

soal 1

Query

```
MariaDB [sekolah_fatsa]> create table tabel_guru (  
-> id_guru int(2) not null primary key,  
-> nama_depan varchar(50) not null unique,  
-> nama_belakang varchar(50) not null,  
-> mapel varchar(50) not null,  
-> jabatan varchar(50) not null,  
-> usia int(2) not null,  
-> tanggal_lahir date not null);  
Query OK, 0 rows affected (0.026 sec)
```

Analisis

CREATE TABLE tabel_guru: Ini adalah pernyataan SQL untuk membuat sebuah tabel baru dengan nama "tabel_guru".

id_guru INT(2) NOT NULL PRIMARY KEY: Kolom "id_guru" adalah bilangan bulat (integer) dengan panjang maksimal 2 digit, tidak boleh bernilai NULL, dan bertindak sebagai primary key untuk tabel ini.

nama_depan VARCHAR(50) NOT NULL UNIQUE: Kolom "nama_depan" adalah string dengan panjang maksimal 50 karakter, tidak boleh bernilai NULL, dan harus bernilai unik (tidak boleh ada duplikat).

nama_belakang VARCHAR(50) NOT NULL: Kolom "nama_belakang" adalah string dengan panjang maksimal 50 karakter, tidak boleh bernilai NULL.

mapel VARCHAR(50) NOT NULL: Kolom "mapel" adalah string dengan panjang maksimal 50 karakter, tidak boleh bernilai NULL.

jabatan VARCHAR(50) NOT NULL: Kolom "jabatan" adalah string dengan panjang maksimal 50 karakter, tidak boleh bernilai NULL.

usia INT(2) NOT NULL: Kolom "usia" adalah bilangan bulat (integer) dengan panjang maksimal 2 digit, tidak boleh bernilai NULL.

tanggal_lahir DATE NOT NULL: Kolom "tanggal_lahir" adalah tipe data tanggal, tidak boleh bernilai NULL.

Kesimpulan

Tabel **"tabel_guru"** ini dibuat untuk menyimpan data-data guru, termasuk informasi tentang nama depan, nama belakang, mata pelajaran yang diajar, jabatan, usia, dan tanggal lahir. Dengan menggunakan tipe data dan batasan yang tepat, tabel ini dapat membantu menjaga integritas dan konsistensi data guru di dalam sistem.

soal 2

Query

```
MariaDB [sekolah_fatsa]> INSERT INTO tabel_guru
-> VALUES
-> (1, "Adrianty", "", "Pemrograman web", "Ketua Jurusan", 34, '1982-06-29');
-> (2, "Ibrahim", "Hilomawang", "Basis Data", "Kepala Sekolah", 21, '2000-09-21');
-> (3, "Muhammad", "Yusuf", "Pemodelan perangkat lunak", "", 25, '1992-12-20');
-> (4, "Rusdiansyar", "", "Pemrograman berorientasi objek", "Asisten Kepala sekolah", 25, '1996-01-21');
Query OK, 4 rows affected (0.004 sec)
```

Analisis

INSERT INTO tabel_guru: Pernyataan SQL ini digunakan untuk menambahkan data baru ke dalam tabel "tabel_guru".

VALUES: Kata kunci ini digunakan untuk menentukan nilai-nilai yang akan dimasukkan ke dalam tabel.

Setiap baris di dalam tabel **VALUES** mewakili satu data guru baru yang akan dimasukkan, dengan urutan sesuai dengan definisi kolom pada tabel.

Baris pertama: **(1, "Adrianty", "", "Pemrograman web", "Ketua Jurusan", 34, '1982-06-29')**:

1: ID guru dengan nilai 1.

"Adrianty": Nama depan guru.

" ": Nama belakang guru (kosong).

"Pemrograman web": Mata pelajaran yang diajar.

"Ketua Jurusan": Jabatan guru.

34: Usia guru.

'1982-06-29': Tanggal lahir guru.

Baris-baris berikutnya mengikuti format yang sama, dengan data guru yang berbeda.

Kesimpulan

Pernyataan **INSERT INTO** ini menambahkan 4 data baru ke dalam tabel "tabel_guru". Data-data ini meliputi informasi tentang ID guru, nama depan, nama belakang (jika ada), mata pelajaran yang diajar, jabatan, usia, dan tanggal lahir. Dengan memasukkan data-data ini,

soal 3

Query

```
MariaDB [sekolah_fatsa]> INSERT INTO tabel_guru
-> VALUES
-> (5, "Ayu", "Cantika", "Sistem Komputer", "guru", 25, '1998-12-29');
Query OK, 1 row affected (0.015 sec)
```

Analisis

INSERT INTO tabel_guru: Pernyataan SQL ini digunakan untuk menambahkan data baru ke dalam tabel "tabel_guru".

VALUES: Kata kunci ini digunakan untuk menentukan nilai-nilai yang akan dimasukkan ke dalam tabel.

Baris **(5, "Ayu", "Cantika", "Sistem Komputer", "guru", 25, '1998-12-29')** menambahkan satu data baru ke dalam tabel:

5: ID guru dengan nilai 5.

