

# STRUKTUR QUERY LANGUAGE

## TYPE DATA SQL (GLOBAL TYPE)

Kategori	Tipe Data	Deskripsi Singkat
Angka (Numerik)	<b>INT</b>	Bilangan bulat (biasanya 4 byte), sangat umum untuk ID atau hitungan
	<b>TINYINT</b>	Bilangan bulat kecil (1 byte), cocok untuk status (0/1) atau boolean
	<b>BIGINT</b>	Bilangan bulat sangat besar (8 byte), cocok untuk jumlah besar atau waktu Unix
	<b>DECIMAL(p, s) / NUMERIC</b>	Bilangan desimal tetap, cocok untuk data keuangan
	<b>FLOAT / DOUBLE</b>	Bilangan pecahan desimal (floating point), cocok untuk data ilmiah atau statistik
Teks (String)	<b>VARCHAR(n)</b>	Teks dengan panjang variabel, paling umum dipakai untuk nama, email, dll
	<b>CHAR(n)</b>	Teks dengan panjang tetap, cocok untuk kode tetap (seperti kode pos, kode negara)
	<b>TEXT</b>	Teks panjang, cocok untuk deskripsi, komentar, atau artikel
Tanggal/Waktu	<b>DATE</b>	Format tanggal (YYYY-MM-DD)
	<b>DATETIME</b>	Format tanggal dan waktu (YYYY-MM-DD HH:MM:SS), sangat umum untuk mencatat waktu



Data:

		id	no	nama	nik	jk	alamat	divisi	level	jabatan	gaji	tanggal_input
<input type="checkbox"/>	Edit Copy Delete	1	A-001	RIFKY	MS-001	pria	Alamat: 2, Jl. RS. Fatmawati Raya No.64, RT.2/RW.5...	Big Data	senior	Manager	10000000	2025-05-28
<input type="checkbox"/>	Edit Copy Delete	2	A-002	HANAFI	MS-002	pria	Alamat: 2, Jl. RS. Fatmawati Raya No.64, RT.2/RW.5...	Backend Dev	middle	Staff	1000000	2025-05-27
<input type="checkbox"/>	Edit Copy Delete	3	A-003	PEPPY	MS-003	pria	Alamat: 2, Jl. RS. Fatmawati Raya No.64, RT.2/RW.5...	Backend Dev	basic	Staff	1000000	2025-05-29
<input type="checkbox"/>	Edit Copy Delete	4	A-004	Penny	MS-004	wanita	Alamat: 2, Jl. RS. Fatmawati Raya No.64, RT.2/RW.5...	Frontend Dev	basic	Staff	1000000	2025-05-25
<input type="checkbox"/>	Edit Copy Delete	5	A-005	BOY	MS-005	pria	Alamat: 2, Jl. RS. Fatmawati Raya No.64, RT.2/RW.5...	Frontend Dev	middle	Staff	1000000	2025-05-30
↑	<input type="checkbox"/> Check all	With selected:		Edit	Copy	Delete	Export					

