relasi yang pada dasarnya mencakup semua tupel yang ada di R atau di S, atau keduanya, menghilangkan tupel duplikat.
Poin penting tentang Operasi UNION:

1. Operasi UNION bersifat komutatif, yaitu : A B = B A
2. UNION bersifat asosiatif, artinya dapat diterapkan pada sejumlah relasi. A (B C) = (A B) C
3. Dalam SQL, operasi UNION sama dengan operasi UNION di sini.
4. Selain itu, Dalam SQL ada operasi multiset UNION ALL.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Operasi INTERSECTION (PERSIMPANGAN)

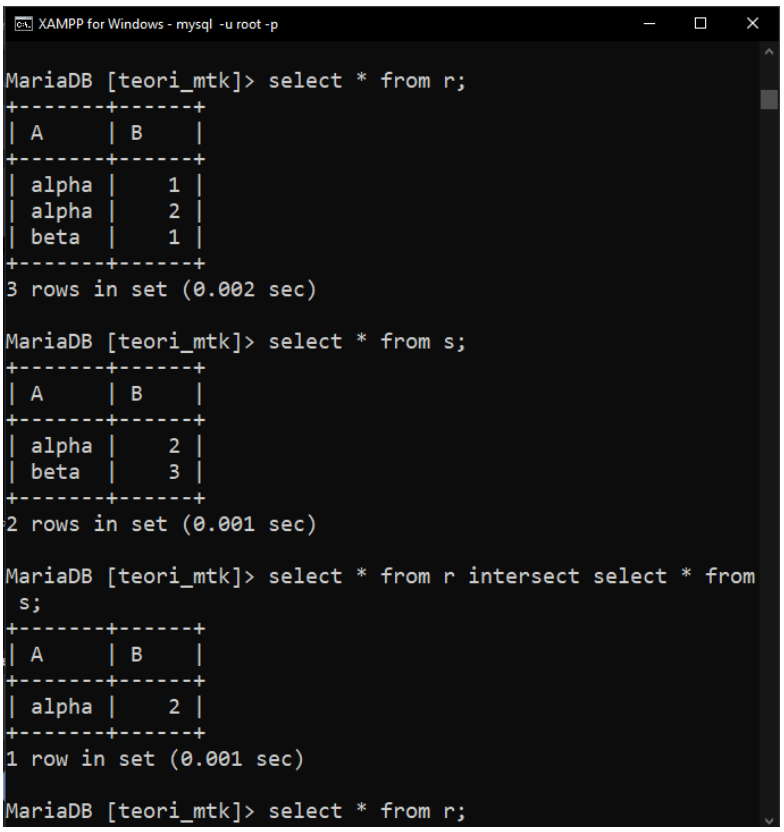
Set-intersection / Intersection (∩) termasuk kedalam operator tambahan, karena operator ini dapat diderivikasi dari operator dasar seperti berikut :

A ∩ B = A - (A – B), atau A ∩ B = B - (B –A)

Operasi ini merupakan operasi binary, yang digunakan untuk membentuk sebuah relasi baru dengan tuple yang berasal dari kedua relasi yang dihubungkan.

Notasi: R S dimana, R dan S adalah relasi,
simbol '∩' digunakan untuk menunjukkan operator titik-temu.
Hasil dari operasi Intersection, yang dilambangkan dengan R S, adalah sebuah relasi yang pada dasarnya mencakup semua tupel yang ada baik di R maupun S.
Poin-poin penting dalam Operasi INTERSECTION:

1. Operasi PERSIMPANGAN bersifat komutatif, yaitu : A B = B A
2. INTERSECTION bersifat asosiatif, artinya berlaku untuk sejumlah relasi. A(B C)=(A B)C
3. PERSIMPANGAN dapat dibentuk menggunakan UNION dan MINUS sebagai berikut: A B = ((A B) - (A - B)) - (B - A)
4. Dalam SQL, operasi INTERSECT sama dengan operasi INTERSECTION di sini.
5. Selain itu, Dalam SQL ada operasi multiset INTERSECT ALL.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Operasi SET DIFFERENCE (SET PERBEDAAN) / MINUS