"Ayu": Nama depan guru.

- "Cantika": Nama belakang guru.
- "Sistem Komputer": Mata pelajaran yang diajar.
- "guru": Jabatan guru.
- 25: Usia guru.
- '1998-12-29': Tanggal lahir guru.

Kesimpulan

Pernyataan **INSERT INTO** ini menambahkan satu data baru ke dalam tabel "tabel_guru". Data baru ini mencakup informasi tentang ID guru, nama depan, nama belakang, mata pelajaran yang diajar, jabatan, usia, dan tanggal lahir. Dengan menambahkan data ini, tabel "tabel_guru" akan memiliki satu entri guru baru yang dapat digunakan dalam sistem.

soal 4

Query

```
MariaDB [sekolah_fatsa]> select * from tabel_guru;
```

Analisis

SELECT * FROM tabel_guru; Pernyataan SQL ini digunakan untuk menampilkan semua kolom dan baris dari tabel "tabel_guru".

SELECT * artinya memilih semua kolom dari tabel.

FROM tabel_guru menentukan bahwa data akan diambil dari tabel "tabel_guru".

Kesimpulan

sistem akan menampilkan seluruh data yang tersimpan dalam tabel "tabel_guru".Ini memungkinkan Anda melihat seluruh daftar guru yang tersimpan dalam tabel.

soal 5

Query

```
MariaDB [sekolah_fatsa]> SELECT * FROM tabel_guru  
-> WHERE nama_depan = "Rusdyansyar";  
Empty set, 10 warnings (0.011 sec)
```

Analisis

SELECT * FROM tabel_guru : Ini memilih semua kolom dari tabel "tabel_guru".

WHERE nama_depan = "Rusdyansyar": Ini menambahkan kondisi agar hanya menampilkan data di mana nilai kolom "nama_depan" sama dengan "Rusdyansyar".

Kesimpulan

Dengan pernyataan yang benar, sistem akan menampilkan semua data dari tabel "tabel_guru" di mana nama depan guru adalah "Rusdyansyar".

soal 6

Query

```
MariaDB [sekolah_fatsa]> UPDATE tabel_guru  
-> SET nama_belakang = "Ganteng"  
-> WHERE id_guru = 3;  
Query OK, 1 row affected (0.017 sec)  
Rows matched: 1 Changed: 1 Warnings: 0
```

Analisis

UPDATE tabel_guru: Pernyataan SQL ini digunakan untuk memperbarui data yang ada dalam tabel "tabel_guru".

SET nama_belakang = "Ganteng": Ini menentukan bahwa kolom "nama_belakang" akan diubah menjadi nilai "Ganteng" untuk data yang cocok dengan kondisi selanjutnya.

WHERE id_guru = 3: Kondisi ini memastikan bahwa hanya data dengan nilai "id_guru" sama dengan 3 yang akan diperbarui. Ini membatasi perubahan hanya pada entri guru tertentu.

Kesimpulan

Pernyataan **UPDATE** ini akan mengubah nilai "nama_belakang" menjadi "Ganteng" untuk data guru yang memiliki "id_guru" bernilai 3 di dalam tabel "tabel_guru". Ini memungkinkan Anda untuk memperbarui informasi tertentu untuk data guru yang spesifik, tanpa mengubah data guru lainnya dalam tabel.

soal 7

Query

```
MariaDB [sekolah_fatsa]> DELETE FROM tabel_guru  
-> WHERE id_guru = 3;  
Query OK, 1 row affected (0.012 sec)
```

Analisis

DELETE FROM tabel_guru: Pernyataan SQL ini digunakan untuk menghapus data dari tabel "tabel_guru".

WHERE id_guru = 3: Kondisi ini memastikan bahwa hanya data dengan nilai "id_guru" sama dengan 3 yang akan dihapus. Ini membatasi penghapusan hanya pada entri guru tertentu.

Kesimpulan

Pernyataan **DELETE** ini akan menghapus baris data dari tabel "tabel_guru" di mana nilai "id_guru" sama dengan 3. Setelah pernyataan ini dijalankan, baris data untuk guru dengan ID 3 akan dihapus dari tabel. Ini memungkinkan Anda untuk menghapus data guru tertentu dari tabel tanpa mempengaruhi data guru lainnya.

soal 8

Query

```
MariaDB [sekolah_fatsa]> SELECT * FROM tabel_guru  
-> WHERE usia <= 30 AND mapel LIKE 'pem%'  
-> ORDER BY usia ASC;
```

Analisis

SELECT * FROM tabel_guru: Ini memilih semua kolom dari tabel "tabel_guru".

WHERE usia <= 30 AND mapel LIKE 'pem%': Ini menambahkan dua kondisi filter:

usia <= 30: Hanya menampilkan data guru dengan usia kurang dari atau sama dengan 30 tahun.

mapel LIKE 'pem%': Hanya menampilkan data guru yang mengajar mata pelajaran yang dimulai dengan "pem" (misalnya "pemrograman", "pembukuan", dll.).

