

Nicky Brilliant Suryadi Langit

2024071027

Semua jawaban teori dari page 1-81

PRAKTIKUM 1 – JAWABAN TEORI

1. Apa perbedaan antara *database* dan *tabel*?

Database adalah wadah besar yang berfungsi untuk menyimpan berbagai macam tabel. Ibarat sebuah lemari besar, di dalamnya terdapat banyak rak. Sedangkan tabel adalah struktur penyimpanan data dalam bentuk baris dan kolom, mirip seperti sebuah spreadsheet. Satu database bisa berisi banyak tabel, tetapi satu tabel tidak dapat berdiri sendiri tanpa berada di dalam database.

2. Mengapa penting memberikan *PRIMARY KEY* pada sebuah tabel?

Primary key digunakan sebagai identitas unik setiap baris data. Fungsi utamanya adalah mencegah data yang sama tersimpan dua kali serta memudahkan proses pencarian, update, dan penghapusan data secara tepat. Tanpa primary key, database akan sulit menentukan mana data yang benar atau mana yang duplikat.

3. Mengapa perlu mendefinisikan tipe data (INT, VARCHAR, DATE, dll.)?

Setiap tipe data punya peran berbeda. Misalnya:

- **INT** untuk angka
- **VARCHAR** untuk teks
- **DATE** untuk tanggal

Dengan mendefinisikan tipe data yang tepat, database dapat mengelola data dengan lebih efisien, mencegah kesalahan input, dan mengoptimalkan penggunaan ruang penyimpanan.

4. Apa kegunaan perintah **CREATE DATABASE** dan **CREATE TABLE**?

- CREATE DATABASE digunakan untuk membuat wadah penyimpanan data baru.
- CREATE TABLE digunakan untuk membuat struktur tabel di dalam database tersebut.

Tanpa kedua perintah ini, sistem tidak akan memiliki tempat ataupun struktur untuk menyimpan data.

5. Mengapa kita perlu menambahkan kolom baru pada tabel?

Seiring berjalannya waktu, kebutuhan sistem sering berubah. Misalnya awalnya hanya menyimpan nama dan alamat, tetapi kemudian dibutuhkan juga nomor telepon. Dengan menambah kolom, tabel bisa menampung informasi baru tanpa harus membuat tabel baru.

6. Apa fungsi dari perintah DROP TABLE?

Perintah ini digunakan untuk menghapus tabel beserta seluruh data di dalamnya. Biasanya dilakukan ketika tabel sudah tidak diperlukan, salah desain, atau ingin diganti dengan struktur tabel yang benar.

7. Apa perbedaan INSERT, SELECT, UPDATE, dan DELETE?

Keempat perintah ini adalah dasar dari manipulasi data:

- **INSERT** → menambahkan data baru
- **SELECT** → menampilkan data
- **UPDATE** → mengubah data
- **DELETE** → menghapus data

Masing-masing memiliki tujuan berbeda dan digunakan sesuai kebutuhan.

8. Mengapa perintah WHERE sangat penting?

WHERE digunakan untuk menentukan data mana yang ingin ditampilkan, diubah, atau dihapus. Tanpa WHERE, perintah seperti UPDATE dan DELETE bisa berbahaya karena dapat mengubah atau menghapus **semua** baris pada tabel.

9. Apa tujuan menggunakan operator pembanding (=, >, <, >=, dll.) dalam query?

Operator ini membantu database dalam memilih data sesuai kondisi tertentu. Misalnya:

- menampilkan mahasiswa dengan umur > 20
- menampilkan barang dengan harga < 50.000

Tanpa operator ini, database tidak bisa melakukan penyaringan data yang spesifik.

10. Mengapa penting memahami struktur tabel sebelum memasukkan data?

Jika kita tidak tahu urutan kolom atau tipe datanya, kita bisa salah memasukkan nilai. Contohnya memasukkan teks ke kolom angka, atau memasukkan tanggal ke kolom teks. Hal ini dapat menyebabkan error atau data menjadi tidak konsisten.

PRAKTIKUM 2 – JAWABAN TEORI (JOIN & CARTESIAN PRODUCT)

1. Apa yang dimaksud dengan *Cartesian Product* dalam SQL?

Cartesian Product adalah hasil dari penggabungan dua tabel tanpa menggunakan kondisi penghubung. Artinya, setiap baris dari tabel pertama akan dipasangkan dengan setiap baris dari tabel kedua.

Akibatnya, jumlah baris yang muncul bisa menjadi sangat besar.

Contoh:

- tabel A punya 5 baris
- tabel B punya 4 baris

$$\text{Cartesian Product} = 5 \times 4 = \mathbf{20 \text{ baris}}$$

Karena hasilnya terlalu banyak dan tidak selalu relevan, Cartesian Product biasanya **tidak digunakan dalam query nyata**, kecuali untuk kasus tertentu seperti analisis kombinasi data.

2. Mengapa *Cartesian Product* jarang digunakan dalam aplikasi nyata?

Karena hasilnya tidak memiliki hubungan logis antara satu data dengan lainnya. Data menjadi acak dan tidak berguna. Selain itu, query menjadi berat karena menghasilkan jumlah baris yang sangat besar.

Biasanya, kita lebih memilih menggunakan **JOIN**, yang menghubungkan data berdasarkan relasi tertentu.

3. Apa pengertian JOIN dalam SQL?

JOIN adalah cara menggabungkan dua tabel (atau lebih) berdasarkan kolom yang saling berhubungan. JOIN memungkinkan kita menampilkan data yang relevan dari lebih dari satu tabel secara bersamaan.

Contohnya:

- tabel mahasiswa (nim, nama)
- tabel nilai (nim, kode_mk, nilai)

Dengan JOIN, kita bisa menampilkan nama mahasiswa sekaligus nilai mata kuliah mereka.

