**MODUL MATA KULIAH**

**PENGANTAR SISTEM BASIS DATA**

**KP213 - 2 SKS**

FAKULTAS TEKNOLOGI INFORMASI

**UNIVERSITAS BUDI LUHUR**



JAKARTA SEPTEMBER 2019

T I M

P E N Y U S U N

Nama 1

Nama 2

Nama 3

|  |  |  |
| --- | --- | --- |
|  | UNIVERSITAS BUDI LUHUR  FAKULTAS TEKNOLOGI INFORMASI |  |

MODUL PERKULIAHAN #7

**ALJABAR RELASIONAL LANJUTAN**

|  |  |  |
| --- | --- | --- |
| Capaian Pembelajaran | : | Mahasiswa mampu memahami penggunaan Aljabar Relasional Lanjutan |
| Sub Pokok Bahasan | : | 1.1. Binary Operations (Theta Join, Natural Join, Outer Join, Semi Join, Division)  1.2. Aggregate Operations  1.3. Grouping Operations |
| Daftar Pustaka | : | 1. Database System A Practical Approach to Design, Implementation, and Management by Thomas M. Connolly, Carolyn E. Begg  2. An Introduction to Database Systems by C.J. Date |

* 1. **BINARY OPERATIONS (THETA JOIN, NATURAL JOIN, OUTER JOIN, SEMI JOIN, DIVISION)**
* Operasi atau perintah Join adalah turunan dari operasi Cartesian product.
* Sama seperti menggunakan operasi Selection, ada Predicate Join sebagai formula yang diterapkan pada Cartesian Product dari 2 relasi sebagai operandnya.

**Theta join (θ-join)**

* **R F S** 
  + Menghasilkan sebuah relasi yang terdiri dari tuple atau record yang memenuhi syarat pada bagian predicate F, dari Cartesian Product antara relasi R dan S.
  + Predicate F bisa menggunakan salah satu dari operator perbandingan ini (<, ≤, >, ≥, =, ≠).
* Theta join dapat ditulis ulang menggunakan kombinasi perintah dasar Selection dan Cartesian product.

**R F S = σF(R Χ S)**

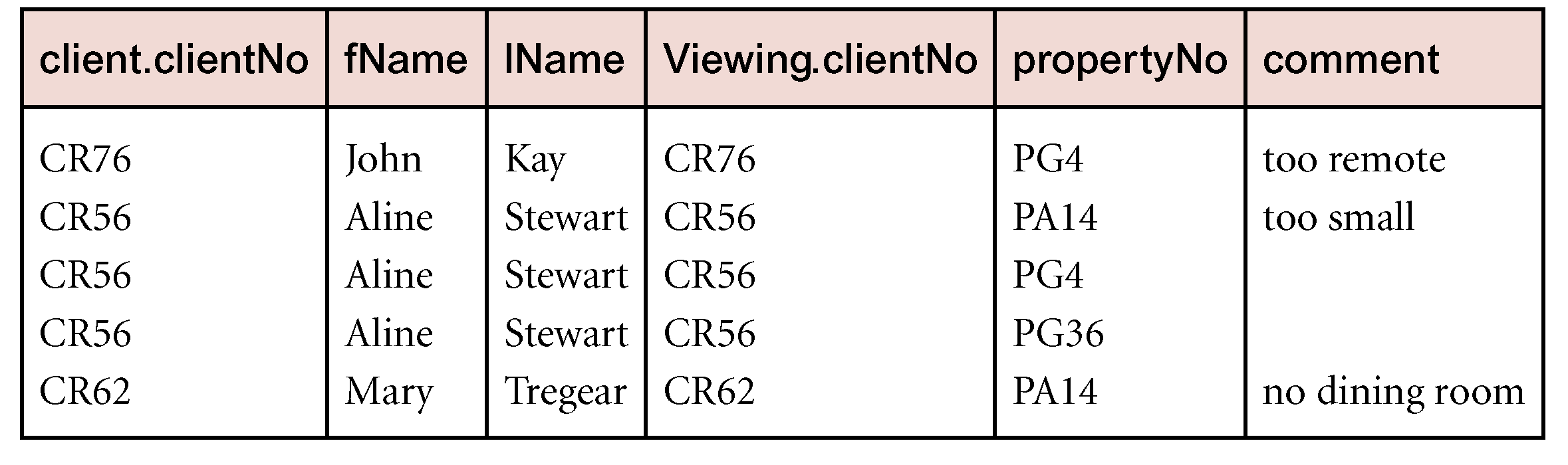
* Jika predicate F terdiri hanya operator equality (=), maka Theta join bisa disebut sebagai Equijoin.
* Contoh :

Tampilkan seluruh nama klien dan komentar mereka dari para klien yang sudah melihat property yang disewakan.

