**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 #10

**PENGOPERASIAN SQL (LANJUTAN)**

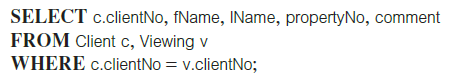
|  |  |  |
| --- | --- | --- |
| Capaian Pembelajaran | : | Mahasiswa mampu memahami pengoperasian SQL (Lanjutan) |
| Sub Pokok Bahasan | : | 1.1. Data Manipulation Language (multi tables query)  1.2. Order By  1.3. Sub Query  1.4. Aggregate Function  1.5. Group By  1.6. Union, Intersect |
| Daftar Pustaka | : | 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. **Data Manipulation Language (multi tables query)**

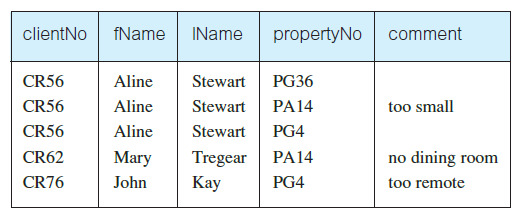
Terkadang data yang ingin kita tampilkan tidak selalu berasal dari satu tabel saja, bisa berasal dari dua tabel atau lebih. Untuk itu kita perlu menggunakan perintah join di mana pada klausa FROM dimasukkan beberapa tabel yang diinginkan dengan menggunakan tanda koma (,) sebagai pemisahnya.

Contoh :

Tampilkan nama dari seluruh klien yang telah melihat property beserta komentarnya



Maka akan dihasilkan data sebagai berikut :



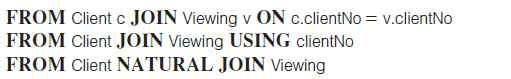
Huruf c dan v pada klausa FROM di atas hanya sebagai alias dari nama tabel saja sehingga pada klausa WHERE menjadi c.clientNo=v.clientNo.

Jika tidak menggunakan alias tabel c dan v, maka cukup gunakan

FROM Client, Viewing

saja, sehingga pada klausa WHERE nya menjadi Client.clientNo=Viewing.clientNo

Pada SQL Standard memberikan alternatif untuk proses join ini, yaitu :



* 1. **Order By**

Secara umum, baris atau record tabel yang dihasilkan dari proses query SQL tidak diurutkan berdasarkan urutan tertentu (meskipun beberapa DBMS dapat menggunakan urutan default, misalnya, berdasarkan primary key-nya). Namun, hasil query dapat diurutkan menggunakan klausa ORDER BY dalam perintah SELECT. Klausa ORDER BY terdiri dari sebuah kolom atau sekumpulan kolom yang digunakan sebagai acuan dalam proses pengurutan, dipisahkan oleh tanda koma.

Klausa ORDER BY memungkinkan hasil yang ditampilkan untuk diurutkan secara ascending (ASC) atau urutan descending (DESC) berdasarkan kolom atau kombinasi kolom yang diinginkan, terlepas apakah kolom tersebut muncul di hasilnya atau tidak

Contoh pengurutan yang menggunakan 1 kolom :

Tampilkan data staff berupa kode staf, nama dan gajinya, yang diurutkan berdasarkan gaji secara descending

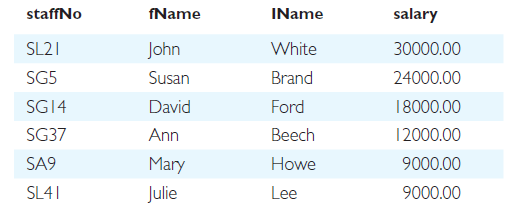
Maka perintah SQL nya adalah :

**SELECT** staffNo, fName, IName, salary

**FROM** Staff

**ORDER BY** salary **DESC**;

Hasil atau baris yang didapat adalah :



Pada hasil query di atas terlihat bahwa data staff ditampilkan dengan urut berdasarkan kolom salary secara DESCENDING atau menurun.

Data yang ditampilkan juga bisa diurutkan berdasarkan 2 kolom atau lebih. Untuk lebih jelasnya, mari lihat contoh berikut ini.

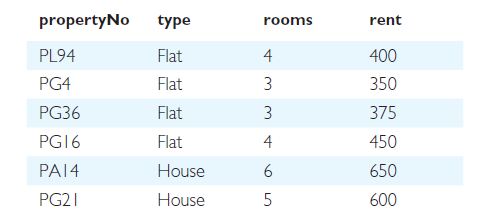
Jika perintah query nya adalah sebagai berikut :

**SELECT** propertyNo, type, rooms, rent

**FROM** PropertyForRent

**ORDER BY** type;

Maka akan ditampilkan data dari table **PropertyForRent** yang urut berdasarkan kolom type seperti data di bawah ini :