4. Apa perbedaan mendasar antara JOIN dan Cartesian Product?

- **Cartesian Product** menggabungkan semua baris, tanpa syarat
- **JOIN** menggabungkan baris yang cocok berdasarkan kolom tertentu

Jadi perbedaannya adalah **kondisi penghubung**. JOIN menghasilkan data yang terfilter dan relevan, sementara Cartesian Product tidak.

5. Jelaskan perbedaan antara INNER JOIN dan OUTER JOIN.

INNER JOIN

Hanya menampilkan data yang cocok di kedua tabel.

Jika salah satu tabel tidak punya pasangan, datanya tidak ditampilkan.

OUTER JOIN

Menampilkan data meskipun tidak ada pasangan di tabel lain.

Dibagi menjadi:

- **LEFT OUTER JOIN** → semua dari kiri + pasangan dari kanan
- **RIGHT OUTER JOIN** → semua dari kanan + pasangan dari kiri
- **FULL OUTER JOIN** → semua data dari kedua tabel, cocok atau tidak cocok

Dengan OUTER JOIN, data yang tidak memiliki pasangan tetap muncul, biasanya ditandai NULL.

6. Kapan kita menggunakan INNER JOIN?

Kita menggunakan INNER JOIN saat kita hanya ingin menampilkan data yang benar-benar memiliki hubungan di kedua tabel.

Contoh:

menampilkan mahasiswa yang **sudah** punya nilai → hanya yang ada di kedua tabel.

7. Kapan kita menggunakan LEFT JOIN?

LEFT JOIN digunakan ketika kita ingin menampilkan **semua** data dari tabel kiri, meskipun tidak memiliki pasangan di tabel kanan.

Contoh:

Menampilkan semua mahasiswa, termasuk yang **belum punya nilai**.

Mahasiswa yang belum punya nilai akan muncul dengan nilai = NULL.

8. Mengapa diperlukan JOIN ketika bekerja dengan banyak tabel?

Karena dalam database relasional, data biasanya dipisahkan ke dalam tabel-tabel kecil agar lebih rapi dan tidak duplikat. JOIN memudahkan kita menggabungkan kembali informasi tersebut ketika dibutuhkan.

Tanpa JOIN, data tidak akan lengkap dan hubungan antar tabel tidak bisa dibaca.

9. Apa yang terjadi jika JOIN dilakukan tanpa ON atau USING?

Hasilnya akan menjadi **Cartesian Product**, yaitu semua baris saling dikalikan.

Hasilnya berantakan dan tidak relevan.

Ini adalah kesalahan umum bagi pemula, sehingga penting untuk selalu menuliskan kondisi JOIN.

10. Mengapa kolom yang menjadi penghubung JOIN harus memiliki tipe data yang sama?

Karena JOIN membandingkan nilai kolom tersebut.

Jika salah satu kolom adalah VARCHAR dan yang lain INT, database tidak bisa membandingkan

secara logis.

Selain itu, kesalahan tipe data dapat menyebabkan JOIN gagal atau hasil yang tidak akurat.

PRAKTIKUM 3 – JAWABAN TEORI (Single Row Function)

1. Apa yang dimaksud dengan Single Row Function?

Single Row Function adalah fungsi yang bekerja pada setiap baris data secara individual dan menghasilkan satu output untuk setiap baris tersebut. Artinya, jika tabel memiliki 10 baris, maka fungsi ini juga menghasilkan 10 output.

Contoh: UPPER(), LOWER(), LENGTH(), ABS(), ROUND().

2. Mengapa single row function penting dalam manipulasi data?

Karena fungsi ini memudahkan kita mengolah data langsung di query tanpa harus mengubah data asli di tabel. Misalnya mengubah huruf menjadi kapital, mengambil sebagian teks, membulatkan angka, atau menghitung panjang karakter.

3. Apa perbedaan antara fungsi karakter dan fungsi numerik?

Fungsi karakter bekerja pada teks (VARCHAR), seperti UPPER(), LOWER(), SUBSTRING(), LENGTH().

Fungsi numerik bekerja pada angka, seperti ROUND(), CEIL(), FLOOR(), MOD().

Perbedaannya ada pada jenis data yang diproses.

4. Jelaskan fungsi UPPER() dan LOWER().

UPPER() mengubah seluruh teks menjadi huruf besar.

LOWER() mengubah seluruh teks menjadi huruf kecil.

Fungsi ini berguna untuk standarisasi data.

5. Jelaskan fungsi LENGTH().

LENGTH() digunakan untuk menghitung jumlah karakter dalam sebuah string. Fungsi ini berguna untuk validasi panjang data atau analisis isi teks.

6. Jelaskan fungsi SUBSTRING().

SUBSTRING() digunakan untuk mengambil sebagian teks dari suatu string berdasarkan posisi awal dan panjang karakter.

Contoh: SUBSTRING('INFORMATIKA', 1, 4) → INFO

7. Jelaskan fungsi ROUND().

ROUND() membulatkan angka ke jumlah digit tertentu.

Misalnya ROUND(12.567, 2) → 12.57.

Fungsi ini penting dalam perhitungan keuangan atau perhitungan yang membutuhkan presisi tertentu.

8. Kapan kita menggunakan fungsi numerik seperti CEIL() dan FLOOR()?

CEIL() digunakan untuk membulatkan angka ke atas.

FLOOR() membulatkan angka ke bawah.

Fungsinya biasanya dipakai dalam analisis perhitungan diskon, pajak, atau pembagian yang butuh pembulatan tertentu.

9. Apa yang dimaksud dengan fungsi tanggal (date function)?

Fungsi tanggal adalah fungsi yang digunakan untuk memproses dan mengambil informasi dari data bertipe DATE.

Contoh:

- NOW() → menampilkan tanggal dan waktu saat ini
- CURDATE() → menampilkan tanggal hari ini
- YEAR(), MONTH(), DAY() → mengambil bagian tertentu dari tanggal

10. Mengapa single row function harus dikuasai oleh pengguna SQL?

Karena hampir semua operasi data dalam aplikasi memerlukan manipulasi pada tingkat baris, seperti mengolah nama, menghitung angka, atau mengekstrak tanggal tertentu.