Jika dilihat dari perintah di atas, maka ada 2 tabel yang terlibat, yaitu table Client untuk mendapatkan data nama klien, dan table Viewing untuk mendapatkan data property yang dilihat dan komentarnya. Oleh karena itu relasi yang akan di-join-kan adalah table Client dan Viewing. Dari table Client, kolom yang akan dibawa pada saat Join adalah clientNo (untuk disesuaikan dengan kolom clientNo pada table Viewing), fName dan lName. Sementara dari table Viewing kolom yang akan dibawa adalah clientNo (untuk disesuaikan dengan kolom clientNo pada table Client), propertyNo dan comment

Relational Algebranya adalah :

(ΠclientNo, fName, lName(Client)) Client.clientNo = Viewing.clientNo (ΠclientNo, propertyNo, comment(Viewing))

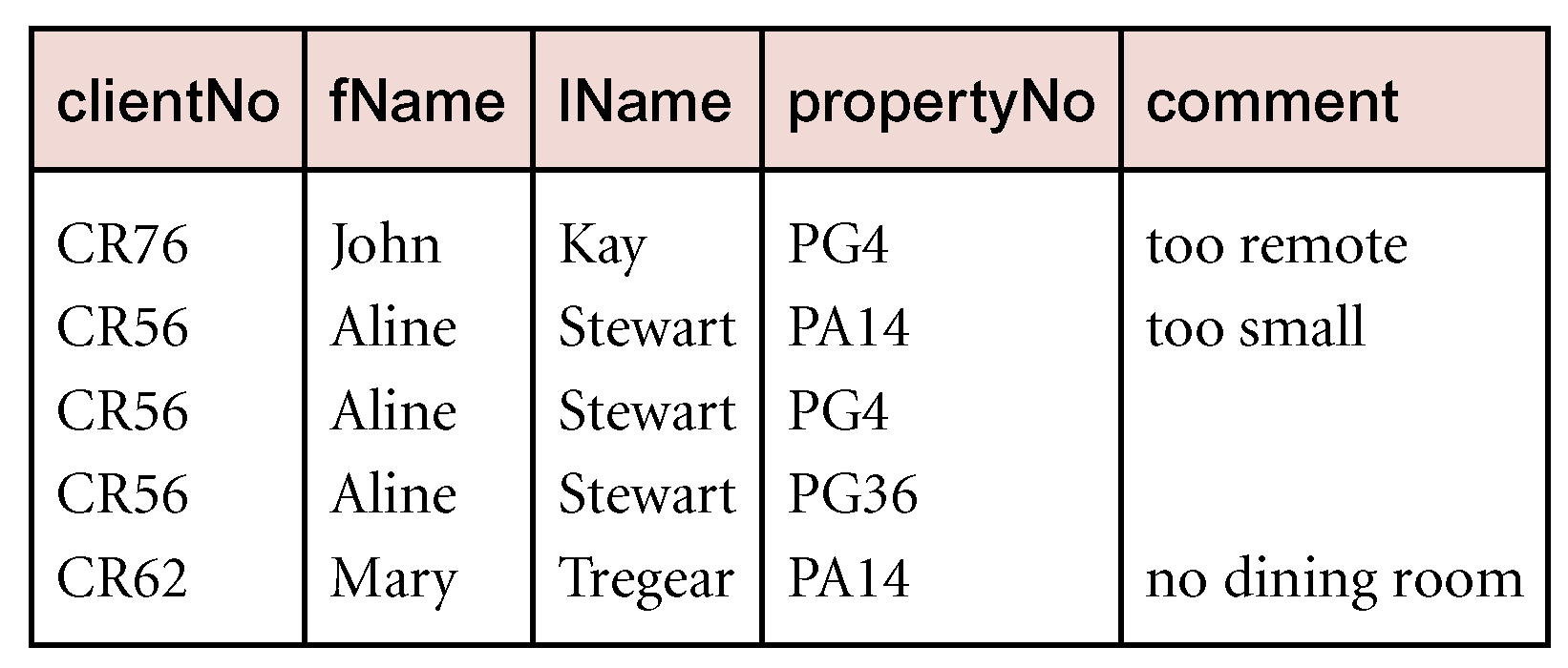


