|  |  |
| --- | --- |
|  | Jurusan Teknologi Informasi Politeknik Negeri Malang  **Jobsheet-7: MySQL – Data Retrieval Language (DRL)**  **Mata Kuliah Basis Data**  Pengampu: Tim Ajar Basis Data  *Marer 2019* |

**Topik**

Data Retrieval Language (DRL) pada DBMS MySQL

**Tujuan**

Mahasiswa diharapkan dapat memahami penggunaan dan dapat membuat SQL statement SELECT dengan berbagai klausa.

**Pendahuluan**

DRL atau DSL (Data Selection Language) adalah set perintah yang dgunakan untuk mengambil data dari server basis data. DRL terdiri dari klausa yang dapat dikembangkan lebih lanjut, antara lain:

1. **FROM** : Memilih nama sebuah tabel
2. **WHERE** : Menunjukkan secara spesifik suatu row/baris yang akan dicari
3. **GROUP BY** : Menyusung data ke dalam grup
4. **HAVING :** Memilih diantara kelompok-kelompok data yang didefinisikan klausa GROUP BY
5. **ORDER BY** :Secara spesifik digunakan untuk menentukan suatu baris pada kolom tertentu
6. **AS** :Memberikan nama alias sementara untuk tabel atau kolom

DRL lebih diterapkan dalam beberapa hal dalam beberapa hal seperti :

1. **QUERY**
2. **SUB-QUERY**
3. **FUNGSI AGREGASI**

**QUERY**

Query merupakan operasi yang melibatkan satu atau lebih tabel untuk melakukan retrival data. Pengambilan data dilakukan dengan beberapa klausa berikut :

1. SELECT \* untuk memilih semua kolom

Format

SELECT \* FROM table\_name;

1. SELECT dengan WHERE untuk menampilkan baris dengan suatu kondisi

Format

SELECT column1,column2,...  
FROM table\_name  
WHERE condition;

1. SELECT dengan DISTINCT untuk menampilkan data dengan eliminasi data yang sama (duplicate)

Format

SELECT DISTINCT *column* FROM *table\_name*;

1. SELECT dengan IN untuk menampilkan data yang spesifik

Format

SELECT column\_name(s)  
FROM table\_name  
WHERE column\_name IN (value1,value2, ...);

1. SELECT dengan BETWEEN untuk menampilkan data pada jarak (range) tertentu

Format

SELECT column\_name(s)  
FROM table\_name  
WHERE column\_nameBETWEEN value1 AND value2;

1. SELECT dengan LIKE untuk menampilkan data yang memiliki kemiripan dengan keyword yang diinginkan

Format

SELECT column1,column2,...  
FROM table\_name  
WHERE columnN LIKE pattern;

1. SELECT dengan GROUP BY untuk menampilkan susunan data dalam bentuk grup

Format

SELECT column1,column2,...

*condition*  
FROM table\_name  
GROUP BY column1, column2, ...;

1. SELECT dengan ORDER BY untuk menampilkan baris secara spesifik dan terurut maju atau mundur

Format

SELECT column1,column2, ...  
FROM table\_name  
ORDER BY column1, column2, ...ASC|DESC;

1. SELECT dengan AND, OR and NOT untuk menampilkan data dengan kondisi dan atau atau tidak

Format

AND

SELECT column1,column2, ...  
FROM table\_name  
WHERE condition1 AND condition2 AND condition3 ...;

OR

SELECT column1,column2, ...  
FROM table\_name  
WHERE condition1 OR condition2 OR condition3 ...;

NOT

SELECT column1,column2, ...  
FROM table\_name  
WHERE NOT condition;

1. SELECT dengan UNION, INTERSECT dan EXCEPT untuk menampikan data dengan operasi himpunan yang melibatkan lebih dari satu tabel

Format

UNION

*Tanpa duplikasi*

SELECT column\_name(s) FROM table1  
UNION  
SELECT column\_name(s) FROM table2;

*Dengan duplikasi*

SELECT column\_name(s) FROM table1  
UNION ALL  
SELECT column\_name(s) FROM table2;

INTERSECT

*Tanpa duplikasi*

SELECT column\_name(s) FROM table1  
INTERSECT  
SELECT column\_name(s) FROM table2;