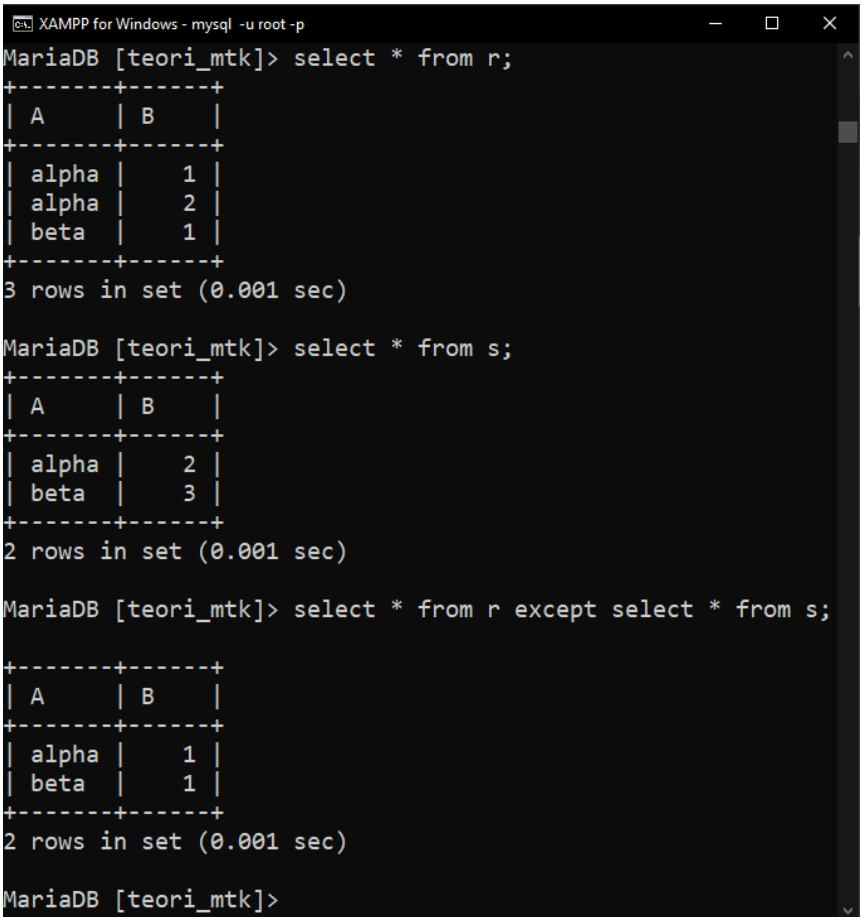
Set-intersection / Intersection (∩) termasuk kedalam operator tambahan, karena operator ini dapat diderivikasi dari operator dasar seperti berikut :

A ∩ B = A - (A – B), atau A ∩ B = B - (B –A)

Operasi ini merupakan operasi binary, yang digunakan untuk membentuk sebuah relasi baru dengan tuple yang berasal dari kedua relasi yang dihubungkan.

Notasi: R S dimana, R dan S adalah relasi,
simbol ' – ' digunakan untuk menunjukkan operator Minus.
Hasil dari operasi Intersection, yang dilambangkan dengan R – S, adalah relasi yang pada dasarnya mencakup semua tupel yang ada di R tetapi tidak ada di S.
Poin penting pada Operasi MINUS (atau SET PERBEDAAN):

1. Operasi EXCEPT tidak komutatif, artinya : A - B != B - A
2. Dalam SQL, operasi KECUALI sama dengan operasi EXCEPT di sini.
3. Selain itu, Dalam SQL ada operasi multiset EXCEPT ALL.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Operasi CARTESIAN PRODUCT

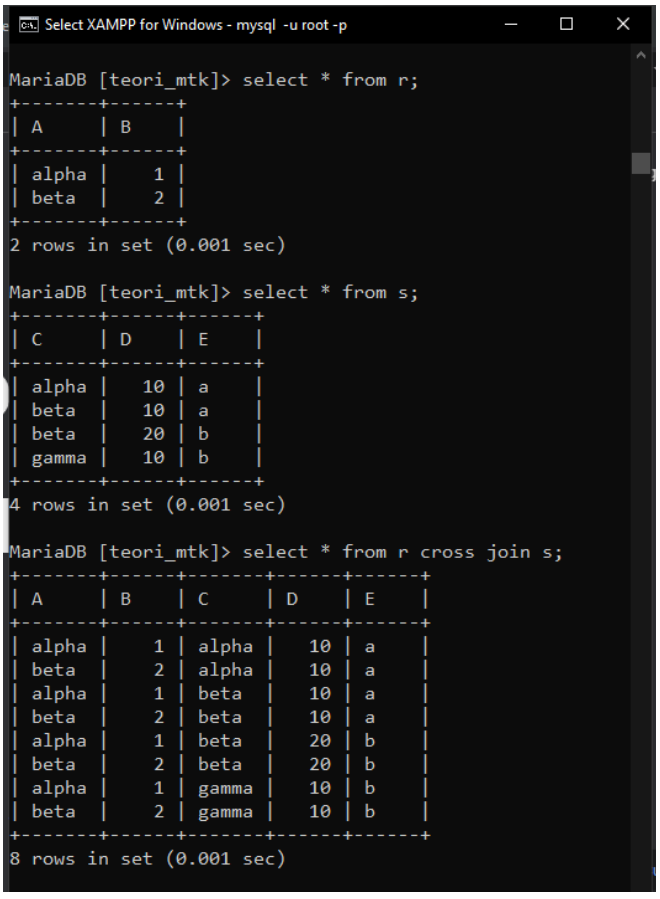
Cartesian-product (X), adalah operasi untuk menghasilkan table hasil perkalian kartesian. Sintaks yang digunakan dalam operasi proyeksi ini adalah sebagai berikut :