Function ini juga mengurangi kebutuhan mengedit data manual dan membuat query lebih fleksibel.

PRAKTIKUM 4 – JAWABAN TEORI (ORDER BY, SORTING DATA)

1. Apa fungsi utama dari ORDER BY dalam SQL?

ORDER BY digunakan untuk mengurutkan data hasil query. Pengurutan bisa berdasarkan satu kolom atau beberapa kolom, dan bisa dilakukan secara menaik (ASC) atau menurun (DESC).

Fungsi ini berguna untuk membuat hasil data lebih rapi dan mudah dibaca.

2. Apa perbedaan ASC dan DESC?

ASC berarti pengurutan dari yang paling kecil ke yang besar (ascending).

DESC berarti dari yang besar ke yang kecil (descending).

Contoh:

- Harga diurutkan dari yang termurah → ASC
- Nilai diurutkan dari yang tertinggi → DESC

3. Mengapa ORDER BY penting dalam laporan data?

Karena urutan data dapat mempengaruhi cara kita memahami informasi. Laporan keuangan,

ranking nilai, data penjualan, hingga daftar produk membutuhkan pengurutan supaya informasi mudah dianalisis oleh pengguna atau manajemen.

4. Apa yang terjadi jika ORDER BY digunakan pada kolom yang berisi teks?

SQL akan mengurutkan teks berdasarkan urutan alfabet. Misal: A-Z untuk ASC dan Z-A untuk DESC.

Runtutan ini tetap konsisten, sehingga cocok untuk mengurutkan nama, kota, atau jenis kategori lain.

5. Apa yang terjadi jika kita mengurutkan data berdasarkan lebih dari satu kolom?

Ketika kita menggunakan ORDER BY dengan lebih dari satu kolom, SQL akan melakukan pengurutan berlapis.

Contoh:

ORDER BY jurusan ASC, nama ASC

Artinya data diurutkan berdasarkan jurusan dulu, baru kemudian nama di dalam jurusan tersebut.

6. Apakah kita bisa menggunakan ORDER BY pada kolom yang tidak ditampilkan di SELECT?

Bisa. ORDER BY tetap bisa bekerja meskipun kolom tersebut tidak ditampilkan. SQL tetap dapat membaca dan mengurutkannya secara internal.

7. Apa perbedaan pengurutan numerik dan pengurutan teks?

Pengurutan numerik berdasarkan nilai angka.

Pengurutan teks berdasarkan alfabet.

Masalah sering terjadi jika angka disimpan dalam bentuk teks karena hasil pengurutannya bisa tidak sesuai logika angka (misalnya "100" bisa muncul sebelum "2").

8. Mengapa kolom hasil perhitungan (misal fungsi SUM atau LENGTH) masih bisa diurutkan?

Karena SQL memperlakukan hasil perhitungan sebagai kolom sementara yang dapat disortir.

Contoh:

SELECT nama, LENGTH(nama) FROM mahasiswa ORDER BY LENGTH(nama);

SQL akan mengurutkan berdasarkan panjang nama, bukan nama asli.

9. Apa yang terjadi jika ORDER BY tidak digunakan sama sekali dalam query?

SQL akan menampilkan data tanpa urutan khusus.

Urutannya tergantung cara data disimpan di memori atau di disk, dan bisa berubah-ubah, sehingga tidak cocok untuk laporan formal.

10. Mengapa penting memahami ORDER BY sejak awal belajar SQL?

Karena hampir semua aplikasi membutuhkan data yang tersusun rapi. Mulai dari laporan sederhana sampai dashboard analitik, pengurutan adalah hal yang mendasar dalam penyajian data. Menguasai ORDER BY membuat query lebih profesional dan mudah dibaca.

PRAKTIKUM 5 – JAWABAN TEORI (SUBQUERY & UNION)

1. Apa yang dimaksud dengan Subquery?

Subquery adalah query di dalam query lain. Biasanya ditempatkan pada bagian SELECT, FROM, atau WHERE untuk membantu mengambil data yang lebih kompleks. Subquery bekerja seperti “query kecil” yang menyediakan hasil sementara bagi query utama.

Fungsinya mirip seperti sebuah langkah perhitungan yang mendukung langkah berikutnya.

2. Mengapa subquery diperlukan dalam SQL?

Karena tidak semua kebutuhan bisa diselesaikan dengan satu query sederhana. Subquery membantu ketika:

- kita ingin mendapatkan data berdasarkan hasil perhitungan lain,
- kita membutuhkan nilai sementara,
- atau kita ingin membandingkan data antara dua tabel tanpa JOIN.

Singkatnya, subquery membuat query lebih fleksibel dan lebih kuat.

3. Apa perbedaan Subquery dan JOIN?

JOIN menggabungkan dua tabel berdasarkan kolom yang saling berhubungan, sedangkan subquery bekerja sebagai query bersarang.

JOIN digunakan untuk menampilkan kolom dari beberapa tabel sekaligus.

Subquery digunakan ketika kita ingin mengambil data berdasarkan hasil query lain, bukan menggabungkan tabel langsung.

Biasanya subquery lebih mudah dipahami, tapi JOIN lebih efisien.

4. Jelaskan apa itu UNION.

UNION digunakan untuk menggabungkan dua hasil SELECT menjadi satu, selama jumlah kolom dan tipe datanya sama.

Misalnya, menggabungkan daftar mahasiswa kelas A dan kelas B dalam satu tampilan.

UNION menghindari duplikasi data karena otomatis menghapus data yang sama.

5. Apa perbedaan UNION dan UNION ALL?

- UNION menggabungkan hasil sekaligus menghilangkan baris duplikat.
- UNION ALL menggabungkan hasil **tanpa** menghapus duplikasi.
UNION ALL lebih cepat karena tidak perlu memeriksa duplikasi.