**Natural join**

* **R S** 
  + Sebuah Equijoin dari dua relasi R dan S atas semua atribut umum x. Satu kemunculan dari setiap atribut umum dihilangkan dari hasilnya.
* Example :

Tampilkan seluruh nama klien dan komentar mereka dari para klien yang sudah melihat property yang disewakan.

(ΠclientNo, fName, lName(Client)) (ΠclientNo, propertyNo, comment(Viewing))

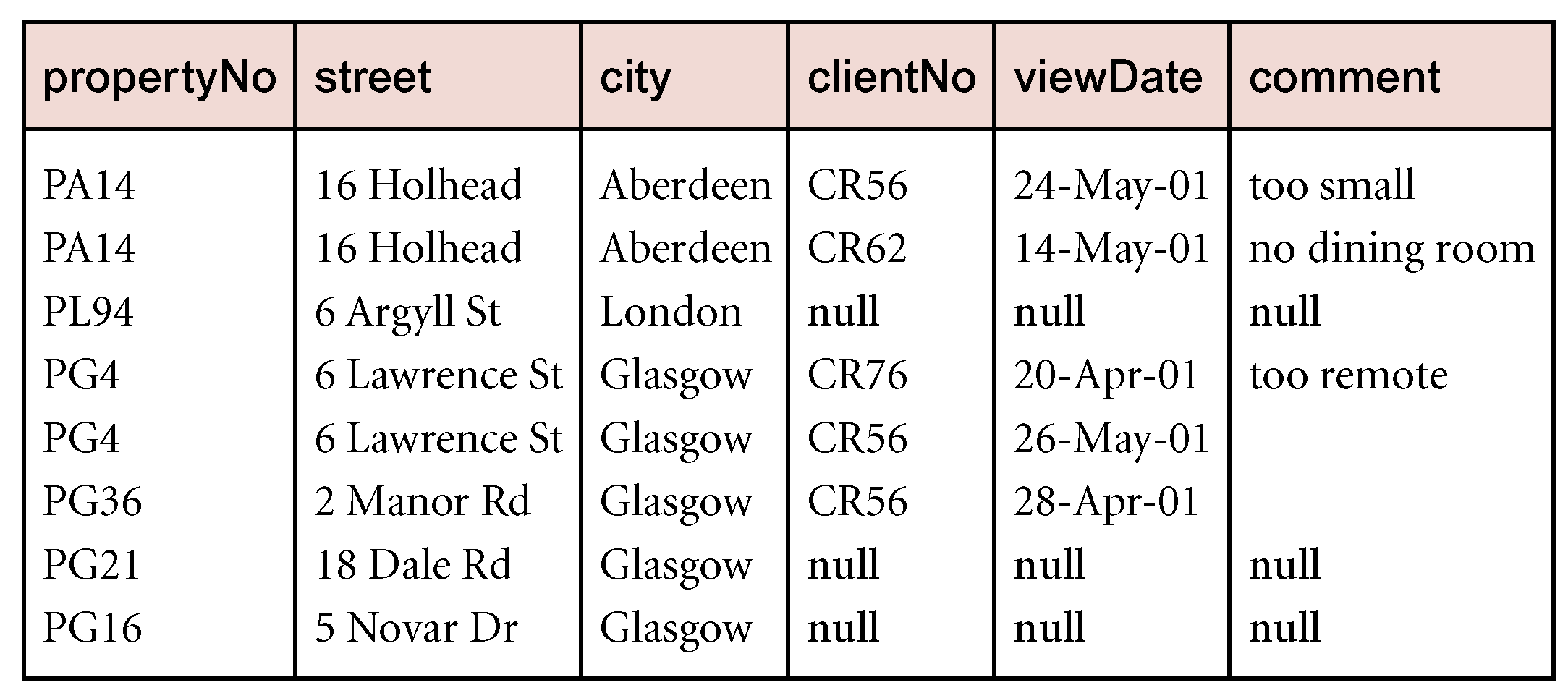


**Outer join**

* Outer Join digunakan untuk menampilkan record yang tidak memiliki nilai yang sesuai pada kolom yang dipakai untuk proses join.
* 
  + (Left) outer join adalah join di mana record dari relasi R yang tidak memiliki matching values pada kolom yang sama di relasi R juga ikut dimasukkan ke dalam relasi hasil.
* Contoh :

Tampilkan data status Viewing dari seluruh property.

