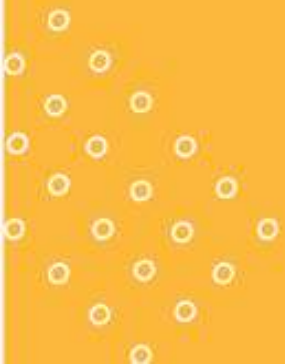
Bab 5:
SQL II



1. DISTINCT

   Perintah SELECT DISTINCT merupakan perintah dasar SQL yang digunakan untuk mengembalikan hanya nilai yang berbeda dari dalam sebuah tabel. Dengan kata lain, semua record duplikat (record bernilai sama) yang terdapat pada tabel akan dianggap sebagai satu record/nilai.

   Sintaks perintah SELECT DISTINCT adalah sebagai berikut:

   SELECT DISTINCT kolom1, kolom2, … FROM nama_tabel;

   Contoh:

   SELECT DISTINCT product FROM tb_product;

   Maka, output yang dikembalikan adalah tampilan semua record pada kolom 'product', dengan nilai yang sama akan ditampilkan hanya satu kali.

2. WHERE

   Perintah WHERE merupakan perintah dasar SQL yang digunakan untuk memberikan kondisi spesifik pada query. Apabila digabungkan dengan perintah SELECT, perintah ini berfungsi untuk memfilter/menyaring hasil yang dikembalikan oleh perintah SELECT.

   Sintaks perintah WHERE adalah sebagai berikut:

   SELECT kolom1, kolom2, … FROM nama_tabel WHERE kondisi;

   Contoh:

   ```
   SELECT no, nama
   FROM tb_user
   WHERE tanggal_lahir = "2021-01-01";
   ```

3. String Functions

PostgreSQL memiliki beberapa fungsi (function) yang dapat digunakan untuk memanipulasi data string. Pengelompokan fungsi-fungsi ini biasa disebut dengan String Functions. Manipulasi yang dimaksudkan adalah untuk mengubah atau memproses data string menjadi bentuk-bentuk yang diinginkan.

Di bawah ini adalah beberapa contoh fungsi string yang sering digunakan, antara lain:

a. LOWER

Fungsi LOWER dapat digunakan untuk mengubah string menjadi huruf kecil.

Sintaks fungsi ini adalah:

: LOWER(string)

Contoh:
: LOWER("Bapak Rahmat")

Output:

: "bapak rahmat"

b. UPPER

Fungsi UPPER pada SQL berfungsi untuk mengubah string menjadi berhuruf besar.

Sintaks fungsi ini adalah:

: UPPER(string)

Contoh:

: UPPER("Bapak Rahmat")

Output:

: "BAPAK RAHMAT"

c. LENGTH

Fungsi LENGTH dapat digunakan untuk menghitung panjang karakter dari string.

Sintaks fungsi ini adalah:

: LENGTH(string)

Contoh:

: LENGTH("ZIco")

Output:

: 4

d. CONCAT

Fungsi CONCAT berfungsi untuk menggabungkan/menyatukan satu string dengan string lainnya.

Sintaks fungsi ini adalah:

: CONCAT(string1,string2,string3,...)

Contoh:

: CONCAT("Bapak","-","Zico")

Output:

: "Bapak-Zico"

e. SUBSTRING

Fungsi SUBSTRING dapat digunakan untuk mengekstrak beberapa karakter string atau memotong karakter string sesuai dengan yang dispesifikkan.

Sintaks fungsi ini adalah:

: SUBSTRING(nama_kolom, index_awal, jumlah_karakter)

Contoh:

: SUBSTRING("Bapak dan Ibu",3 ,3)

Output:

: "pak"

4. Aggregate Functions

Aggregate Functions adalah kelompok fungsi-fungsi yang dapat melakukan kalkulasi pada sejumlah nilai data, dan kemudian mengembalikannya menjadi satu nilai tunggal.

Di bawah ini adalah beberapa fungsi agregasi pada SQL yang sering digunakan:

a. AVG

Fungsi AVG digunakan untuk menghitung rata-rata dari sebuah data atau record di dalam tabel.

Sintaks fungsi ini adalah:

```
SELECT AVG(nama_kolom)
FROM nama_tabel
[GROUP BY nama_kolom];
```

b. COUNT

Fungsi COUNT digunakan untuk menghitung banyaknya data pada suatu tabel atau kolom.

Sintaks fungsi ini adalah:

```
SELECT COUNT(nama_kolom)
FROM nama_tabel
[GROUP BY nama_kolom];
```

c. MAX

Fungsi Max digunakan untuk mencari dan mengembalikan nilai terbesar (maximum) dari sebuah data atau record di dalam tabel.