6. Kapan sebaiknya menggunakan UNION?

UNION digunakan ketika kita memang ingin memastikan data yang muncul tidak ada yang ganda. Biasanya dipakai pada laporan unik, misalnya daftar semua pelanggan dari dua cabang.

7. Kapan sebaiknya menggunakan UNION ALL?

UNION ALL digunakan ketika kita ingin menggabungkan data apa adanya, dan kinerja lebih diutamakan. Ini digunakan pada dataset besar dan situasi di mana duplikasi bukan masalah.

8. Apa kegunaan subquery di bagian WHERE?

Subquery digunakan di WHERE untuk mengambil nilai pembanding.

Contoh klasik: memilih mahasiswa yang memiliki nilai di atas rata-rata:

nilai > (SELECT AVG(nilai) FROM nilai)

Artinya, subquery membantu menghasilkan angka atau data pembanding untuk query utama.

9. Apa kelemahan dari penggunaan subquery secara berlebihan?

Subquery terlalu banyak dapat membuat query lambat dan sulit dibaca.

Kadang-kadang apa yang dilakukan subquery sebenarnya bisa diganti dengan JOIN yang jauh lebih cepat.

Subquery sebaiknya digunakan ketika benar-benar diperlukan.

10. Mengapa memahami UNION dan Subquery penting dalam pembuatan laporan?

Karena laporan sering membutuhkan data dari beberapa sumber, beberapa tabel, atau hasil penggabungan dan perhitungan kompleks. UNION dan Subquery memungkinkan kita mengambil data yang lebih kaya dan lengkap, sehingga laporan menjadi akurat dan informatif.

PRAKTIKUM 6 – JAWABAN TEORI (VIEW, VARIABLE, CONTROL FLOW)

1. Apa yang dimaksud dengan VIEW dalam SQL?

VIEW adalah tabel virtual yang dibuat dari hasil sebuah query. VIEW tidak menyimpan data secara fisik, tetapi hanya menyimpan perintah query yang nantinya akan dipanggil.

View sangat berguna untuk menyederhanakan query yang panjang, membatasi akses data, dan membuat tampilan data menjadi lebih rapi bagi pengguna.

2. Mengapa VIEW sering digunakan dalam sistem database?

Karena VIEW dapat membantu:

- menyembunyikan struktur tabel yang rumit,
- membatasi hak akses pada kolom tertentu,
- mempercepat pekerjaan pengguna agar tidak perlu menulis query panjang berulang-ulang.

Intinya, VIEW membantu membuat database lebih aman, mudah digunakan, dan lebih efisien.

3. Apa kelemahan dari penggunaan VIEW?

VIEW tidak selalu cocok untuk operasi INSERT, UPDATE, dan DELETE, terutama jika view dibuat dari beberapa tabel sekaligus. Selain itu, VIEW bisa memperlambat kinerja jika query yang mendasarinya sangat kompleks dan sering dipanggil.

4. Apa yang dimaksud dengan variabel dalam SQL?

Variabel adalah tempat untuk menyimpan nilai sementara saat menjalankan query atau prosedur. Variabel biasanya diawali dengan simbol seperti @, contoh:

SET @total = 100;

Variabel digunakan untuk menghitung, menyimpan hasil perhitungan sementara, atau dipakai dalam kondisi tertentu.

5. Mengapa variabel diperlukan dalam perhitungan SQL?

Karena variabel membantu menyimpan nilai yang ingin digunakan berulang kali dalam satu transaksi. Misalnya total belanja, jumlah mahasiswa, atau nilai sementara yang diproses dan dianalisis sebelum disimpan dalam tabel.

6. Apa yang dimaksud dengan Control Flow?

Control Flow adalah mekanisme logika dalam SQL yang mengatur jalannya alur eksekusi. Control Flow memungkinkan query berjalan berdasarkan kondisi tertentu, mirip dengan IF/ELSE dalam bahasa pemrograman.

Contohnya:

IF, CASE, atau IFNULL.

7. Jelaskan fungsi IF() dalam SQL.

IF() digunakan untuk memilih nilai berdasarkan kondisi tertentu. Jika kondisi benar, maka satu nilai ditampilkan, jika salah maka nilai lainnya ditampilkan.

Contoh:

SELECT IF(nilai >= 60, 'Lulus', 'Tidak Lulus');

8. Apa perbedaan CASE dan IF?

CASE lebih fleksibel dan bisa digunakan untuk banyak kondisi sekaligus, sedangkan IF hanya cocok untuk kondisi sederhana (benar/salah).

CASE sering digunakan saat sebuah kolom memiliki lebih dari dua kemungkinan output.

9. Mengapa control flow sangat berguna dalam laporan data?

Karena laporan sering membutuhkan klasifikasi, keterangan, atau status tertentu yang dihitung berdasarkan kondisi. Misalnya:

- kategori nilai: A, B, C
 - status pembayaran: Lunas / Belum
 - result perhitungan berdasarkan kondisi
- Control Flow mempermudah pembuatan informasi tambahan tanpa harus mengubah data asli.

10. Apa hubungan antara VIEW, variabel, dan control flow dalam SQL?

Ketiganya digunakan untuk membuat pengelolaan data lebih efisien dan fleksibel. VIEW menyederhanakan tampilan, variabel menyimpan nilai sementara, dan control flow memberikan logika pada query.

Ketika digabung, kita dapat membuat query yang jauh lebih kuat untuk analisis maupun laporan.

PRAKTIKUM 7 – JAWABAN TEORI (UNION, INTERSECT, EXCEPT)

1. Apa perbedaan utama antara UNION, INTERSECT, dan EXCEPT?

- **UNION** menggabungkan dua hasil query menjadi satu dan menghapus duplikasi.
- **INTERSECT** mengambil bagian data yang sama-sama muncul di kedua query.
- **EXCEPT** mengambil data yang ada di query pertama tetapi tidak ada di query kedua.
Ketiganya digunakan untuk mengolah hasil SELECT dan bukan untuk menggabungkan struktur tabel.