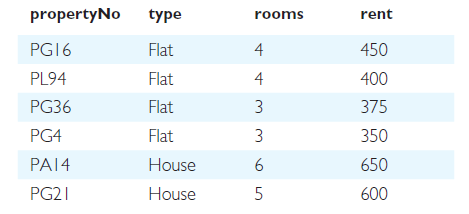
Namun jika dilihat dari biaya sewanya atau pada kolom Rent, maka data yang ditampilkan adalah tidak urut meski pun berada pada type property yang sama yaitu Flat. Jika data yang ditampilkan ingin diurutkan juga berdasarkan biaya sewanya, maka perintah SQL di atas harus diubah menjadi :

**SELECT** propertyNo, type, rooms, rent

**FROM** PropertyForRent

**ORDER BY** type, rent **DESC**;

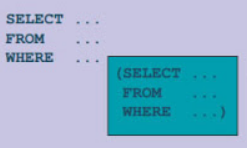
Sehingga akan didapatkan data yang lebih terurut seperti di bawah ini di mana data property bertipe Flat, biaya sewanya dapat terlihat urut secara menurun atau DESCENDING, demikian halnya dengan data property yang bertipe House.



* 1. **Sub Query**

Sub Query adalah nested query di mana terdapat query di dalam query. Biasanya sub query digunakan untuk menampilkan data yang cukup kompleks.

Bentuk umum sub query adalah :



Contoh :

Tampilkan data kantor cabang yang memiliki data property yang disewakan

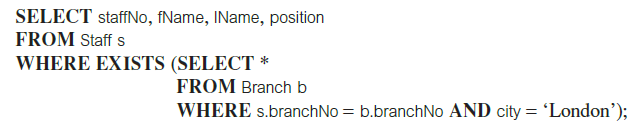
**SELECT** \*

**FROM** Branch

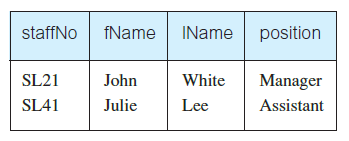
**WHERE** brancNo **IN** (**SELECT** branchNo **FROM** PropertyForRent)

Sub query di atas menggunakan keyword IN.

Sub query bisa juga menggunakan keyword EXISTS, seperti contoh berikut :



Contoh di atas adalah menampilkan data Staff yang bekerja di kantor cabang yang berada di London, sehingga akan menampilkan data sebagai berikut :



* 1. **Aggregate Function**

Selain menampilkan baris dan kolom dari basis data, terkadang data yang ingin ditampilkan adalah dalam bentuk ringkasan atau agregasi data, yang mirip seperti total akhir pada bagian bawah laporan. Standar ISO mendefinisikan ada lima fungsi agregat, yaitu :

* COUNT() – menghasilkan jumlah nilai yang terdapat pada kolom yang diinginkan
* SUM() – menghasilkan total dari nilai yang terdapat pada kolom yang diinginkan
* AVG() – menghasilkan rata-rata dari nilai yang terdapat pada kolom yang diinginkan
* MIN() – menghasilkan nilai yang terkecil pada kolom yang diinginkan
* MAX() – menghasilkan nilai yang terbesar pada kolom yang

Fungsi-fungsi ini beroperasi pada satu kolom tabel dan mengembalikan nilai yang tunggal. Fungsi COUNT, MIN, dan MAX berlaku untuk kolom yang bersifat numerik maupun yang non-numerik, tetapi SUM dan AVG hanya dapat digunakan pada kolom yang bersifat numerik.

Contoh penggunaan fungsi COUNT()

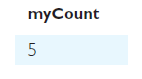
Tampilkan ada berapa banyak property yang biaya sewa per bulannya lebih dari £350

**SELECT COUNT**(\*) **AS** myCount

**FROM** PropertyForRent

**WHERE** rent > 350;

Maka akan didapatkan data :



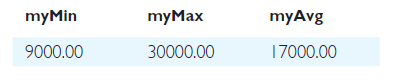
Contoh penggunaan fungsi COUNT()

Tampilkan data gaji terkecil, gaji terbesar dan rata-rata gaji dari staf yang ada

**SELECT MIN**(salary) **AS** myMin, **MAX**(salary) **AS** myMax, **AVG**(salary) **AS** myAvg

**FROM** Staff;

Maka akan didapatkan data :



* 1. **Group By**

Kueri ringkasan atau rangkuman sebelumnya (fungsi agregat) mirip dengan total di bagian bawah laporan. Seluruh data yang ada dipadatkan dalam 1 baris rangkuman, seperti jumlah seluruh record, nilai terkecil, nilai terbesar dan lain-lain.

Namun, sering kali yang diperlukan adalah subtotal dalam laporan yang dikelompokkan berdasarkan kriteria tertentu. Untuk melakukan ini bisa menggunakan klausa GROUP BY pada perintah SELECT. Kueri yang menyertakan klausa GROUP BY akan menghasilkan satu baris ringkasan tunggal untuk setiap grupnya.