R X S = {(x,y) | x ∈ R dan y ∈ S}

Operasi cartesian-product memungkinkan kita mengkombinasikan informasi beberapa relasi, operasi ini adalah operasi biner.

Notasi: R S dimana R dan S adalah relasi,
simbol '✕' digunakan untuk menunjukkan operator CROSS PRODUCT.
Pada penerapan CARTESIAN PRODUCT pada dua relasi yaitu pada dua set tupel, ia akan mengambil setiap tupel satu per satu dari himpunan kiri (relasi) dan akan memasangkannya dengan semua tupel pada himpunan kanan (relasi).
Poin penting pada Operasi MINUS

1. Jadi, CROSS PRODUCT dari dua relasi A(R1, R2, R3, ..., Rp) dengan derajat p, dan B(S1, S2, S3, ..., Sn) dengan derajat n, adalah relasi C(R1, R2, R3, ..., Rp, S1, S2, S3, ..., Sn) dengan atribut derajat p + n.

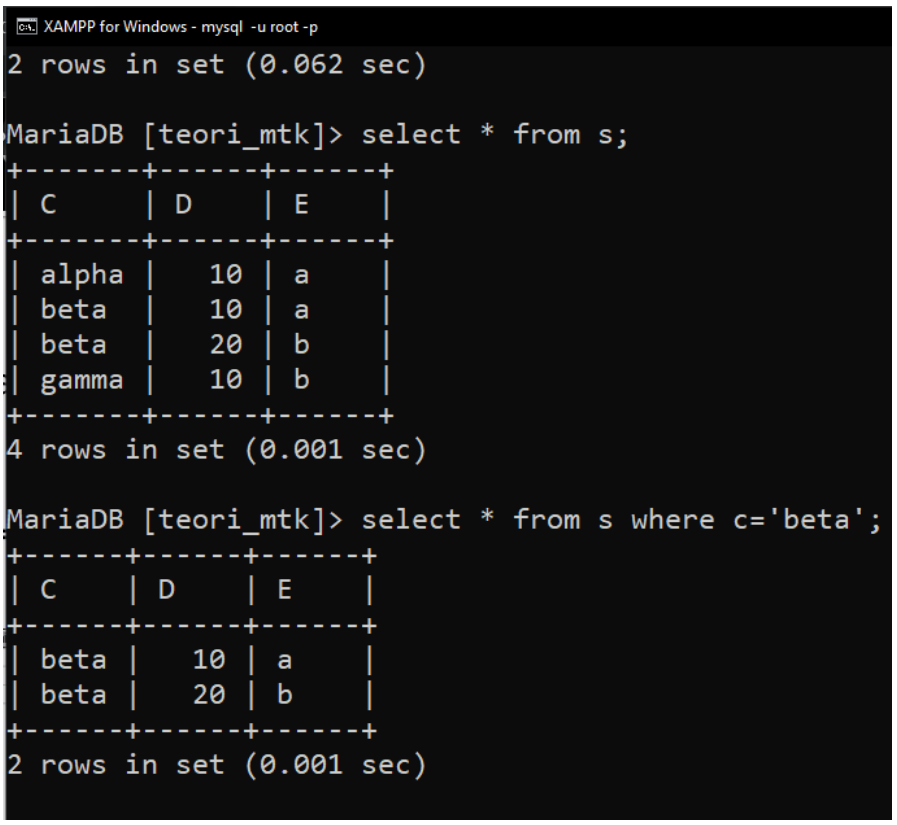


KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Operasi SELECTION / SELECT

Selection / Select (ς), adalah operasi untuk menyeleksi tupel – tupel yang memenuhi suatu predikat, kita dapat menggunakan operator perbandingan **(<,>,>=,<=,=,#)** pada predikat.