ΠpropertyNo, street, city(PropertyForRent) Viewing



Karena relasi atau tabel PropertyForRent berada di sisi kiri dari simbol left outer join, maka semua record dari tabel PropertyForRent muncul, meskipun ada record yang tidak berpasangan atau tidak ditemukan data propertyNo nya di tabel Viewing. Jika ada data dari tabel PropertyForRent tidak ditemukan, maka di sisi Viewingnya akan ditampilkan sebagai null, seperti pada hasil relational algebra di atas untuk propertyNo PG21 dan PG16, karena tidak ada klien yang melihat property dengan kode tersebut (lihat di tabel Viewing).

**Semijoin**

* **R F S** 
  + Menghasilkan sebuah relasi yang terdiri dari tuple dari relasi R yang berpartisipasi pada proses join antara relasi R dan S.
* Semijoin dapat ditulis menggunakan Projection dan Join :

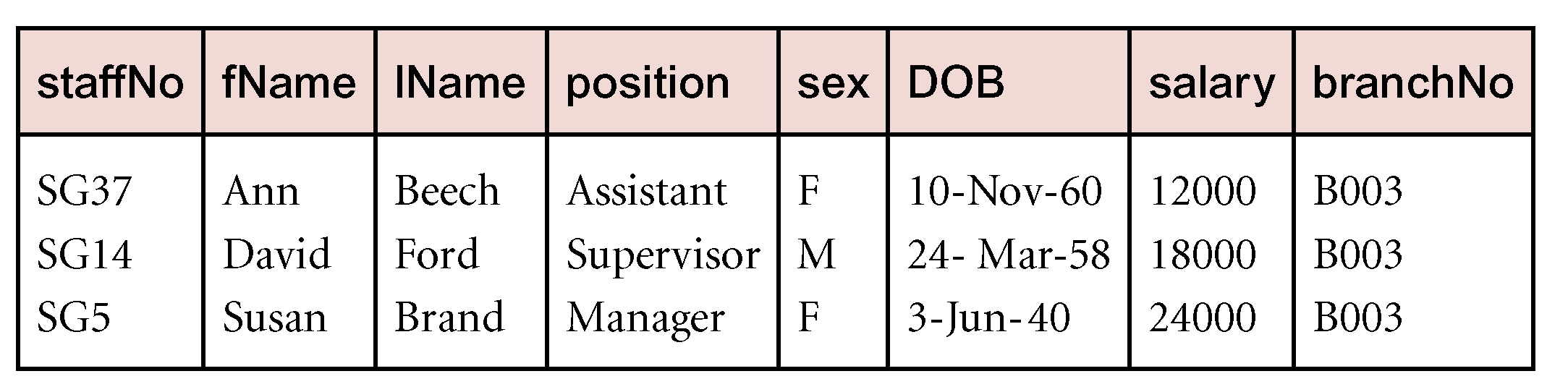


* Contoh :

Tampilkan data lengkap staff yang bekerja pada kantor cabang di Glasgow**.**

Staff Staff.branchNo=Branch.branchNo(σcity=‘ Glasgow’(Branch))

Maka relational algebra ini akan menghasilkan data ini :



**Division**

* **R** ÷ **S**
  + Menghasilkan sebuah relasi melalui atribut C yang terdiri atas record dari relasi R yang sesuai dengan tiap record atau tuple dari relasi.
  + Jika diekspresikan menggunakan operasi dasar:

T1 ← ΠC(R)