2. Mengapa UNION membutuhkan jumlah kolom dan tipe data yang sama?

Karena UNION menggabungkan dua kumpulan data secara vertikal, sehingga setiap baris dari query pertama harus memiliki pasangan struktur yang sama dengan query kedua. Jika jumlah atau tipe kolom berbeda, SQL tidak dapat menyatukan baris-baris tersebut.

3. Apa perbedaan UNION dan UNION ALL?

- UNION menghapus baris yang duplikat.
- UNION ALL menampilkan semua baris, termasuk yang duplikat.
UNION ALL lebih cepat karena tidak perlu memeriksa data yang sama.

4. Apa kegunaan INTERSECT dalam analisis data?

INTERSECT digunakan untuk mencari data yang muncul di dua daftar sekaligus. Misalnya mencari pelanggan yang membeli produk A **dan** produk B, atau siswa yang terdaftar di dua kelas.

Fungsi ini mempermudah mencari irisan data.

5. Apa kegunaan EXCEPT dalam SQL?

EXCEPT digunakan untuk mencari data yang unik di query pertama yang tidak muncul di query kedua.

Contoh kegunaan:

- mencari siswa yang sudah bayar tetapi tidak terdaftar di kelas,
- mencari barang yang ada di gudang A tetapi tidak ada di gudang B.

6. Mengapa INTERSECT dan EXCEPT tidak tersedia di semua versi MySQL?

Karena MySQL lebih lama fokus pada UNION dan JOIN daripada operator set lengkap. Pada beberapa versi lama, INTERSECT dan EXCEPT tidak tersedia dan harus disimulasikan menggunakan JOIN.

Namun konsepnya tetap penting untuk dipahami sebagai operator himpunan dalam SQL.

7. Bagaimana cara meniru INTERSECT jika database tidak mendukungnya?

Menggunakan INNER JOIN.

Contoh:

```
SELECT kolom FROM tabelA
```

```
INNER JOIN tabelB USING(kolom);
```

INNER JOIN menghasilkan baris yang ada di kedua tabel, sama seperti INTERSECT.

8. Bagaimana cara meniru EXCEPT jika database tidak mendukungnya?

Menggunakan LEFT JOIN dengan kondisi IS NULL.

Contoh:

```
SELECT a.kolom
```

```
FROM tabelA a
```

```
LEFT JOIN tabelB b USING(kolom)
```

```
WHERE b.kolom IS NULL;
```

Artinya menampilkan data yang hanya ada di A tetapi tidak di B.

9. Mengapa operator himpunan sangat berguna dalam laporan data?

Karena membantu menggabungkan, membandingkan, atau memisahkan data dari berbagai sumber.

Hal ini sering digunakan dalam:

- laporan kehadiran,

- laporan transaksi,
- daftar gabungan beberapa cabang,
- analisis data pelanggan.

10. Kapan lebih baik menggunakan operator himpunan dibanding JOIN?

Operator himpunan dipakai ketika **struktur kolom sama**, dan kita ingin menggabungkan atau membandingkan dua hasil query.

JOIN dipakai ketika kita ingin menampilkan kolom dari beberapa tabel sekaligus.

Jadi operator himpunan lebih cocok untuk analisis data, sedangkan JOIN lebih cocok untuk relasi tabel.

PRAKTIKUM 8 – JAWABAN TEORI (DDL DASAR: CREATE, ALTER, DROP)

1. Apa yang dimaksud dengan DDL (Data Definition Language)?

DDL adalah sekumpulan perintah SQL yang digunakan untuk membuat, mengubah, dan menghapus struktur database. DDL tidak berhubungan dengan isi data, tetapi berfokus pada bentuk tabel, kolom, dan objek lainnya.

Contoh perintah DDL: CREATE, ALTER, DROP.

2. Apa fungsi utama dari perintah CREATE?

CREATE digunakan untuk membuat objek baru di database, seperti:

- database,
- tabel,
- view,
- index.

Perintah ini selalu digunakan saat memulai pembuatan struktur database dari nol.

3. Apa fungsi dari perintah ALTER TABLE?

ALTER TABLE digunakan untuk mengubah struktur tabel yang sudah ada. Misalnya:

- menambah kolom baru,
- menghapus kolom,
- mengubah tipe data kolom,

- mengganti nama kolom.
- ALTER sangat berguna saat kebutuhan sistem berubah.

4. Mengapa kita perlu menggunakan perintah DROP TABLE?

DROP TABLE digunakan untuk menghapus tabel beserta seluruh isinya. DROP TABLE sering digunakan bila struktur tabel sudah tidak sesuai, salah desain, atau ingin mengganti tabel dengan yang baru.

Namun DROP bersifat permanen dan tidak bisa dibatalkan.

5. Apa perbedaan TRUNCATE dan DROP?

- TRUNCATE hanya menghapus semua data di dalam tabel, tetapi struktur tabel tetap ada.
 - DROP menghapus seluruh tabel termasuk struktur dan datanya.
- TRUNCATE cocok digunakan ketika kita ingin mengosongkan tabel tanpa menghapus bentuk tabelnya.

6. Mengapa penentuan tipe data penting ketika membuat tabel?

Karena tipe data menentukan bagaimana database menyimpan dan memproses data.

Jika salah memilih tipe data, bisa menyebabkan:

- pemborosan penyimpanan,
- error,
- perhitungan yang tidak tepat,
- data tidak bisa diurutkan atau dicari dengan benar.

7. Apa tujuan penggunaan NOT NULL dalam kolom?

NOT NULL memastikan bahwa kolom tersebut wajib diisi.

Ini digunakan pada data yang penting, seperti NIM mahasiswa atau kode barang, supaya tidak ada baris yang tidak lengkap.

8. Mengapa PRIMARY KEY harus unik?

Karena PRIMARY KEY adalah identitas dari setiap baris. Jika ada nilai yang sama, database tidak bisa membedakan data mana yang benar.

PRIMARY KEY juga digunakan untuk melakukan relasi antar tabel melalui FOREIGN KEY.