Beberapa predikat dapat dikombinasikan menjadi predikat manjemuk menggunakan penghubung **AND (∧) dan OR (∨).**

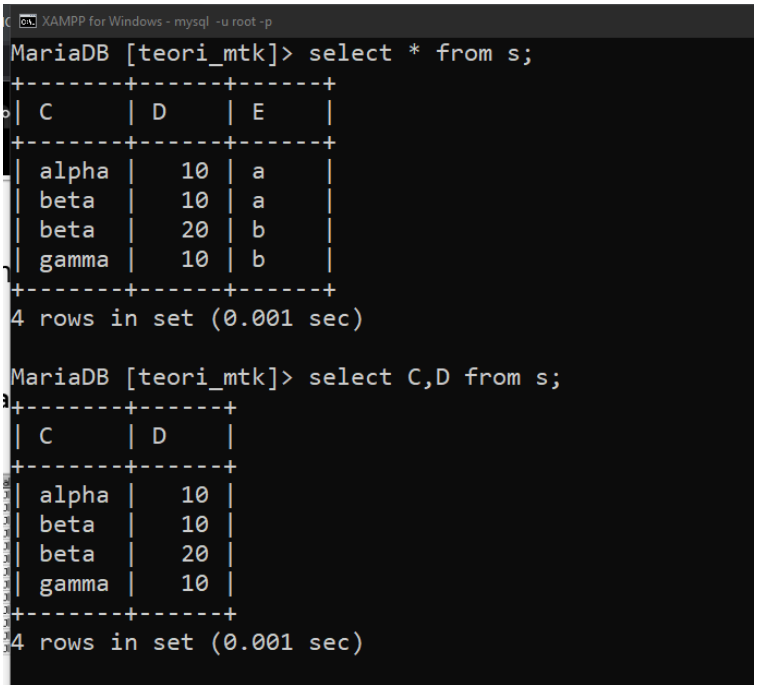


KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Operasi PROJECT

Projection / Project (π), adalah operasi untuk memperoleh kolom – kolom tertentu. Operasi project adalah operasi unary yang mengirim relasi argumen dengan kolom – kolom tertentu. Karena relasi adalah himpunan, maka baris – baris duplikasi dihilangkan. Sintaks yang digunakan dalam operasi proyeksi ini adalah sebagai berikut :

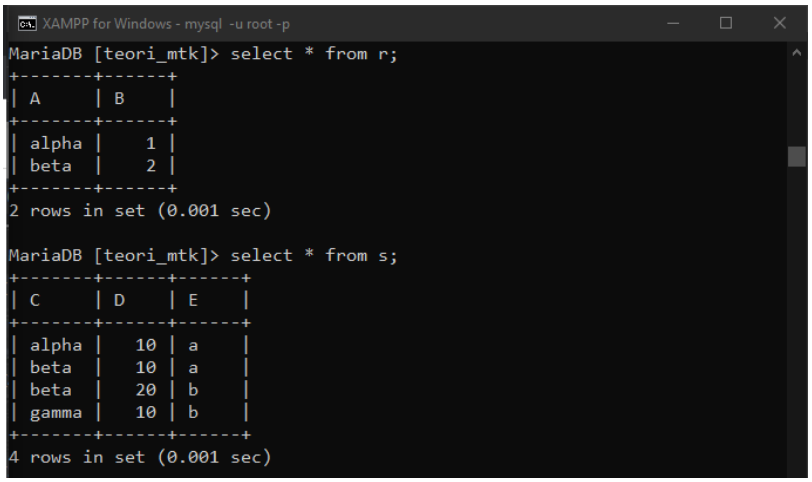
π colum1,...,column (tabel)