Sintaks fungsi ini adalah:

```
SELECT MAX(nama_kolom)
FROM nama_tabel
[GROUP BY nama_kolom];
```

d. MIN

Fungsi MIN digunakan untuk mencari dan mengembalikan nilai terkecil (minimum) dari sebuah data atau record di dalam tabel.

Sintaks fungsi ini adalah:

```
SELECT MIN(nama_kolom)
FROM nama_tabel
[GROUP BY nama_kolom];
```

e. SUM

Fungsi SUM digunakan untuk mencari jumlah total dari nilai sebuah data atau record di dalam tabel.

Sintaks fungsi ini adalah:

```
SELECT SUM(nama_kolom)
FROM nama_tabel
[GROUP BY nama_kolom];
```

5. GROUP BY

Klausa GROUP BY digunakan untuk melakukan pengelompokan baris di dalam tabel yang memiliki data yang identik. Klausa

GROUP BY ini mengikuti klausa FROM dan WHERE dalam pernyataan SELECT, dan mendahului klausa ORDER BY.

Secara umum, sintaks klausa GROUP BY adalah sebagai berikut:

```
SELECT daftar-kolom
FROM nama_tabel
GROUP BY kolom1, kolom2, ... kolomN
```

Query di atas akan menampilkan semua data yang sesuai dengan ekpresi 'daftar-kolom' dari tabel 'nama_tabel', dan mengelompokkan hasilnya ke dalam grup 'kolom1, kolom2, … kolomN'.

Klausa GROUP BY dapat digabungkan dengan fungsi-fungsi lain untuk menspesifikkan tujuan pengelompokan barisnya, misal untuk mengurangi redundansi dalam output.

Di bawah ini adalah beberapa contoh penggunaan klausa GROUP BY yang digabungkan dengan fungsi lain, antara lain:

- Penggunaan GROUP BY dengan WHERE

  Sintaks perintah ini adalah:

```
SELECT daftar-kolom
FROM nama_tabel
WHERE [kondisi ]
GROUP BY kolom1, kolom2, … kolomN
```

- Penggunaan GROUP BY dengan ORDER BY

  Sintaks perintah ini adalah:

```
SELECT daftar-kolom
```

```
FROM nama_tabel
GROUP BY kolom1, kolom2, ... kolomN
ORDER BY kolom1, kolom2, ... kolomN
```

- Penggunaan GROUP BY dengan WHERE dan ORDER BY

  Sintaks perintah ini adalah:

```
SELECT daftar-kolom
FROM nama_tabel
WHERE [kondisi ]
GROUP BY kolom1, kolom2, … kolomN
ORDER BY kolom1, kolom2, … kolomN
```

  Beberapa hal penting yang perlu Anda catat terkait dengan GROUP BY:

  - Klausa GROUP BY ini digunakan dengan pernyataan SELECT.
  - Di dalam sintaks, klausa GROUP BY diletakkan setelah klausa FROM dan WHERE.
  - Di dalam sintaks, klausa GROUP BY diletakkan sebelum klausa ORDER BY

6. JOIN TABLE

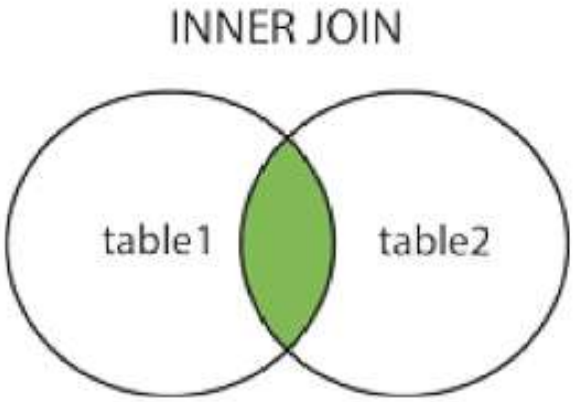
   Klausa JOIN pada SQL pada dasarnya digunakan untuk menggabungkan dan menampilkan data dari dua tabel atau lebih berdasarkan relasi logis di antara satu dan lainnya. Data-data dari dua tabel atau lebih yang saling berelasi atau memiliki hubungan tersebut nantinya akan ditampilkan menjadi satu tabel sementara.

   Ada tiga tipe perintah dasar untuk klausa JOIN ini, yaitu:

a. INNER JOIN TABLE

Perintah INNER JOIN digunakan untuk menggabungkan dua tabel atau lebih yang saling berelasi dengan menampilkan data-data yang sama/saling terhubung. Kesamaan hubungan tersebut didasarkan pada kepemilikan pasangan pada masing-masing tabel. Artinya apabila terdapat data yang terhubung tidak memiliki pasangan, maka data tersebut tidak akan ditampilkan.