*Dengan duplikasi*

SELECT column\_name(s) FROM table1  
INTERSECT ALL  
SELECT column\_name(s) FROM table2;

EXCEPT

*Tanpa duplikasi*

SELECT column\_name(s) FROM table1  
EXCEPT  
SELECT column\_name(s) FROM table2;

*Dengan duplikasi*

SELECT column\_name(s) FROM table1  
EXCEPT ALL  
SELECT column\_name(s) FROM table2;

**SUB-QUERY**

Sub-query adalah adanya query di dalam query lain. Sub-query juga disebut dengan perintah SELECT bersarang (nested SELECT). Retrival data dalam sub-query dilakukan oleh SELECT untuk menemukan data pada SELECT utama. Klausa GROUP BY dan HAVING juga dapat digunakan untuk retrival data pasa sub-query. Sub-query biasanya digunakan untuk test keanggotaan himpunan atau perbandingan himpunan dan kardinalitas.

Format

*Untuk test keanggotaan*

SELECT column\_name(s)  
FROM table\_name  
WHERE column\_name IN (SELECTSTATEMENT);

*atau*

SELECT column\_name(s)  
FROM table\_name

GROUP BY column1, column2, ...;

HAVING column\_name IN (SELECTSTATEMENT);

*Untuk perbandingan himpunan*

SELECT column\_name(s)  
FROM table\_name

operator  
(SELECTSTATEMENT);

**FUNGSI AGREGASI**

Fungsi agregasi akan melakukan perhitungan kepada sekumpulan nilai dan menghasilkan suatu nilai tunggal.

Jenis:

1. **AVG** untuk menghitung rata-rata

Format

SELECT AVG(column\_name)  
FROM table\_name  
WHERE condition;

1. **MAX** untuk mencari nilai tertinggi

Format

SELECT MAX(column\_name)  
FROM table\_name  
WHERE condition;

1. **MIN** untuk mencari nilai terendah

Format

SELECT MIN(column\_name)  
FROM table\_name  
WHERE condition;

1. **SUM** untuk menjumlahkan sekumpulan nilai

Format

SELECT SUM(column\_name)  
FROM table\_name  
WHERE condition;

1. **COUNT()** untuk menunjukkan jumlah baris pada kolom yang diinginkan

Format

SELECT COUNT(column\_name)  
FROM table\_name  
WHERE condition;

**Praktikum – Bagian 1: Membuat Database untuk Percobaan**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Studi kasus yang digunakan sama dengan jobsheet DML dengan skema/model relasional/EER diagram dari database berikut. |
|  | ../../../../../../Desktop/Jepretan%20Layar%202017-03-25%20pada%2010.11.3 |
| 2 | Skema tersebut adalah skema database pada sebuah sistem informasi yang digunakan oleh suatu perusahaan yang bergerak di bidang pembuatan perangkat lunak. Sesuai dengan percobaan pada jobsheet DML. Insert kembali departemen yang mengandung nama ‘dan’, dengan cara menyalin dari tabel departemen\_backup, sehingga menghasilkan record keseluruhan tabel sebagai berikut (record tidak harus sama persis dengan tampilan ini): |
|  | SELECT \* FROM departemen; |
|  | SELECT \* FROM karyawan; |
|  | SELECT \* FROM klien; |
|  | SELECT \* FROM proyek; |
|  | SELECT \* FROM penugasan; |
| 3 | Setelah selesai membuat database diatas, lanjutkan ke **Praktikum – Bagian 2**. |