Operasi JOIN

- Theta-join dan equi-join adalah operasi untuk menggabungkan operasi selection dan cartesian-product dengan suatu kriteria.
- Natural-join sama seperti operasi tetha-join/equi-join adalah operasi untuk menggabungkan operasi selection dan cartesian-product dengan suatu kriteria pada kolom yang sama.
- Outer-join adalah operasi untuk menggabungkan operasi selection dan cartesian-product dengan suatu kriteria pada kolom yang sama.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id




```
MariaDB [teori_mtk]> select * from r theta join s;
+-------+------+-------+------+------+
| A     | B    | C     | D    | E    |
+-------+------+-------+------+------+
| alpha |    1 | alpha |   10 | a    |
| beta  |    2 | alpha |   10 | a    |
| alpha |    1 | beta  |   10 | a    |
| beta  |    2 | beta  |   10 | a    |
| alpha |    1 | beta  |   20 | b    |
| beta  |    2 | beta  |   20 | b    |
| alpha |    1 | gamma |   10 | b    |
| beta  |    2 | gamma |   10 | b    |
+-------+------+-------+------+------+
8 rows in set (0.001 sec)
```

```
MariaDB [teori_mtk]> select * from r equi join s;
+-------+------+-------+------+------+
| A     | B    | C     | D    | E    |
+-------+------+-------+------+------+
| alpha |    1 | alpha |   10 | a    |
| beta  |    2 | alpha |   10 | a    |
| alpha |    1 | beta  |   10 | a    |
| beta  |    2 | beta  |   10 | a    |
| alpha |    1 | beta  |   20 | b    |
| beta  |    2 | beta  |   20 | b    |
| alpha |    1 | gamma |   10 | b    |
| beta  |    2 | gamma |   10 | b    |
+-------+------+-------+------+------+
8 rows in set (0.001 sec)
```

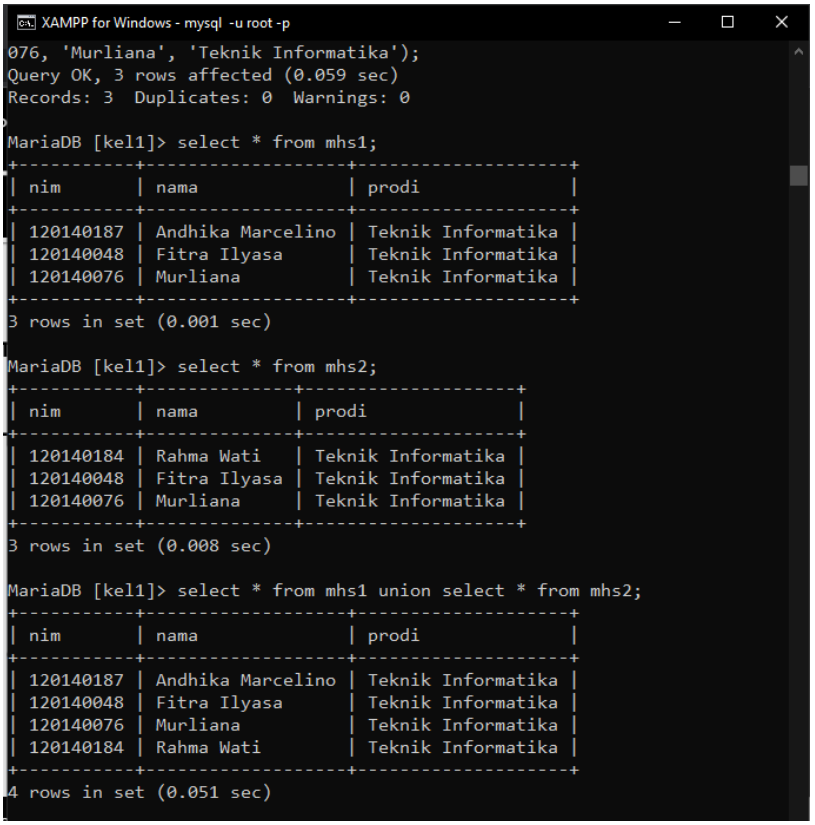
```
MariaDB [teori_mtk]> select * from r natural join s;
+-------+------+-------+------+------+
| A     | B    | C     | D    | E    |
+-------+------+-------+------+------+
| alpha |    1 | alpha |   10 | a    |
| beta  |    2 | alpha |   10 | a    |
| alpha |    1 | beta  |   10 | a    |
| beta  |    2 | beta  |   10 | a    |
| alpha |    1 | beta  |   20 | b    |
| beta  |    2 | beta  |   20 | b    |
| alpha |    1 | gamma |   10 | b    |
| beta  |    2 | gamma |   10 | b    |
+-------+------+-------+------+------+
8 rows in set (0.001 sec)
```

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

2. Demonstrate

*queries in the **relational algebra***

Aljabar relasional adalah sebuah bahasa query prosedural yang terdiri dari sekumpulan operasi dimana masukannya adalah satu atau dua relasi dan keluarannya adalah sebuah relasi baru sebagai hasil dari operasi tersebut.



