

Laporan Pengerjaan

Kelas PPL PeTIK II Jombang | Materi Database SQL Pertemuan 16

Hanif | Ilham | Mu’aadz | Fandy

22255601010 | 22255601012 | 22255601001 | 22255601016

Jumat, 14 Oktober 2022

# Daftar Isi

[**Daftar Isi**](#_gg3op6vfw19s) **1**

[**Fungsi Aggregate**](#_9qoxqeduybtc) **2**

[**Subquery**](#_7o0wxjer8teo) **2**

[**SQL Join Table**](#_i4w0tf1rbbv9) **2**

[**View MySQL**](#_gna1dibtgjiz) **2**

[**Stored Procedure**](#_dilldyvymr68) **3**

[**Function**](#_185dnrptybxb) **3**

[**Trigger**](#_46700fpy6936) **3**

[**Transaction**](#_725wa6rwlh9f) **4**

[**Manajemen User**](#_ibaxlt63vae4) **5**

[**Backup & Restore**](#_5n5d5kfmktp2) **5**

[**Referensi**](#_q30r1bd1p3yd) **6**

# Fungsi Aggregate

Fungsi yang menerima koleksi nilai dan mengembalikan nilai tunggal data sebagai hasilnya.

Contoh fungsi aggregate :

* **COUNT**, untuk menghitung jumlah data
* **SUM**, untuk menghitung total nilai suatu kolom
* **AVG**, untuk menghitung rata-rata suatu kolom
* **MAX**, untuk mencari nilai terbesar
* **MIN**, untuk mencari nilai minimum suatu kolom

# Subquery

Sebuah subquery digunakan untuk mengembalikan data yang akan digunakan dalam query utama sebagai syarat untuk lebih membatasi data yang akan diambil. Subqueries dapat digunakan dengan **SELECT, INSERT, UPDATE,** dan **DELETE** statements bersama dengan operator seperti =, <,>,> =, <=, **IN, BETWEEN** dll

# SQL Join Table

**JOIN** merupakan salah satu fungsi yang ada di SQL yang digunakan untuk penggabungan table melalui kolom atau key tertentu dimana memiliki nilai terkait untuk mendapatkan satu set data dengan informasi lengkap

# View MySQL

Tujuan dari pembuatan **VIEW** adalah untuk kenyamanan (mempermudah penulisan query), untuk keamanan (menyembunyikan beberapa kolom yang bersifat rahasia), atau dalam beberapa kasus bisa digunakan untuk mempercepat proses menampilkan data (terutama jika kita akan menjalankan query tersebut secara berulang)

# Stored Procedure

Fungsi ***Stored Procedure*** adalah untuk menggantikan berbagai kumpulan perintah seperti perintah **CRUD, Sorting Table, SELECT**

# Function

Fungsi (***function***) atau kata lainnya *method* merupakan suatu sub program yang diperuntukan untuk mengerjakan suatu perintah tertentu sesuai dengan fungsi method itu sendiri

# Trigger

Trigger dalam database adalah kode prosedural yang secara otomatis dijalankan untuk menanggapi perubahan tertentu pada table tertentu atau tampilan dalam database.

Trigger dapat didefinisikan untuk menjalankan perintah sebelum atau setelah eksekusi **DML** (**Data Manipulation Language**) seperti **INSERT**, **UPDATE**, dan **DELETE**. Trigger banyak digunakan untuk menjaga integritas informasi pada database.

Trigger didukung oleh banyak RDBMS, seperti : MySQL, PostgreSQL, SQLite, Firebird, DB2, Microsoft SQL Server dan Oracle. Implementasinya bervariasi sehingga Anda harus memeriksa dokumentasi database yang sesuai sebelum coding.

# 

# Transaction

Transactions adalah sekelompok operasi sekuensial untuk memanipulasi data dalam database, yang dilakukan seolah-olah merupakan satu unit kerja tunggal.

Dengan kata lain, transactions akan berhasil jika setiap operasi individu dalam grup juga berhasil. Jika salah satu operasi dalam transactions gagal, seluruh transactions akan gagal.

Sebuah transactions dimulai dengan pernyataan SQL executable. Sebuah transactions berakhir saat commit atau rolled back, baik secara eksplisit dengan pernyataan COMMIT atau ROLLBACK atau secara implisit ketika **DDL** (**Data Definition Language**) digunakan untuk mengelola struktur tabel dan indeks dengan pernyataan **CREATE**, **ALTER**, **RENAME**, **DROP** dan **TRUNCATE**.

Transactions memiliki empat sifat standar berikut, biasanya disebut dengan akronim ACID :

* **Atomicity** : memastikan bahwa semua operasi di dalam unit kerja selesai dengan sukses. Jika tidak, maka transaction akan dibatalkan pada pada titik kegagalan kemudian operasi sebelumnya dikembalikan ke keadaan semula.
* **Consistency** : memastikan bahwa database benar-benar mengubah keadaan pada transactions yang berhasil dilakukan.
* **Isolation** : memungkinkan transactions beroperasi secara independen dan transparan satu sama lain.
* **Durability** : memastikan bahwa hasil atau akibat dari transactions yang dilakukan tetap dijalankan meski terjadi kegagalan sistem.

# Manajemen User

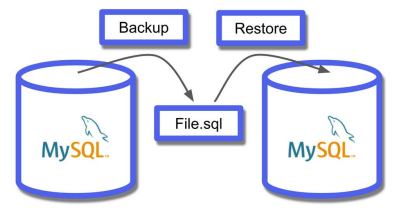
Begitu MySQL digunakan, Anda akan langsung diberikan username dan password. Dengan adanya informasi login ini, Anda akan memperoleh akses root atau kendali penuh atas semua database dan tabel.

Seiring dengan berjalannya waktu dan proyek online yang makin berkembang, pastinya Anda berencana untuk memberikan akses database ke user lain. Namun, dalam hal ini Anda tidak ingin mengizinkan user tersebut memegang kendali penuh.

Misalnya, Anda meminta bantuan developer untuk mengelola database. Akan tetapi, Anda tidak ingin mereka punya hak untuk menghapus atau memodifikasi informasi yang bersifat sensitif.

Maka dari itu, tipe informasi login yang sebaiknya diberikan adalah yang bersifat non-root user. Nantinya Anda bisa melacak apa saja yang boleh dan tidak boleh dilakukan oleh developer terhadap data Anda.

# Backup & Restore

Backup dan Restore database adalah praktik untuk membuat salinan berkala data pada sistem database ke perangkat sekunder yang terpisah dan kemudian menggunakan salinan tersebut untuk memulihkan data ke sebuah sistem database lain.

# Referensi

* Materi Database SQL PeTIK
* <https://sis.binus.ac.id/2018/09/10/sql-subquery>
* <https://www.duniailkom.com/tutorial-belajar-mysql-pengertian-view-dan-cara-penggunaan-view-dalam-mysql>
* <https://dqlab.id/mengenal-macam-macam-fungsi-join-table-sql-dan-perbedaannya>
* <https://www.dignitas.id/blog/mengenaltrigger>
* <https://kelasprogrammer.com/belajar-penggunaan-function-di-mysql/>
* <https://tutorallprogramming.blogspot.com/2017/11/stored-procedure-trigger-dan-function.html>
* <https://www.yudana.id/berkenalan-dengan-transactions-dalam-database-mysql>
* <https://www.hostinger.co.id/tutorial/cara-membuat-hak-akses-user-di-mysql>