ORDER BY usia ASC: Ini mengurutkan hasil query berdasarkan kolom "usia" secara ascending (dari yang terkecil ke yang terbesar).

Kesimpulan

Pernyataan SQL ini akan menampilkan semua data dari tabel "tabel_guru" di mana usia guru kurang dari atau sama dengan 30 tahun dan mengajar mata pelajaran yang dimulai dengan "pem". Data akan ditampilkan dalam urutan ascending berdasarkan usia guru. Ini memungkinkan Anda untuk mendapatkan daftar guru muda yang mengajar mata pelajaran tertentu, diurutkan dari yang paling muda.

soal 9

Query

```
MariaDB [sekolah_fatsa]> SELECT id_guru, nama_depan FROM tabel_guru  
-> WHERE nama_depan BETWEEN "Adrianty" AND "Ibrahim";
```

Analisis

SELECT id_guru, nama_depan: Ini memilih kolom "id_guru" dan "nama_depan" dari tabel.

FROM tabel_guru: Menentukan bahwa data akan diambil dari tabel "tabel_guru".

WHERE nama_depan BETWEEN "Adrianty" AND "Ibrahim": Ini menambahkan kondisi filter untuk hanya menampilkan data di mana nilai kolom "nama_depan" berada di antara "Adrianty" dan "Ibrahim" (termasuk keduanya).

Kesimpulan

Pernyataan SQL ini akan menampilkan ID guru dan nama depan untuk semua guru yang memiliki nama depan antara "Adrianty" dan "Ibrahim" (termasuk keduanya) yang tersimpan dalam tabel "tabel_guru".

Ini berguna jika Anda ingin mendapatkan daftar guru dengan nama depan tertentu, tanpa perlu mengetahui semua nama depan secara spesifik.

soal 10

Query

```
MariaDB [sekolah_fatsa]> SELECT CONCAT(nama_depan, ' ', nama_belakang) AS nama_lengkap  
-> FROM tabel_guru;
```

Analisis

SELECT CONCAT(nama_depan, ' ', nama_belakang) AS nama_lengkap: Ini menggunakan fungsi **CONCAT()** untuk menggabungkan nilai kolom "nama_depan" dan "nama_belakang" dengan spasi di antaranya. Hasil gabungan ini akan ditampilkan dengan alias "nama_lengkap".

FROM tabel_guru: Menentukan bahwa data akan diambil dari tabel "tabel_guru".

Kesimpulan

Pernyataan SQL ini akan menampilkan satu kolom bernama "nama_lengkap" yang berisi nama lengkap guru (nama depan dan nama belakang digabungkan) dari semua baris dalam tabel "tabel_guru".

Ini berguna jika Anda ingin menampilkan nama lengkap guru secara mudah, tanpa perlu menggabungkan kolom "nama_depan" dan "nama_belakang" secara manual.

soal 11

Query

```
MariaDB [sekolah_fatsa]> ALTER TABLE tabel_guru  
-> ADD COLUMN status ENUM('PNS', 'PPPK', 'Honorer') NOT NULL;  
Query OK, 0 rows affected (0.078 sec)
```

Analisis

ALTER TABLE tabel_guru: Ini menunjukkan bahwa perubahan akan dilakukan pada tabel "tabel_guru".

ADD COLUMN status ENUM('PNS', 'PPPK', 'Honorer') NOT NULL: Ini menambahkan sebuah kolom baru bernama "status" ke tabel "tabel_guru". Tipe data kolom ini adalah ENUM, yang berarti hanya salah

satu dari tiga nilai yang ditentukan ('PNS', 'PPPK', 'Honorar') yang dapat dimasukkan. Juga, kolom ini ditandai sebagai NOT NULL, artinya setiap baris harus memiliki nilai terisi untuk kolom ini.

Kesimpulan

Pernyataan SQL ini akan menambahkan sebuah kolom baru bernama "status" ke tabel "tabel_guru". Kolom ini akan menyimpan informasi status kepegawaian guru, yang dapat berupa 'PNS', 'PPPK', atau 'Honorar'. Setiap baris dalam tabel harus memiliki nilai yang diisi untuk kolom ini.

soal 12

Query

```
MariaDB [sekolah_fatsa]> SELECT nama_depan, MAX(usia) AS usia_tertua  
-> FROM tabel_guru;
```

Analisis

SELECT nama_depan, MAX(usia) AS usia_tertua: Ini memilih dua kolom: "nama_depan" dan nilai maksimum dari kolom "usia" yang akan ditampilkan dengan alias "usia_tertua".

FROM tabel_guru: Menentukan bahwa data akan diambil dari tabel "tabel_guru".

Kesimpulan

Ini berguna jika Anda ingin mengetahui usia tertua guru dalam tabel tersebut. Perlu diperhatikan bahwa hasil hanya akan menampilkan satu baris, karena fungsi **MAX()** akan mengembalikan nilai usia tertua secara keseluruhan.