3. Demonstrate

*queries in the **domain relational calculus (DRC)***

Domain Relational Calculus adalah bahasa query non-prosedural yang setara dengan *Tuple Relational Calculus*. *Domain Relational Calculus* hanya menyediakan deskripsi query tetapi tidak menyediakan metode untuk menyelesaikannya. Dalam Domain Relational Calculus, sebuah query dinyatakan sebagai, **{ < x1, x2, x3, ..., xn > | P (x1, x2, x3, ..., xn) }**

di mana, **< x1, x2, x3, ..., xn >** mewakili variabel domain yang dihasilkan dan **P (x1, x2, x3, .., xn)** mewakili kondisi atau rumus yang setara dengan kalkulus Predikat. Set dari semua operator perbandingan Himpunan kata penghubung seperti dan, atau, bukan Set quantifier.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Contoh:
Menggunakan database kel_1 dan membuat beberapa table, yaitu

- Table konsumen (nama_konsumen, alamat_konsumen, kota_konsumen)
- Table loan (nomor_loan, branch, amount)
- Table peminjam (nama_konsumen, nomor_loan)

```
MariaDB [(none)]> create database kel_1;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> use kel_1;
Database changed
MariaDB [kel_1]> create table konsumen (
    -> Nama_konsumen varchar(20) not null primary key,
    -> alamat_konsumen varchar(20) not null,
    -> kota_konsumen varchar(20) not null
    -> ) ENGINE = InnoDB;
Query OK, 0 rows affected (0.039 sec)

MariaDB [kel_1]> insert into konsumen values
    -> ("Fitra", "Simpang Jaya", "Garut"),
    -> ("Andika", "Kedaton", "Bandarlampung"),
    -> ("Rahma", "Pahoman", "Bandarlampung"),
    -> ("Murliana", "Seyegan", "Yogyakarta");
Query OK, 4 rows affected (0.048 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [kel_1]> select * from konsumen;
+---------------+----------------+---------------+
| Nama_konsumen | alamat_konsumen | kota_konsumen |
+---------------+----------------+---------------+
| Andika        | Kedaton        | Bandarlampung |
| Fitra         | Simpang Jaya   | Garut         |
| Murliana      | Seyegan        | Yogyakarta    |
| Rahma         | Pahoman        | Bandarlampung |
+---------------+----------------+---------------+
4 rows in set (0.004 sec)
```

```
MariaDB [(none)]> use kel_1;
Database changed
MariaDB [kel_1]> insert into loan values
    -> ("L01", "Sub", 230),
    -> ("L03", "Main", 150),
    -> ("L02", "Main", 75),
    -> ("L04", "Sub", 90);
Query OK, 4 rows affected (0.005 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [kel_1]> select * from loan;
+------------+--------+--------+
| nomor_loan | branch | amount |
+------------+--------+--------+
| L01        | Sub    |    230 |
| L02        | Main   |     75 |
| L03        | Main   |    150 |
| L04        | Sub    |     90 |
+------------+--------+--------+
4 rows in set (0.000 sec)
```

```
MariaDB [kel_1]> create table peminjam (
    -> nama_konsumen varchar(20) not null,
    -> nomor_loan varchar(3) primary key
    -> ) ENGINE = InnoDB;
Query OK, 0 rows affected (0.027 sec)

MariaDB [kel_1]> insert into peminjam values
    -> ("Fitra", "L01"),
    -> ("Andika", "L04"),
    -> ("Murliana", "L03"),
    -> ("Rahma", "L02");
Query OK, 4 rows affected (0.004 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [kel_1]> select * from peminjam;
+---------------+------------+
| nama_konsumen | nomor_loan |
+---------------+------------+
| Fitra         | L01        |
| Rahma         | L02        |
| Murliana      | L03        |
| Andika        | L04        |
+---------------+------------+
4 rows in set (0.000 sec)
```

Query-1:
Mencari nomor_loan, branch, loan lebih besar dari atau sama dengan 100

$$\{\langle l, b, a\rangle \mid \langle l, b, a\rangle \in loan \wedge (a \geq 100)\}$$

Untuk menampilkan nomor_loan, branch, dan loan kita menggunakan table loan dengan syntax select dengan kondisi amount >= 100.

```
MariaDB [kel_1]> select * from loan;
+------------+--------+--------+
| nomor_loan | branch | amount |
+------------+--------+--------+
| L01        | Sub    |    230 |
| L02        | Main   |     75 |
| L03        | Main   |    150 |
| L04        | Sub    |     90 |
+------------+--------+--------+
4 rows in set (0.001 sec)

MariaDB [kel_1]> select * from loan where amount >= 100;
+------------+--------+--------+
| nomor_loan | branch | amount |
+------------+--------+--------+
| L01        | Sub    |    230 |
| L03        | Main   |    150 |
+------------+--------+--------+
2 rows in set (0.006 sec)

MariaDB [kel_1]>
```