## Latihan

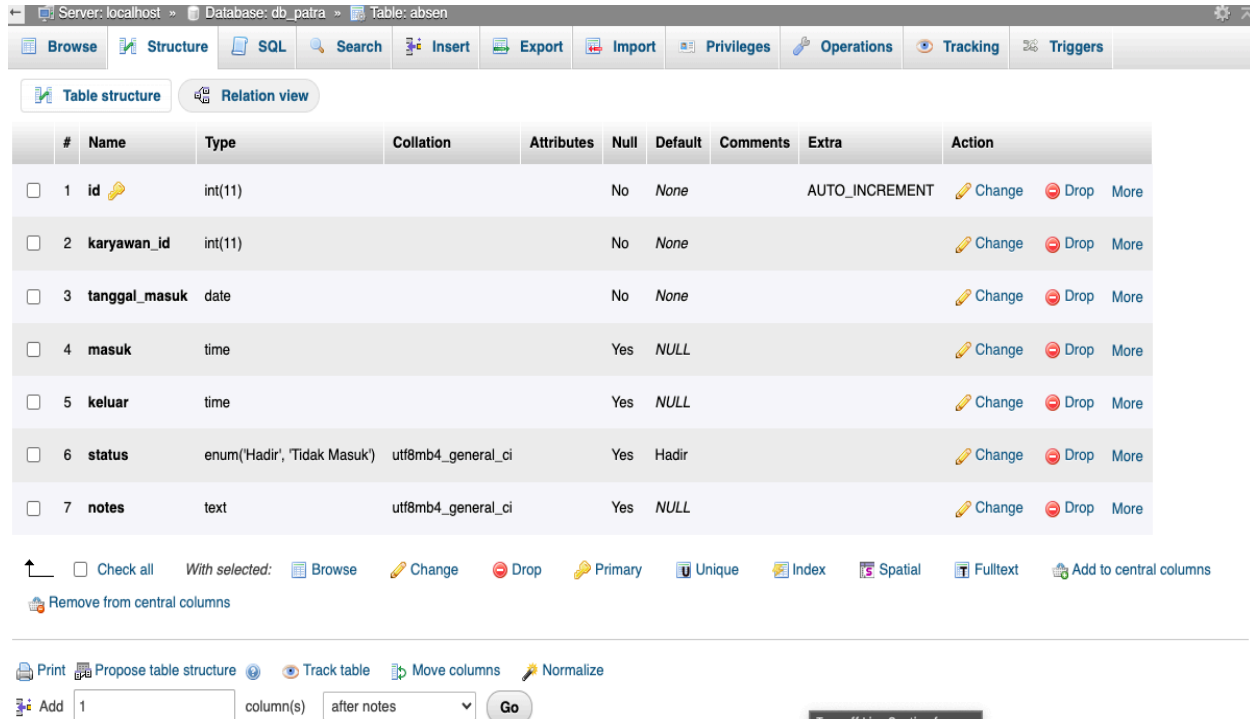
1. Buatlah database patra dan tabel karyawan
2. Buatlah filter data berdasarkan nama depan yang berawalan huruf "R"
3. Buatlah filter data berdasarkan nama belakang yang berawalan huruf "Y"
4. Buatlah urutan data berdasarkan a - z
5. Buatlah urutan data berdasarkan z - a
6. Buatlah query data antara 1 dan 3
7. Buat Konversi Tanggal yang dari angka menjadi huruf supaya lebih dipahami
8. Buatlah Konversi bulan menjadi huruf supaya lebih mudah di pahami
9. Buat tanggal\_input menjadi tanggal indonesia
10. Buatlah selisih antara 2 tanggal
11. Buatlah query untuk menghilangkan penggandaan pada data level
12. Buat lah query mencari gaji terbesar
13. Buat lah query mencari gaji terkecil
14. Buat lah query mencari gaji rata-rata
15. Buat lag query mencari jumlah data gaji
16. Buat lah query untuk menampilkan jika wanita mendapatkan cuti hamil sedangkan pria tidak
17. Buatlah query untuk mengubah nama menjadi huruf besar
18. Buatlah query untuk mengubah nama menjadi huruf kecil
19. Buatlah query untuk memotong nama antara 1 dan 3 huruf
20. Buatlah query untuk menggabungkan huruf

**Jawaban Soal No.2**

- 2. SELECT \* FROM `karyawan` where nama LIKE 'R%';**
- 3. SELECT \* FROM `karyawan` where nama LIKE '%Y';**
- 4. SELECT \* FROM `karyawan` ORDER BY ID ASC**
- 5. SELECT \* FROM `karyawan` ORDER BY ID DESC**
- 6. SELECT \* FROM `karyawan` WHERE id BETWEEN 1 AND 3**
- 7. SELECT DAYNAME(tanggal\_input) as tanggal\_huruf FROM `karyawan`**
- 8. SELECT MONTHNAME(tanggal\_input) as bulan\_huruf FROM `karyawan`;**
- 9. SELECT DATE\_FORMAT(tanggal\_input, '%d-%M-%Y') as tanggal\_full\_huruf FROM `karyawan`;**
- 10. SELECT DATEDIFF(tanggal\_input, NOW()) as tanggal\_selesih FROM `karyawan`;**
- 11. SELECT DISTINCT(level) as hilangkan\_penggandaan FROM `karyawan`;**
- 12. SELECT MAX(gaji) as gaji\_teratas FROM `karyawan`**
- 13. SELECT MIN(gaji) as gaji\_terbawah FROM `karyawan`**
- 14. SELECT AVG(gaji) as gaji\_rata\_rata FROM `karyawan`**
- 15. SELECT COUNT(gaji) as jumlah\_gaji FROM `karyawan`**
- 16. SELECT IF(jk = 'wanita', "cuti hamil", "tidak dapat cuti") as cuti FROM karyawan;**
- 17. SELECT UPPER(nama) as nama\_besar FROM `karyawan`**
- 18. SELECT LOWER(nama) as nama\_kecil FROM `karyawan`**
- 19. SELECT SUBSTR(nama, 1, 2) as potong\_text FROM `karyawan`**
- 20. SELECT CONCAT(nama, "-", nik) AS gabung\_text FROM karyawan;**