9. Apa perbedaan CREATE DATABASE dan USE?

- CREATE DATABASE → membuat database baru

- USE → memilih database yang ingin dipakai sebelum membuat tabel atau memasukkan data
USE memastikan bahwa semua perintah berikutnya berjalan di database yang tepat.

10. Mengapa perintah DDL harus dipelajari sebelum DML?

Karena struktur database harus dibuat dulu sebelum datanya bisa dimasukkan atau dimanipulasi.

DDL adalah pondasi awal, sedangkan DML (INSERT, UPDATE, DELETE) bekerja pada struktur yang dibuat oleh DDL.

PRAKTIKUM 9 – JAWABAN TEORI (HAK AKSES USER / GRANT)

1. Apa tujuan utama membuat user baru di dalam MySQL?

Tujuan utamanya adalah membagi akses dan keamanan dalam sistem database. Setiap user bisa diberi hak akses berbeda sesuai kebutuhan. Dengan begitu, tidak semua orang bisa mengubah, menghapus, atau melihat semua data. Ini penting untuk menjaga keamanan dan integritas database.

2. Mengapa hak akses (privileges) perlu diatur?

Hak akses menentukan apa yang boleh dan tidak boleh dilakukan oleh seorang user.

Tanpa pengaturan hak akses:

- data bisa terhapus oleh orang yang tidak berhak,
- pengguna bisa mengubah struktur database,
- keamanan sistem menjadi lemah.

Dengan privilege yang tepat, setiap user hanya bisa bekerja sesuai tugasnya.

3. Apa fungsi perintah GRANT?

GRANT digunakan untuk memberi hak akses kepada user. Hak akses ini dapat berupa:

- SELECT → hanya melihat data
- INSERT → menambah data
- UPDATE → mengubah data
- DELETE → menghapus data
- ALL PRIVILEGES → memberikan semua hak akses

GRANT juga bisa diterapkan di level database, tabel, bahkan kolom.

4. Apa perbedaan hak akses di level global, database, tabel, dan kolom?

- **Global:** berlaku untuk semua database dan semua tabel
- **Database:** berlaku hanya untuk satu database tertentu
- **Tabel:** berlaku hanya untuk tabel tertentu
- **Kolom:** paling spesifik, hanya kolom tertentu dalam tabel
Semakin sempit levelnya, semakin detail kontrolnya.

5. Mengapa kita perlu menggunakan FLUSH PRIVILEGES?

FLUSH PRIVILEGES digunakan untuk memuat ulang hak akses yang baru saja diberikan atau dicabut.

MySQL menyimpan privilege di memori. Dengan FLUSH, perubahan langsung berlaku tanpa perlu restart server.

6. Apa perbedaan GRANT dan REVOKE?

- **GRANT** digunakan untuk memberikan hak akses
- **REVOKE** digunakan untuk mencabut hak akses
Keduanya digunakan untuk mengatur apa yang boleh dilakukan oleh user.

7. Mengapa sistem database perusahaan besar wajib menggunakan user akses terbatas?

Karena setiap bagian perusahaan punya kebutuhan berbeda. Misalnya:

- staf kasir hanya boleh input data
 - manajer hanya boleh melihat laporan
 - admin IT boleh mengatur struktur database
- Dengan user terbatas, risiko kesalahan dan keamanan lebih terjaga.

8. Apa dampaknya jika semua user diberi ALL PRIVILEGES?

Dampaknya sangat berbahaya:

- tabel bisa terhapus tidak sengaja
 - data sensitif bisa terganti
 - sistem bisa rusak jika user salah menjalankan query
- Itulah sebabnya privilege harus diberikan secara selektif.

9. Mengapa penting mencatat daftar user menggunakan query SELECT FROM mysql.user?

Karena administrator harus mengetahui siapa saja yang punya akses ke server database.

Dengan melihat daftar user, admin bisa mengecek apakah ada user tidak dikenal, atau user lama yang seharusnya dihapus.

10. Apa hubungan antara user, password, dan privilege dalam sistem database?

Ketiganya adalah komponen keamanan database:

- user = identitas
- password = autentikasi
- privilege = izin melakukan sesuatu

Jika ketiganya disusun dengan tepat, database menjadi aman dan stabil.

PRAKTIKUM 10 – JAWABAN TEORI (DDL LANJUTAN: CREATE, ALTER, DROP, RENAME, MODIFY)

1. Apa yang dimaksud dengan DDL (Data Definition Language) dalam SQL?

DDL adalah kumpulan perintah SQL yang digunakan untuk mengatur struktur database.

Termasuk membuat, mengubah, dan menghapus tabel ataupun objek lain. DDL tidak mengelola isi data, tetapi mengatur bentuk dan komponen database tempat data disimpan.

2. Apa fungsi perintah CREATE TABLE?

CREATE TABLE digunakan untuk membuat tabel baru lengkap dengan kolom dan tipe datanya.

Perintah ini merupakan langkah pertama sebelum data dapat dimasukkan atau diolah di dalam database.

3. Mengapa perintah ALTER TABLE sangat penting?

ALTER TABLE digunakan untuk mengubah struktur tabel yang sudah ada tanpa harus menghapus tabel.

Penggunaannya meliputi:

- menambah kolom baru,
- menghapus kolom,
- mengubah tipe data,
- mengganti nama kolom,
- menambahkan constraint.

ALTER membuat tabel dapat berkembang sesuai dengan kebutuhan sistem.

4. Apa yang terjadi jika kita menggunakan DROP TABLE?

DROP TABLE menghapus tabel beserta seluruh datanya secara permanen. Tabel yang dihapus tidak bisa dikembalikan kecuali terdapat backup. Oleh karena itu, penggunaannya harus sangat hati-hati.

5. Apa perbedaan DROP TABLE dan DELETE?

- DROP TABLE menghapus seluruh struktur tabel beserta isinya.
- DELETE hanya menghapus isi tabel (baris data), tetapi struktur tabel tetap ada.
DROP adalah aksi struktural (DDL), sedangkan DELETE adalah aksi manipulasi data (DML).