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI SUMATERA
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
Jalan Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365
Telepon: (0721) 8030188
Email: jtpi@itera.ac.id, Website : http://itera.ac.id

Query-2:
Temukan nomor_loan untuk setiap pinjaman dengan amount yang lebih besar atau sama dengan 150

$$\{<l> \mid \exists\ b,\ a\ (<l,\ b,\ a> \in loan \wedge (a \geq 150)\}$$

*Menampilkan data nomor_loan menggunakan syntax select from loan yang memiliki amount >= 150

```
MariaDB [kel_1]> select nomor_loan from loan where amount >= 150;
+------------+
| nomor_loan |
+------------+
| L01        |
| L03        |
+------------+
2 rows in set (0.000 sec)
```

Query-3:
Temukan nama_konsumen yang memiliki pinjaman di branch “Main” dan temukan amount

$$\{<c,\ a> \mid \exists\ l\ (<c,\ l> \in borrower \wedge \exists\ b\ (<l,\ b,\ a> \in loan \wedge (b = \text{“Main”}))\}$$

Untuk menampilkan nama_konsumen dan amount memerlukan dua table, yaitu table loan untuk mencari nomor_konsumen yang memiliki pinjaman di branch “Main”. Kemudian ke table peminjam untuk mencari nama_konsumen dan amountnya dengan menggunakan nomor_konsumen yang sudah ditemukan tadi.

```
MariaDB [kel_1]> select nomor_loan, amount from loan where branch = "Main";
+------------+--------+
| nomor_loan | amount |
+------------+--------+
| L02        |     75 |
| L03        |    150 |
+------------+--------+
2 rows in set (0.000 sec)

MariaDB [kel_1]> select nama_konsumen from peminjam where nomor_loan between "L02" and "L03";
+---------------+
| nama_konsumen |
+---------------+
| Murliana      |
| Rahma         |
+---------------+
2 rows in set (0.000 sec)
```

Keterangan simbol yang digunakan pada rumus:
l : nomor_loan
b : Branch
a : Amount
c : nama_konsumen
∧ : and/dan

4. Demonstrate

queries in the **tuple relational calculus (T*RC*).**
Tuple Relational Calculus adalah bahasa query non-prosedural tidak seperti aljabar relasional. Kalkulus Tuple hanya menyediakan deskripsi kueri tetapi tidak menyediakan metode untuk menyelesaikannya. Jadi, ini menjelaskan apa yang harus dilakukan tetapi tidak bagaimana melakukannya. Dalam Kalkulus Tuple, kueri dinyatakan sebagai :
{t| P(t)}
di mana **t** = tupel yang dihasilkan, **P(t)** = dikenal sebagai Predikat dan ini adalah kondisi yang digunakan untuk mengambil t Dengan demikian, menghasilkan himpunan semua tupel t, sehingga Predikat P(t) benar untuk t. P(t) mungkin memiliki berbagai kondisi yang digabungkan secara logis dengan **OR (∨), AND (∧), NOT().**
Ini juga menggunakan quantifier:
t r (Q(t)) = ”ada” sebuah tupel dalam t dalam relasi r sehingga predikat Q(t) benar.
t r (Q(t)) = Q(t) benar “untuk semua” tupel dalam relasi r.

Contoh :

1. Tabel Customer

```
MariaDB [peminjaman]> insert into customer value
    -> ("Andika", "A7", "Jakarta"),
    -> ("Fitra", "B6", "Garut"),
    -> ("Murli", "D9", "Lampung"),
    -> ("Rahma", "A5", "Jakarta");
Query OK, 4 rows affected (0.110 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [peminjaman]> select * from customer;
+---------------+--------+---------+
| customer_name | street | city    |
+---------------+--------+---------+
| Rahma         | A5     | Jakarta |
| Andika        | A7     | Jakarta |
| Fitra         | B6     | Garut   |
| Murli         | D9     | Lampung |
+---------------+--------+---------+
4 rows in set (0.001 sec)
```