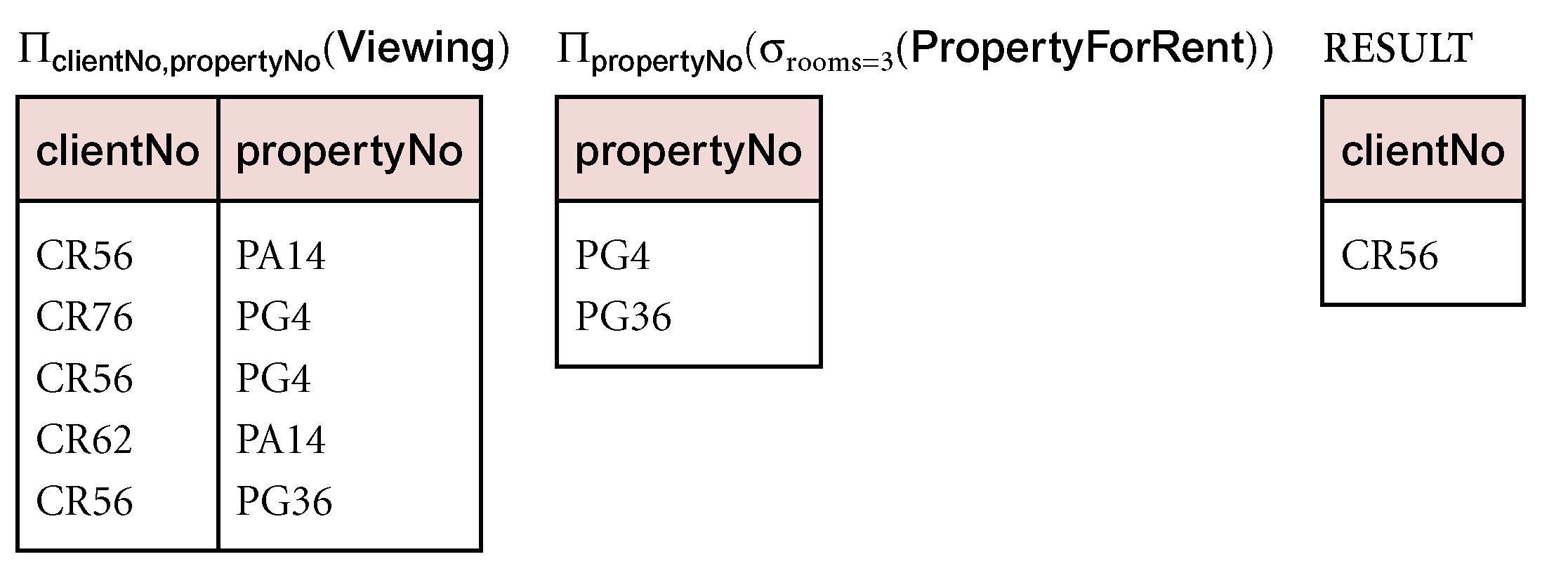
T2 ← ΠC((S X T1) – R)

T ← T1 – T2

* Contoh :

Tampilkan seluruh klien yang sudah melihat seluruh property yang memiliki 3 kamar.

(ΠclientNo, propertyNo(Viewing)) ÷ (ΠpropertyNo(σrooms = 3 (PropertyForRent)))



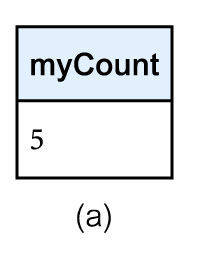
* 1. **AGGREGATE OPERATIONS**

**ℑAL(R)**

* + Menerapkan serangkaian fungsi agregat AL pada relasi R.
  + AL terdiri atas satu atau lebih pasangan (<fungsi agregat>, <atribut>)**.**
  + Fungsi agregat utama adalah: COUNT, SUM, AVG, MIN, and MAX.
  + Contoh :

Berapa jumlah property yang biaya sewanya lebih dari £350 per bulan?

ρR(myCount) ℑCOUNT propertyNo (σrent > 350 (PropertyForRent))

****

ρR(myCount) berfungsi untuk memberikan nama kolom, dalam hal ini adalah untuk kolom Count propertyNo.