Sederhananya, INNER JOIN TABLE dapat diilustrasikan dengan gambar di bawah ini. Irisan berwarna hijau adalah data-data yang akan ditampilkan melalui perintah INNER JOIN.



Sintaks perintah INNER JOIN pada SQL adalah sebagai berikut:

```
SELECT *
FROM   A  INNER JOIN B ON A.key = B.key
```

Contoh:

```
SELECT *
FROM   orders
       INNER JOIN order_details ON orders.order_id = order_details.
order_id
```

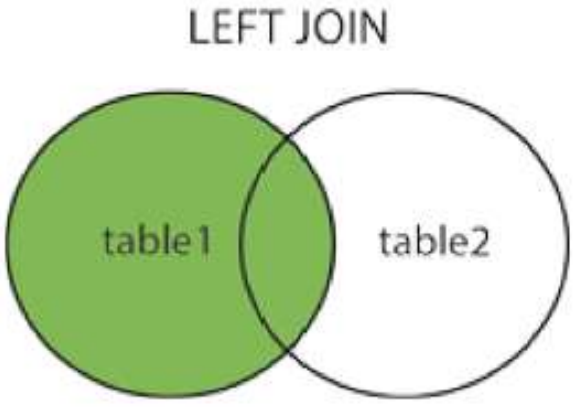
Output :

index	order_id	customer_id	order_date	deliver_date	index	order_id	product_id	quantity
0	0	1,006	2016-10-21	2016-10-31	0	0	8,048	9
1	1	1,050	2016-10-21	2016-10-29	1	1	8,110	5
1	1	1,050	2016-10-21	2016-10-29	2	1	8,051	9
1	1	1,050	2016-10-21	2016-10-29	3	1	8,032	4
1	1	1,050	2016-10-21	2016-10-29	4	1	8,091	4
2	2	1,152	2016-10-21	2016-10-27	5	2	8,014	9
3	3	1,167	2016-10-21	2016-10-27	6	3	8,030	8
4	4	1,191	2016-10-22	2016-10-27	7	4	8,015	3
4	4	1,191	2016-10-22	2016-10-27	8	4	8,077	2
4	4	1,191	2016-10-22	2016-10-27	9	4	8,141	3
5	5	1,086	2016-10-22	2016-10-29	10	5	8,069	9
5	5	1,086	2016-10-22	2016-10-29	11	5	8,095	5
5	5	1,086	2016-10-22	2016-10-29	12	5	8,025	3
5	5	1,086	2016-10-22	2016-10-29	13	5	8,092	5
5	5	1,086	2016-10-22	2016-10-29	14	5	8,146	2

b. LEFT JOIN TABLE

Perintah LEFT JOIN digunakan untuk menggabungkan yang saling berelasi dengan menampilkan semua data pada tabel sebelah kiri. Sementara data yang tidak memiliki pasangan atau terhubung akan diberi nilai NULL.

LEFT JOIN TABLE dapat diilustrasikan dengan gambar di bawah ini. Perintah LEFT JOIN ini akan menampilkan semua data pada bagian berwarna hijau.



Sintaks perintah LEFT JOIN pada SQL adalah sebagai berikut:

```
SELECT *
```

```
FROM    A  LEFT JOIN B ON A.key = B.key
```

Contoh:

```
SELECT *
FROM    order_details
        LEFT JOIN products ON order_details.product_id = products.
product_id
```

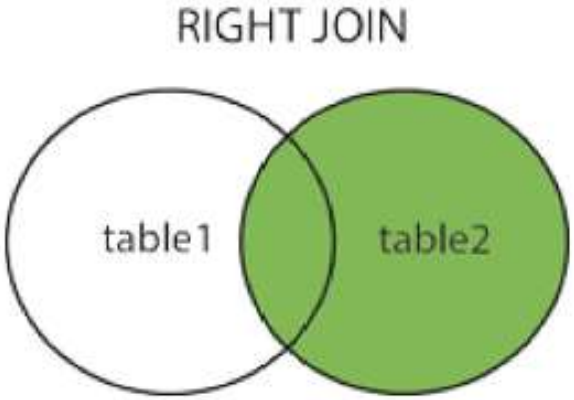
Output :