## 2 = Case Study Data Karyawan Dan Absen

### Join Data Antar Tabel



The screenshot shows the phpMyAdmin interface for a database named 'db\_patra'. The 'absen' table is selected, and its structure is displayed in 'Table structure' view. The table has 7 columns: 'id' (int(11), primary key, AUTO\_INCREMENT), 'karyawan\_id' (int(11)), 'tanggal\_masuk' (date), 'masuk' (time), 'keluar' (time), 'status' (enum('Hadir', 'Tidak Masuk'), utf8mb4\_general\_ci), and 'notes' (text, utf8mb4\_general\_ci). Below the table structure, there are various tools and options for managing the table, including 'Check all', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', 'Add to central columns', 'Remove from central columns', 'Print', 'Propose table structure', 'Track table', 'Move columns', 'Normalize', and a 'Go' button.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	karyawan_id	int(11)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	tanggal_masuk	date			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
4	masuk	time			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
5	keluar	time			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
6	status	enum('Hadir', 'Tidak Masuk')	utf8mb4_general_ci		Yes	Hadir			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
7	notes	text	utf8mb4_general_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

### Latihan

1. Buatlah tabel di atas
2. Gabungkan lah 2 tabel antara karyawan dan absen untuk mendapatkan hasil karyawan yang masuk saja gambaran seperti berikut :
  - a. Nama karyawan, Tanggal Masuk, Masuk , Keluar dan Status Kehadiran.
3. Gabungkan lah 2 tabel antara karyawan dan absen untuk mendapatkan hasil karyawan yang tidak masuk saja gambaran seperti berikut :
  - a. Nama karyawan, Tanggal Masuk, Masuk , Keluar dan Status Kehadiran.
4. Gabungkan lah 2 tabel untuk mendapatkan jumlah karyawan yang masuk dalam waktu sebulan
5. Gabungkan lah 2 tabel untuk mendapatkan jumlah karyawan yang tidak masuk dalam waktu sebulan

6. Gabungkan lah 2 tabel untuk di jadikan sebagai data grafik untuk menampilkan data absen karyawan di bulan yang sedang berjalan

Jawaban

2. **SELECT \* FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE absen.status = 'hadir' GROUP BY karyawan.id;**

3. **SELECT \* FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE absen.status = 'tidak\_hadir' GROUP BY karyawan.id;**

4. **SELECT \*,COUNT(absen.status) as jml FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE absen.status = 'hadir' AND MONTH(karyawan.tanggal\_input) = MONTH(NOW()) GROUP BY karyawan.id;**

5.**SELECT \*,COUNT(absen.status) as jml FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE absen.status = 'tidak\_hadir' AND MONTH(karyawan.tanggal\_input) = MONTH(NOW()) GROUP BY karyawan.id;**

6.**SELECT karyawan.id,karyawan.nama, absen.status, karyawan.tanggal\_input, absen.masuk, absen.keluar, COUNT(absen.status) as jml FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE MONTH(karyawan.tanggal\_input) = MONTH(NOW()) GROUP BY karyawan.id;**

### 3 = Case Study Stored Prosedure

Create new routine ☐

Details

Routine name

filter\_user

Type

PROCEDURE ▾

Parameters

Direction	Name	Type	Length
IN ▾	filter_username	VARCHAR ▾	255

Add parameter

Definition

```
1 SELECT * FROM users WHERE username = filter_username;
```

Is deterministic

☐

Go

Close

Selanjutnya kita bisa execute dengan memanggil fungsi tersebut dengan command, CALL, seperti contoh berikut :

Call filter\_user()