6. Mengapa penentuan tipe data yang tepat sangat penting saat membuat tabel?

Karena tipe data menentukan bagaimana database menyimpan, mengelola, dan membandingkan nilai dalam kolom.

Tipe data yang salah dapat menyebabkan:

- error saat memasukkan data,
- perhitungan salah,
- penggunaan storage yang boros.

Tipe data yang tepat membuat database lebih efisien dan konsisten.

7. Apa fungsi RENAME TABLE?

RENAME TABLE digunakan untuk mengganti nama tabel tanpa mengubah isi atau strukturnya. Fitur ini berguna ketika nama tabel lama kurang tepat, kurang deskriptif, atau perlu diganti agar lebih sesuai dengan kebutuhan sistem.

8. Apa tujuan menggunakan MODIFY dalam ALTER TABLE?

MODIFY digunakan untuk mengubah tipe data atau karakteristik kolom.

Misalnya:

- mengubah panjang VARCHAR,
- memperbesar ukuran INT,
- mengubah kolom menjadi NOT NULL.

MODIFY membantu memastikan kolom tetap sesuai dengan perubahan kebutuhan data.

9. Apa dampak jika kolom PRIMARY KEY diubah atau dihapus?

PRIMARY KEY adalah identitas utama data. Jika dihapus:

- tabel kehilangan keunikan baris,

- relasi dengan tabel lain bisa rusak,
 - proses update dan delete bisa salah target.
- Karena itu perubahan primary key harus dilakukan dengan sangat hati-hati.

10. Mengapa DDL dianggap sebagai pondasi dalam pembuatan database?

Karena sebelum data ditambahkan atau diproses, struktur database harus dibuat terlebih dahulu. DDL menentukan bentuk tabel, hubungan antar tabel, dan aturan dasar penyimpanan data.

Tanpa DDL yang baik, database akan sulit dikembangkan dan rentan terhadap ketidakteraturan.

PRAKTIKUM 11 – JAWABAN TEORI (DML: INSERT, UPDATE, DELETE, SELECT)

1. Apa yang dimaksud dengan DML (Data Manipulation Language)?

DML adalah kumpulan perintah SQL yang digunakan untuk mengolah data di dalam tabel. Berbeda dengan DDL yang mengatur struktur tabel, DML berfokus pada isi tabel, yaitu bagaimana data ditambahkan, diubah, dilihat, dan dihapus.

2. Apa fungsi perintah INSERT?

INSERT digunakan untuk menambahkan baris data baru ke dalam tabel. Perintah ini penting karena merupakan cara utama untuk mengisi data ke dalam database.

3. Mengapa penting menuliskan urutan kolom saat melakukan INSERT?

Karena urutan kolom memastikan data masuk ke tempat yang sesuai. Jika urutan tidak disebutkan, MySQL akan mengikuti default struktur tabel. Kesalahan urutan bisa menyebabkan data salah tempat atau error.

4. Apa fungsi perintah UPDATE?

UPDATE digunakan untuk memperbarui data pada baris tertentu dalam tabel. UPDATE sangat berguna ketika data sudah tersimpan namun harus direvisi atau disesuaikan dengan kondisi terbaru.

5. Mengapa perintah WHERE sangat penting saat UPDATE?

Tanpa WHERE, perintah UPDATE akan mengubah semua baris dalam tabel. Ini dapat menyebabkan kerusakan data besar-besaran. Oleh karena itu, WHERE adalah bagian penting agar perubahan hanya dilakukan pada baris yang tepat.

6. Apa fungsi perintah DELETE?

DELETE digunakan untuk menghapus baris data dari tabel. Seperti UPDATE, DELETE harus digunakan dengan hati-hati, terutama saat menghapus data berdasarkan kondisi tertentu.

7. Apa perbedaan DELETE dan TRUNCATE?

- DELETE menghapus baris tertentu berdasarkan kondisi.
- TRUNCATE menghapus **semua** data di dalam tabel dan lebih cepat daripada DELETE. TRUNCATE cocok untuk mengosongkan tabel tanpa menghapus struktur tabel.

8. Apa kegunaan SELECT?

SELECT adalah perintah paling penting dalam SQL karena digunakan untuk mengambil dan menampilkan data. SELECT dapat digabungkan dengan kondisi, pengurutan, pembatasan jumlah baris, dan fungsi lainnya untuk analisis data.

9. Mengapa LIMIT digunakan dalam SELECT?

LIMIT digunakan untuk membatasi jumlah baris yang ditampilkan. Ini sangat berguna saat tabel berisi banyak data, sehingga hasil menjadi lebih ringkas dan mudah dibaca. LIMIT juga mempercepat query.

10. Mengapa DML penting dalam aplikasi nyata?

Karena hampir semua aplikasi memerlukan:

- input data,
- edit data,
- hapus data,
- dan menampilkan data.

DML adalah pusat aktivitas database sehari-hari. Tanpa penguasaan DML, aplikasi tidak dapat berjalan dengan baik.

PRAKTIKUM 12 – JAWABAN TEORI (Operator SQL: LIKE, BETWEEN, IN, > < = !=)

1. Apa fungsi operator LIKE dalam SQL?

Operator LIKE digunakan untuk mencari data berdasarkan pola tertentu. Biasanya dipadukan dengan simbol wildcard:

- % untuk banyak karakter

- _ untuk satu karakter

LIKE sangat berguna untuk mencari data yang mengandung kata tertentu, misalnya mencari nama yang diawali huruf "A".

2. Mengapa LIKE banyak digunakan dalam pencarian data teks?

Karena LIKE memungkinkan pencarian yang fleksibel. Tidak harus sama persis, tetapi cukup mirip dengan pola tertentu. Ini membantu pengguna menemukan data meskipun tidak ingat seluruh kata secara lengkap.