2. Tabel Branch

```
MariaDB [peminjaman]> insert into branch value
    -> ("ABC", "Jakarta"),
    -> ("DEF", "Lampung"),
    -> ("GHI", "Garut");
Query OK, 3 rows affected (0.208 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [peminjaman]> select * from branch;
+-------------+-------------+
| branch_name | branch_city |
+-------------+-------------+
| ABC         | Jakarta     |
| DEF         | Lampung     |
| GHI         | Garut       |
+-------------+-------------+
3 rows in set (0.005 sec)
```

3. Tabel Account

```
MariaDB [peminjaman]> insert into account value
    -> (1111, "ABC", 50000),
    -> (1112, "DEF", 10000),
    -> (1113, "GHI", 9000),
    -> (1114, "ABC", 7000);
Query OK, 4 rows affected (0.190 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [peminjaman]> select * from account;
+----------------+-------------+---------+
| account_number | branch_name | balance |
+----------------+-------------+---------+
|           1111 | ABC         |   50000 |
|           1112 | DEF         |   10000 |
|           1113 | GHI         |    9000 |
|           1114 | ABC         |    7000 |
+----------------+-------------+---------+
4 rows in set (0.031 sec)
```

4. Tabel Loan

```
MariaDB [peminjaman]> insert into loan value
    -> ("L33", "ABC", 10000),
    -> ("L35", "DEF", 15000),
    -> ("L49", "GHI", 9000),
    -> ("L98", "DEF", 65000);
Query OK, 4 rows affected (0.153 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [peminjaman]> select * from loan;
+-------------+-------------+--------+
| loan_number | branch_name | amount |
+-------------+-------------+--------+
| L33         | ABC         |  10000 |
| L35         | DEF         |  15000 |
| L49         | GHI         |   9000 |
| L98         | DEF         |  65000 |
+-------------+-------------+--------+
4 rows in set (0.001 sec)
```

5. Tabel Borrower

```
MariaDB [peminjaman]> insert into loan value
    -> ("L33", "ABC", 10000),
    -> ("L35", "DEF", 15000),
    -> ("L49", "GHI", 9000),
    -> ("L98", "DEF", 65000);
Query OK, 4 rows affected (0.153 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [peminjaman]> select * from loan;
+-------------+-------------+--------+
| loan_number | branch_name | amount |
+-------------+-------------+--------+
| L33         | ABC         |  10000 |
| L35         | DEF         |  15000 |
| L49         | GHI         |   9000 |
| L98         | DEF         |  65000 |
+-------------+-------------+--------+
4 rows in set (0.001 sec)
```

6. Tabel Depositor

```
MariaDB [peminjaman]> insert into depositor value
    -> ("Andika", 1111),
    -> ("Fitra", 1113),
    -> ("Murli", 1114);
Query OK, 3 rows affected (0.193 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [peminjaman]> select * from depositor;
+---------------+----------------+
| customer_name | account_number |
+---------------+----------------+
| Andika        |           1111 |
| Fitra         |           1113 |
| Murli         |           1114 |
+---------------+----------------+
3 rows in set (0.001 sec)
```

- Queries 1 : Temukan loan number, branch, amount >= 10000

$$\{t \mid t \in loan \ \wedge t[amount] >= 10000\}$$

```
MariaDB [peminjaman]> select * from loan where amount >=10000;
+-------------+-------------+--------+
| loan_number | branch_name | amount |
+-------------+-------------+--------+
| L33         | ABC         |  10000 |
| L35         | DEF         |  15000 |
| L98         | DEF         |  65000 |
+-------------+-------------+--------+
3 rows in set (1.543 sec)
```

Pada query diatas t [amount] sebagai variabel tuple

- Queries 2 : temukan loan number dengan jumlah amount >= 10000

$$\{t \mid \exists s \in loan(t[loan\ number] = s[loan\ number] \wedge s[amount] >= 10000)\}$$

```
MariaDB [peminjaman]> select loan_number from loan where amount >= 10000;
+-------------+
| loan_number |
+-------------+
| L33         |
| L35         |
| L98         |
+-------------+
3 rows in set (0.158 sec)

MariaDB [peminjaman]>
```

- Queries 3 : temukan semua nama pada customer_name yang memiliki loan number dari depositor

$$\{t \mid \exists s \in borrower(t[customer\text{-}name] = s[customer\text{-}name]) \wedge \exists u \in depositor([customer\text{-}name] = u[$$

```
MariaDB [peminjaman]> select customer.customer_name from depositor natural join customer;
+---------------+
| customer_name |
+---------------+
| Andika        |
| Fitra         |
| Murli         |
+---------------+
3 rows in set (0.391 sec)
```