Cara Lain Untuk Mengolah data stored procedure ini bisa dengan menggunakan full command line dengan mengetikkan perintah seperti ini :

```
DELIMITER //  
CREATE PROCEDURE GetKaryawanByID(IN karyawanID INT)  
BEGIN  
    SELECT * FROM karyawan WHERE id_karyawan = karyawanID;  
END //  
DELIMITER ;
```

Selanjutnya sama seperti di atas dengan memanggil fungsi tersebut dengan call , “call GetKaryawanByID(1)”;

#### Latihan

1. Buatlah Stored Prosedured Diatas dengan menggunakan command line sesuai dari code diatas.
2. Panggilah dengan nama fungsi , bisa disesuaikan dengan nama fungsi diatas



# SOAL SOAL SQL LANJUTAN

## 1. Total Gaji per Departemen

Tampilkan nama departemen dan total gaji yang dibayarkan ke semua karyawan di masing-masing departemen, urutkan dari yang paling besar.

## 2. Karyawan Berdasarkan Lama Kerja

Tampilkan nama karyawan, nama departemen, dan jumlah tahun kerja (dibulatkan ke bawah) bagi karyawan yang sudah bekerja lebih dari 5 tahun.

## 3. Rata-rata Gaji Berdasarkan Jenis Kelamin

Hitung rata-rata gaji karyawan berdasarkan jenis kelamin (L atau P), dan tampilkan dalam urutan rata-rata tertinggi.

## 4. Karyawan Tertua di Setiap Departemen

Tampilkan satu karyawan tertua dari masing-masing departemen beserta nama dan tanggal lahirnya.

## 5. Jumlah Karyawan yang Ulang Tahun Bulan Ini

Hitung berapa jumlah karyawan yang berulang tahun di bulan yang sama dengan bulan saat ini (`MONTH(NOW())`).

## Jawaban

1. SELECT d.nama\_departemen, SUM(k.gaji) AS total\_gaji FROM karyawan k LEFT JOIN departemen d ON k.id\_departemen = d.id\_departemen GROUP BY d.nama\_departemen ORDER BY total\_gaji DESC;

2. SELECT k.nama, d.nama\_departemen,  
FLOOR(DATEDIFF(CURDATE(), k.tanggal\_masuk) / 365) AS tahun\_kerja  
FROM karyawan k  
JOIN departemen d ON k.id\_departemen = d.id\_departemen  
WHERE DATEDIFF(CURDATE(), k.tanggal\_masuk) / 365 > 5;

3. SELECT jk, ROUND(AVG(gaji), 2) AS rata\_gaji  
FROM karyawan  
GROUP BY jk  
ORDER BY rata\_gaji DESC;

4. SELECT k.id, k.nama, d.nama\_departemen, k.tanggal\_lahir  
FROM karyawan k  
JOIN departemen d ON k.id\_departemen = d.id\_departemen  
WHERE k.tanggal\_lahir = (  
SELECT MIN(k2.tanggal\_lahir)  
FROM karyawan k2  
WHERE k2.id\_departemen = k.id\_departemen  
);

5. SELECT COUNT(\*) AS jumlah\_ulang\_tahun  
FROM karyawan  
WHERE MONTH(tanggal\_lahir) = MONTH(CURDATE());

# UJIAN FINAL SQL QUERY

## Case Studi, E-Commerce.

1. Normalization
  - a. Database e-commerce
    - i. Table product
    - ii. Table orders
    - iii. Table Users
    - iv. Product\_category , next
2. Database Relation Table
3. Xampp Implementation
4. Examination