**Praktikum – Bagian 2: Percobaan SELECT QUERY**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Untuk menampilkan baris dengan suatu kondisi tertentu, digunakan statement **SELECT** dengan **WHERE**. Eksekusi SQL berikut untuk menampilkan (record) pada tabel **karyawan** yang memiliki kolom departemen\_kode=’MKT’. |
|  | SELECT nama  FROM karyawan  WHERE departemen\_kode = 'MKT'; |
|  |  |
| 2 | Untuk menampilkan data dengan eliminasi data yang sama (duplicate), digunakan statement **SELECT** dengan **DISTINCT**. Eksekusi SQL berikut untuk menampilkan kolom total\_jam dari tabel **penugasan** yang bernilai tidak sama. |
|  | SELECT DISTINCT total\_jam  FROM penugasan; |
|  |  |
| 3 | Untuk menampilkan data yang spesifik digunakan statement **SELECT** dengan **IN**. Eksekusi SQL berikut untuk menampilkan kolom nim dan nik dari tabel **karyawan** yang memiliki gaji 3100000 atau 6900000. |
|  | SELECT nik, nama  FROM karyawan  WHERE gaji IN (3100000,6900000); |
|  |  |
| 4 | Untuk menampilkan data pada jarak (range) tertentu digunakan statement **SELECT** dengan **BETWEEN**. Eksekusi SQL berikut untuk menampilkan kolom nik dan nama dari tabel **karyawan** yang memiliki gaji diantara 2000000 dan 4000000. |
|  | SELECT nik, nama  FROM karyawan  WHERE gaji  BETWEEN 2000000 AND 4000000; |
|  |  |
| 5 | Untuk menampilkan data yang memiliki kemiripan dengan keyword yang diinginkan digunakan **SELECT** dengan **LIKE**. Eksekusi SQL berikut untuk menampilkan kolom nama, alamat, dan jenis pada tabel **klien** yang memiliki nama dengan huruf awal ‘a’. |
|  | SELECT nama, alamat, jenis FROM klien WHERE nama like 'a%'; |
|  |  |
| 6 | Untuk menampilkan susunan data dalam bentuk grup, digunakan **SELECT** dengan **GROUP BY**. Eksekusi SQL berikut untuk menampilkan kolom nik, nama, gaji, dan departemen\_kode pada tabel **karyawan** yang dikelompokkan berdasarkan gaji. |
|  | SELECT nik, nama, gaji, departemen\_kode  FROM karyawan  GROUP BY gaji; |
|  |  |
| 7 | Untuk menampilkan baris secara spesifik dan terurut maju atau mundur, digunakan **SELECT** dengan **ORDER BY**. Eksekusi SQL berikut untuk menampilkan kolom nama dan tanggal\_mulai pada tabel **proyek** dari tanggal yang paling lama hingga baru. |
|  | SELECT nama, tanggal\_mulai  FROM proyek  ORDER BY tanggal\_mulai ASC; |
|  |  |
| 8 | Untuk menampilkan data dengan kondisi dan atau atau tidak, digunakan **SELECT** dengan **AND**, **OR** and **NOT**. Eksekusi SQL berikut untuk menampilkan semua kolom pada tabel **karyawan** dengan jenis kelamin ‘WANITA’ dan gaji < 5000000. |
|  | SELECT \*  FROM karyawan  WHERE jenis\_kelamin = 'WANITA' AND gaji < 5000000; |
|  |  |
| 9 | Untuk menampilkan data dari kolom yang terlibat dalam dua tabel dapat digunakan **SELECT** dengan **UNION**. **UNION** secara otomatis akan menghilangkan duplikasi. Eksekusi SQL berikut untuk menampilkan kolom nik yang yang ada di tabel **penugasan** atau **departemen**. |
|  | SELECT nik  FROM penugasan  UNION ALL  SELECT nik\_manajer FROM departemen; |
|  |  |
| 10 | Untuk menampilkan data dari kolom yang terlibat dalam dua tabel dapat digunakan **SELECT** dengan **UNION ALL**. **UNION** ALL juga akan menampilkan duplikasi data. Eksekusi SQL berikut untuk menampilkan kolom nik yang yang ada di tabel **penugasan** atau **departemen**. |
|  | SELECT nik  FROM penugasan  UNION ALL  SELECT nik\_manajer FROM departemen; |
|  |  |
| 11 | Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke **Praktikum - Bagian 3**. |