3. Apa fungsi IN dalam SQL?

IN digunakan untuk mencari data yang cocok dengan salah satu dari beberapa nilai. IN sangat membantu ketika kita membandingkan satu kolom dengan banyak kemungkinan nilai tanpa harus menulis OR panjang.

Misalnya mencari barang dengan harga 5000, 10000, atau 15000.

4. Kapan operator IN lebih efisien daripada OR?

IN lebih rapi dan cepat ketika membandingkan satu kolom dengan banyak nilai.

Contoh buruk:

harga = 5000 OR harga = 10000 OR harga = 15000

Contoh baik:

harga IN (5000, 10000, 15000)

5. Apa fungsi operator BETWEEN?

BETWEEN digunakan untuk mencari data yang berada dalam rentang tertentu, misalnya tanggal atau angka.

Contoh: mencari data dengan nilai antara 60 sampai 80.

BETWEEN sangat mudah dibaca dan lebih efisien daripada menggunakan operator > dan < secara manual.

6. Apa perbedaan BETWEEN dan penggunaan > < secara manual?

BETWEEN lebih rapi dan mudah dipahami.

Contoh:

nilai BETWEEN 60 AND 80

setara dengan

nilai >= 60 AND nilai <= 80

Namun BETWEEN menyertakan batasnya (inclusive).

7. Mengapa operator perbandingan (=, >, <, >=, <=, !=) penting dalam SQL?

Operator ini memungkinkan SQL memilih data berdasarkan kondisi tertentu. Hampir semua query dengan WHERE menggunakan salah satu operator ini.

Tanpa operator perbandingan, kita tidak bisa melakukan filter data.

8. Apa kegunaan operator != atau <> dalam SQL?

Operator != atau <> digunakan untuk mencari data yang **tidak sama** dengan nilai tertentu.

Contoh: memilih semua mahasiswa kecuali yang jurusannya TI.

Operator ini berguna ketika kita ingin mengecualikan data tertentu.

9. Mengapa penggunaan operator sangat penting untuk laporan dan analisis?

Karena laporan dan analisis hampir selalu membutuhkan data yang sudah difilter.

Misalnya:

- barang dengan harga di atas 100000,
- transaksi bulan tertentu,
- mahasiswa dengan nilai tertentu.

Operator memberikan fleksibilitas agar data yang diambil tepat sasaran.

10. Apa yang terjadi jika operator digunakan tanpa WHERE?

Operator tidak akan berfungsi sama sekali. WHERE adalah bagian yang memberitahu SQL untuk melakukan filter.

Misalnya:

SELECT * FROM buku harga > 50000;  salah

SELECT * FROM buku WHERE harga > 50000;  benar

PRAKTIKUM 13 – JAWABAN TEORI (Fungsi Agregasi, DISTINCT, GROUP BY)

1. Apa yang dimaksud dengan fungsi agregasi?

Fungsi agregasi adalah fungsi yang digunakan untuk menghitung data dalam suatu kelompok atau keseluruhan tabel. Fungsi ini tidak bekerja per baris seperti single row function, tetapi menghasilkan satu nilai dari banyak baris.

Contoh fungsi agregasi:

- COUNT()
- SUM()
- AVG()
- MAX()
- MIN()

2. Mengapa fungsi agregasi penting dalam laporan data?

Karena laporan biasanya memerlukan nilai ringkasan, seperti total penjualan, jumlah mahasiswa, rata-rata nilai, atau nilai tertinggi.

Fungsi agregasi membantu merangkum data dalam bentuk yang mudah dipahami.

3. Apa fungsi COUNT()?

COUNT() digunakan untuk menghitung jumlah baris dalam tabel atau jumlah baris yang memenuhi kondisi tertentu.

Contoh: menghitung jumlah mahasiswa dalam jurusan tertentu.

COUNT bisa digunakan saat kita ingin tahu seberapa banyak data yang ada.

4. Apa fungsi SUM()?

SUM() digunakan untuk menjumlahkan seluruh nilai dari satu kolom numerik.

Cocok digunakan untuk laporan keuangan, total gaji, total transaksi, dan lain-lain.

5. Apa fungsi AVG()?

AVG() digunakan untuk mencari rata-rata nilai dari kolom tertentu.

Biasanya dipakai dalam laporan nilai mahasiswa, rata-rata harga barang, atau reviewing data statistik dalam perusahaan.

6. Apa fungsi MAX() dan MIN()?

- **MAX()** → mengambil nilai terbesar
- **MIN()** → mengambil nilai terkecil

Fungsi ini banyak dipakai dalam mencari nilai ekstrem seperti gaji tertinggi, nilai terendah, atau harga paling murah.

7. Apa fungsi DISTINCT dalam SQL?

DISTINCT digunakan untuk menghilangkan data yang duplikat sehingga hanya menampilkan nilai yang unik.

Misalnya ingin menampilkan daftar jurusan tanpa ada pengulangan.

8. Mengapa DISTINCT perlu digunakan dalam laporan?

DISTINCT memastikan bahwa informasi yang ditampilkan tidak berulang. Contohnya jika ada banyak mahasiswa dengan jurusan yang sama, DISTINCT akan menampilkan nama jurusan hanya satu kali saja.

9. Apa fungsi GROUP BY?

GROUP BY digunakan untuk mengelompokkan data berdasarkan satu atau beberapa kolom. Setelah dikelompokkan, kita bisa menggunakan fungsi agregasi untuk menghitung nilai per kelompok.

Misalnya:

- jumlah mahasiswa per jurusan,
- total gaji per divisi,
- rata-rata nilai per kelas.

10. Mengapa GROUP BY sering digunakan bersamaan dengan fungsi agregasi?

Karena GROUP BY menentukan bagaimana data dikelompokkan sebelum dihitung.

Tanpa GROUP BY, COUNT atau SUM hanya memberikan hasil untuk seluruh tabel.

Dengan GROUP BY, kita bisa menghitung total, jumlah, atau rata-rata untuk setiap kategori tertentu, misalnya berdasarkan jurusan atau kelas.