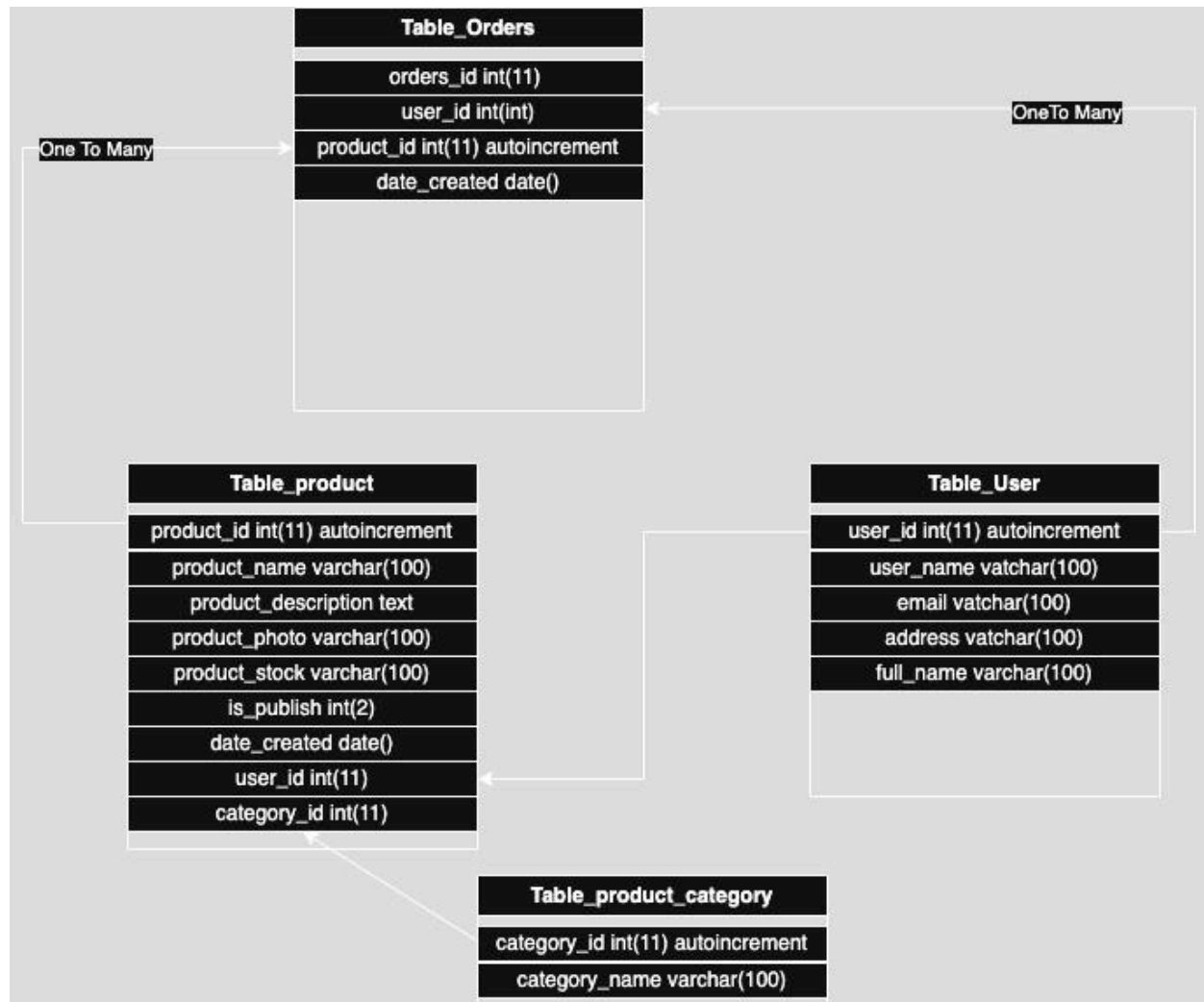
### 1. Normalization

Adalah Pembuatan Relasi tabel sehingga tidak ada lagi redundan data atau duplikasi tabel antar tabel

Latihan :

- a. Buatlah normalisasi dari database e-commerce
- b. Buatlah Tabel Beserta Relasi Antar Field nya
- c. Design Normalisasi menggunakan  
="https://app.diagrams.net/"=
- d. Jika Ada kekurangan tabel bisa ditambahkan atau di modify
- e. next

## 2. Class Diagram



### **Latihan**

- 1. Buat lah class diagram diatas dengan menggunakan sql**
- 2. Setelah di buat tolong tampilkan semua list produk**
- 3. Tampilkan produk berdasarkan kategorinya**
- 4. Tampilkan produk yang paling banyak di beli**
- 5. Tampilkan produk yang paling sedikit di beli**
- 6. Tampilkan pengguna yang sering melakukan pemesanan**
- 7. Tampilkan list pengguna secara keseluruhan**
- 8. Create data per masing masing table minimal 5 data dengan validasi dari pengajar**
- 9. Update data per masing masing table dengan validasi dari pengajar**
- 10. Delete data satuan table dengan validasi dari pengajar**