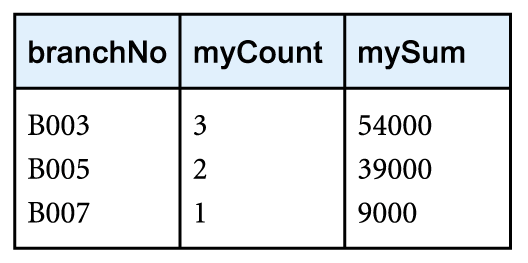
* 1. **GROUPING OPERATIONS**

**GA**ℑ**AL(R)**

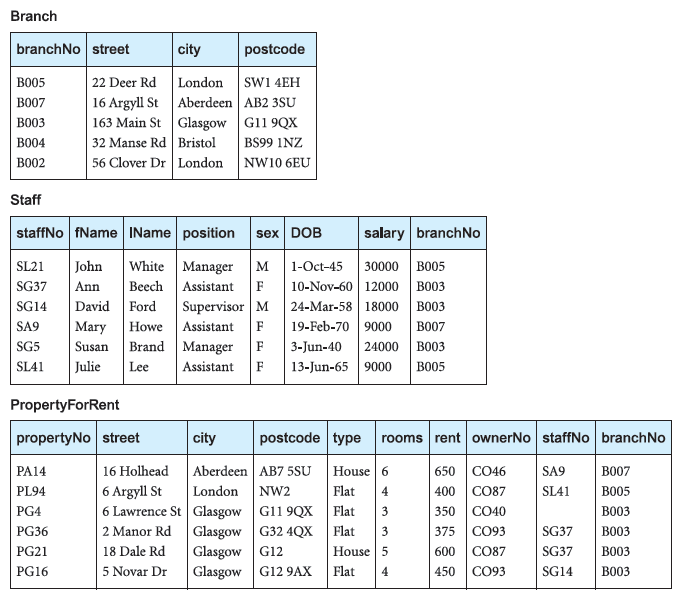
* + Mengelompokkan record dari relas R melalui attribute GA, kemudian menerapkan fungsi agregat AL untuk menghasilkan relasi yang baru.
  + AL terdiri atas satu atau lebih pasangan (<fungsi agregat>, <atribut>).
  + Relasi hasil terdiri atas atribut pengelompokkan GA, bersamaan dengan hasil dari tiap fungsi agregat AL.
  + Contoh :

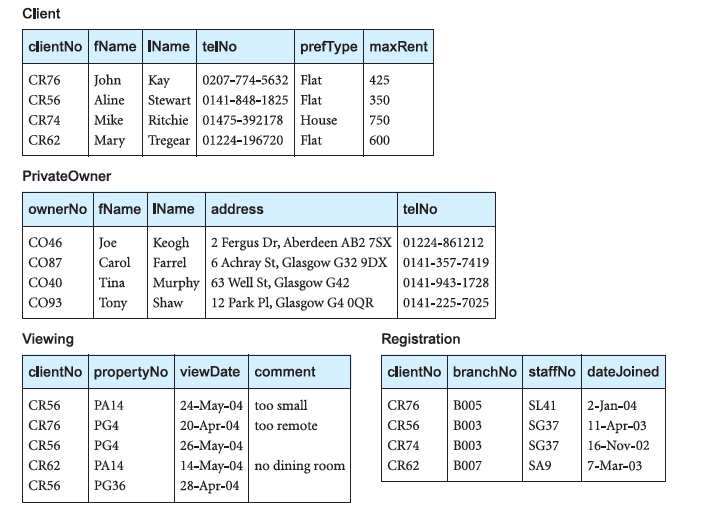
Tampilkan jumlah staf yang bekerja di tiap cabang dan total gajinya.

ρR(branchNo, myCount, mySum) branchNo ℑ COUNT staffNo, SUM salary (Staff)



**Relasi-relasi yang digunakan pada modul ini :**





**LATIHAN**

1. Buatlah relational algebra untuk menampilkan data staf dan property yang dikelolanya menggunakan semijoin !
2. Buatlah relational algebra untuk menampilkan data staf dan property yang dikelolanya menggunakan natural join !
3. Buatlah relational algebra untuk menampilkan data staf dan property yang dikelolanya menggunakan left outer join !
4. Buatlah relational algebra untuk menampilkan jumlah property yang disewakan berdasarkan kantor cabangnya !
5. Buatlah relational algebra untuk menampilkan jumlah property yang disewakan berdasarkan kantor cabangnya, dan tampilkan juga alamat kantor cabangnya !

**HALAMAN TERAKHIR MODUL**



**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS BUDI LUHUR**

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

http://fti.budiluhur.ac.id