**Praktikum - Bagian 3: Percobaan SELECT Sub-Query**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Untuk menampilkan data (test keanggotaan sub-query) yang berasal dari pemilihan tampilan data lain digunakan tambahan statement **IN**. Eksekusi SQL berikut untuk menampilkan kolom nomor\_proyek, nik, total\_jam pada tabel **penugasan** dimana total\_jam adalah total\_jam yang paling lama pada tabel **penugasan**. |
|  | SELECT nomor\_proyek, nik, total\_jam  FROM penugasan  WHERE total\_jam  in (SELECT MAX(total\_jam) FROM penugasan); |
|  |  |
| 2 | Untuk menampilkan data (perbandingan himpunan sub-query) yang berasal dari pemilihan tampilan data lain digunakan tambahan operator. Eksekusi SQL berikut untuk menampilkan kolom nomor\_proyek, nik, total\_jam pada tabel **penugasan** berdasarkan semua record kolom total\_jam harus bernilai kurang dari total\_jam yang dimiliki oleh nomor proyek 1 dari tabel **penugasan**. |
|  | SELECT nomor\_proyek, nik, total\_jam  FROM penugasan  WHERE total\_jam < ALL (SELECT total\_jam FROM penugasan WHERE nomor\_proyek = 1); |
|  |  |
| 3 | Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke **Praktikum - Bagian 4**. |

**Praktikum - Bagian 4: Percobaan FUNGSI AGREGASI**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Untuk menghitung rata-rata, digunakan tambahan statement **AVG.** Eksekusi SQL berikut untuk menampilkan rata-rata gaji sesuai kolom gaji dari tabel **karyawan**. |
|  | SELECT AVG(gaji) FROM karyawan; |
|  |  |
| 2 | Untuk menghitung nilai tertinggi, digunakan tambahan statement **MAX.** Eksekusi SQL berikut untuk menampilkan gaji tertinggi sesuai kolom gaji dari tabel **karyawan**. |
|  | SELECT MAX(gaji) FROM karyawan; |
|  |  |
| 3 | Untuk menghitung nilai terendah, digunakan tambahan statement **MIN.** Eksekusi SQL berikut untuk menampilkan gaji terendah sesuai kolom gaji dari tabel **karyawan**. |
|  | SELECT MIN(gaji) FROM karyawan; |
|  |  |
| 4 | Untuk menghitung total gaji keseluruhan digunakan tambahan statement **SUM.** Eksekusi SQL berikut untuk menampilkan total gaji sesuai kolom gaji dari tabel **karyawan**. |
|  | SELECT SUM(gaji) FROM karyawan |
|  |  |
| 5 | Untuk menunjukkan jumlah baris pada kolom yang diinginkan digunakan COUNT(). Eksekusi SQL berikut untuk menampilkan jumlah baris pada kolom dari tabel **karyawan**. |
|  | SELECT COUNT(gaji) FROM karyawan; |
|  |  |
| 6 | Lanjutkan ke bagian **Tugas**! |

**Tugas**

1. Himpunlah *screenshot* sintaksis SQL berikut hasil SELECT yang Anda dapat di komputer Anda pada bagian **Praktikum** dalam sebuah laporan dengan format **PDF**!
2. Bagaimana sintaksis untuk melakukan pencarian tanggal selesai proyek yang paling akhir dengan memberikan nama alias ‘baru’, dan juga pencarian tanggal selesai proyek paling awal dengan memberikan nama alias kolom ‘lama’?Tunjukkan outputnya!
3. Lakukan pencarian jumlah karyawan yang bekerja pada kode\_departemen ‘CSR’ kemudian tunjukkan sintaksis dan outputnya!
4. Carilah data yang menunjukkan semua nik dari seorang manajer yang berperan dalam penugasan. Gunakan operasi intersect! Tunjukkan sintaksis dan output pencarian tersebut!
5. Tunjukkan total nik, nomor\_proyek dan total\_jam dari tabel penugasan yang dikelompokkan berdasarkan nomor proyek yang sama berjumlah kurang dari dua. Kelompok tersebut diurutkan sesuai nomor\_proyek secara terurut dari atas ke bawah. Sertakan sintaksis dan output dari sintaksis SELECT tersebut.

***-- Selamat Mengerjakan –***

**Daftar Pustaka**

* Dwi Puspitasari, S.Kom, “**Buku Ajar Dasar Basis Data**”, *Program Studi Manajemen Informatika Politeknik Negeri Malang*, 2012.
* <http://www.tutorialride.com>
* <https://www.w3schools.com/>