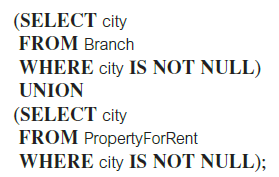
* 1. **Union, Intersect**

Kita bisa juga menggabungkan hasil dari dua atau lebih query ke dalam sebuah hasil query tunggal. Hal ini bisa kita lakukan dengan menggunakan perintah UNION dan INTERSECT.

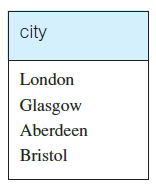
Untuk bisa melakukan UNION dan INTERSECT, maka relasi-relasi yang digunakan ini harus union-compatible, yaitu harus memiliki struktur yang sama.

Contoh penggunaan UNION :

Tampilkan seluruh kota di mana di kota tersebut ada kantor cabang atau ada property yang disewakan

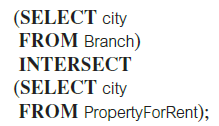


Proses UNION di atas akan menghasilkan informasi sebagai berikut :

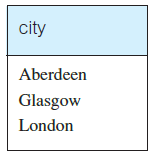


Contoh penggunaan INTERSECT :

Tampilkan seluruh kota di mana di kota tersebut ada kantor cabang dan ada property yang disewakan



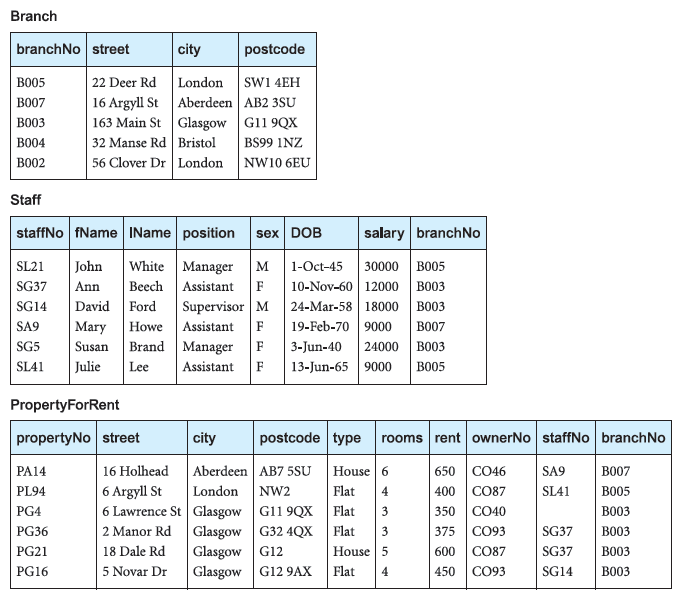
Proses INTERSECT di atas akan menghasilkan informasi sebagai berikut :

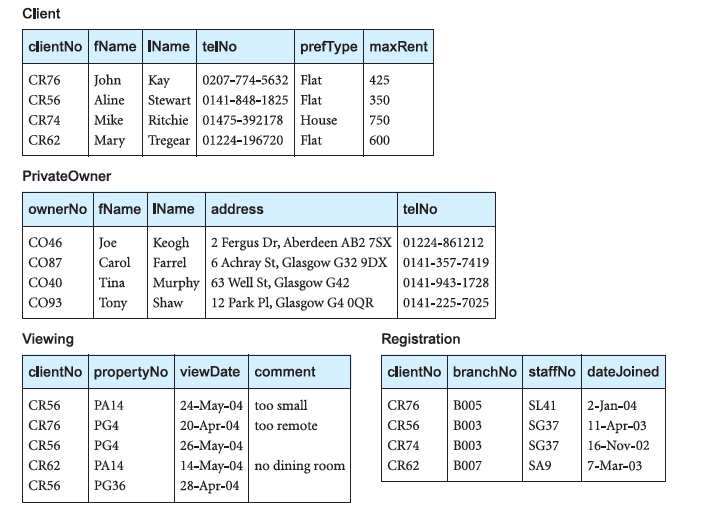


**RANGKUMAN**

* Multi-table queries dapat dilakukan dengan menggunakan Join
* Klausa Order By dapat digunakan untuk mengurutkan data secara ascending atau descending
* Sub query adalah nested query atau query di dalam query
* Fungsi agregat digunakan pada satu kolom dan menghasilkan satu baris atau tunggal. Contoh fungsi agregat adalah SUM(), MAX(). MIN(), AVG() dan COUNT()
* UNION dan INTERSECT adalah perintah yang digunakan untuk menggabung hasil dari dua atau lebih query ke dalam satu hasil query yang tunggal

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





**LATIHAN**

1. Buatlah perintah SQL untuk menampilkan data staf dan property yang dikelolanya !
2. Buatlah perintah SQL untuk menampilkan data klien yang melihat data property (Viewing) dan urut berdasarkan nama klien!
3. Buatlah perintah SQL untuk melihat data property yang sudah pernah dilihat oleh klien, tampilkan alamat dan jumlah kamar dari property tersebut !
4. Buatlah perintah SQL untuk menampilkan jumlah property yang disewakan berdasarkan kantor cabangnya !
5. Buatlah perintah SQL 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