index	order_id	product_id	quantity	index	product_id	product_name	price
28	13	8,000	5	0	8,000	Small Soft Chips	88
127	46	8,000	5	0	8,000	Small Soft Chips	88
185	67	8,000	2	0	8,000	Small Soft Chips	88
704	239	8,000	3	0	8,000	Small Soft Chips	88
709	240	8,000	5	0	8,000	Small Soft Chips	88
734	246	8,000	6	0	8,000	Small Soft Chips	88
1,052	358	8,000	4	0	8,000	Small Soft Chips	88
1,246	426	8,000	10	0	8,000	Small Soft Chips	88
1,391	470	8,000	3	0	8,000	Small Soft Chips	88
1,542	518	8,000	6	0	8,000	Small Soft Chips	88
1,583	532	8,000	10	0	8,000	Small Soft Chips	88
1,614	541	8,000	2	0	8,000	Small Soft Chips	88
1,635	545	8,000	10	0	8,000	Small Soft Chips	88
1,701	575	8,000	4	0	8,000	Small Soft Chips	88

c. RIGHT JOIN TABLE

Perintah RIGHT JOIN digunakan untuk menggabungkan dua tabel atau lebih yang saling berelasi dengan menampilkan semua data pada tabel sebelah kanan. Sementara data yang tidak memiliki pasangan atau terhubung akan diberi nilai NULL.

RIGHT JOIN TABLE dapat diilustrasikan dengan gambar di bawah ini. Perintah RIGHT JOIN ini akan menampilkan semua data pada bagian berwarna hijau.



Sintaks perintah RIGHT JOIN pada SQL adalah sebagai berikut:

```
SELECT *
FROM   A  RIGHT JOIN B ON A.key = B.key
```

Contoh:

```
SELECT *
FROM   orders
       RIGHT JOIN customers ON orders.customer_id = customers.
user_id
```

Output :

index	order_id	customer_id	order	deliver_date	index	user_id	name	email	address	state
0	0	1,006	2016-10-	2016-10-31	6	1,006	Melissa Soto	dmoore@hotmai	78887 Allen Forge	SC
1	1	1,050	2016-10-	2016-10-29	50	1,050	Lynn Henderson	owensmichael@a	79733 Hoffman Skywa	AR
2	2	1,152	2016-10-	2016-10-27	152	1,152	Bonnie Hall	michelle54@torre	648 John Falls	MA
3	3	1,167	2016-10-	2016-10-27	167	1,167	Jennifer Chavez	brownmary@perl	907 Johnson Parkways	ND
4	4	1,191	2016-10-	2016-10-27	191	1,191	Kristin Davis	katherinecook@g	32496 Cathy Stravenue	VA
5	5	1,086	2016-10-	2016-10-29	86	1,086	Jessica Miller	ericareese@mcint	9470 Joe Village Suite 1	ID
6	6	1,141	2016-10-	2016-10-27	141	1,141	Tanya Carson	alyssa13@yahoo.	573 Colin Well	IL
7	7	1,150	2016-10-	2016-10-31	150	1,150	Elaine Valdez	jgonzalez@brower	85142 Ford Pass	AL
8	8	1,109	2016-10-	2016-10-29	109	1,109	William Edwards	dhernandez@har	93395 Carlos Ways Sui	RI
9	9	1,144	2016-10-	2016-11-01	144	1,144	Helen Price	halltammy@gma	4596 Madison Gardens	MA
10	10	1,009	2016-10-	2016-10-24	9	1,009	Michael Flores	twhite@yahoo.co	262 Oneal Flats	AZ

7. Subqueries

Subqueries, pada dasarnya, adalah sebuah query di dalam query.

Subqueries biasa juga disebut sebagai Inner Query atau Nested Query.

Sebuah subqueries digunakan di dalam query utama sebagai syarat untuk lebih membatasi/menspesifikkan data yang akan ditampilkan. Penulisan subqueries selalu di dalam tanda kurung.

Di bawah ini adalah beberapa contoh penggunaan subqueries, yaitu:

a. Subquery pada Filter WHERE

Hasil data dari perintah SQL yang berhasil dijalankan dapat dipergunakan kembali untuk diproses oleh perintah SQL yang lainnya dengan menggunakan subquery. Misal, untuk mencari tanggal kapan saja yang memiliki total transaksi di atas rata-rata total transaksi. Sintaks perintah SQL untuk menjawabnya adalah sebagai berikut:

```
SELECT      hari_transaski,
            amount_transaksi
FROM        transaksi WHERE amount_transaksi > (    SELECT
            AVG(amount_transaksi)
                                                    FROM transaksi )
```

b. Subquery pada Ditabel

Subquery tidak hanya dapat dilakukan pada filter, tetapi dapat juga dilakukan pada hasil query yang sudah ada. Seperti di bawah ini:

```
SELECT      MAX(avg_amount_transaksi) AS max_avg_amount_transaksi
FROM (      SELECT  hari_transaski,
            AVG(amount_transksi) AS avg_amount_transaksi
            FROM transaksi
            GROUP BY hari_transaksi);
```