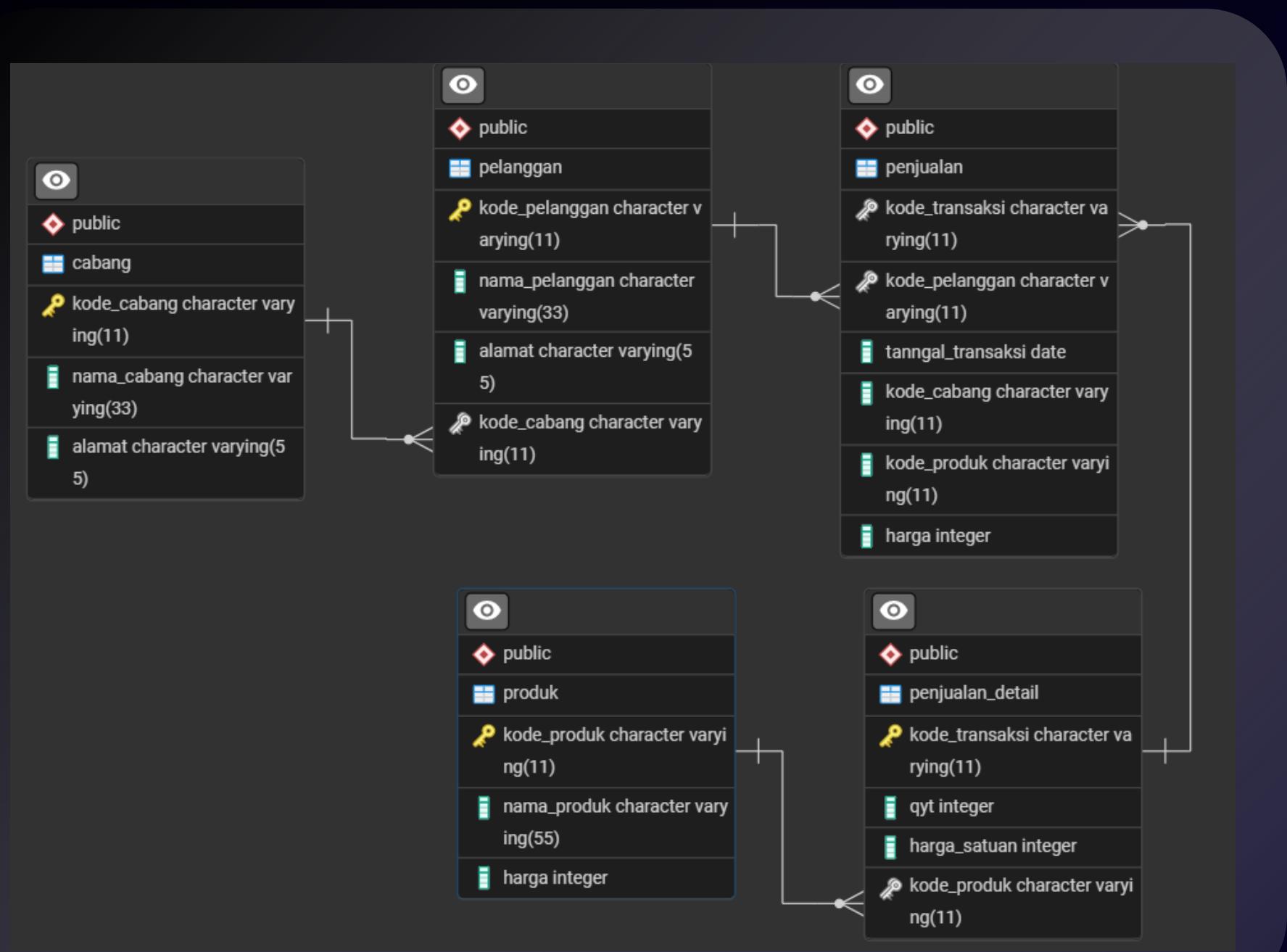


Data Engineer with SQL

GitHub :

<https://github.com/sefzaa/Data-Engginner-SQL.git>

CREATE TABLE



```
CREATE TABLE PENJUALAN (
    kode_transaksi varchar (11),
    kode_pelanggan varchar (11),
    tanggal_transaksi date,
    kode_cabang varchar (11),
    kode_produk varchar (11),
    harga int,
    FOREIGN KEY (kode_transaksi) REFERENCES PENJUALAN_DETAIL(kode_transaksi),
    FOREIGN KEY (kode_pelanggan)
    REFERENCES PELANGGAN(kode_pelanggan)
);
```

Menampilkan daftar produk yang memiliki harga antara 50.000 and 150.000

	kode_produk [PK] character varying (11)	nama_produk character varying (55)	harga integer
1	prod-001	Kotak Pensil DQLab	60500
2	prod-002	Flashdisk DQLab 64 GB	55000
3	prod-003	Gift Voucher DQLab 100rb	100000
4	prod-006	Pulpen Multifunction + Laser DQLab	92500
5	prod-009	Buku Planner Agenda DQLab	92000
6	prod-010	Sticky Notes DQLab 500 sheets	55000

```
select * from produk where harga >= 50000  
and harga <= 150000;
```

Menampilkan semua produk yang mengandung kata Flashdisk.

```
SELECT * FROM produk  
WHERE nama_produk LIKE '%Flashdisk%';
```

	kode_produk [PK] character varying (11)	nama_produk character varying (55)	harga integer
1	prod-002	Flashdisk DQLab 64 GB	55000
2	prod-004	Flashdisk DQLab 32 GB	40000

Menampilkan hanya nama-nama pelanggan yang hanya memiliki gelar-gelar berikut: S.H, Ir. dan Drs.

kode_pelanggan [PK] character varying (11)	nama_pelanggan character varying (33)	alamat character varying (55)
cust0001	Eva Novianti, S.H.	Vila Sempilan, No. 67
cust0009	Ir. Ita Nugraha	Perumahan Sagitarius, Gang Kelapa No. 6
cust0010	Djoko Wardoyo, Drs.	Bukit Pintar Data, Blok A1 No. 1

```
SELECT kode_pelanggan, nama_pelanggan, alamat
      from pelanggan
 WHERE nama_pelanggan LIKE '%S.H%'
   OR nama_pelanggan LIKE '%Ir.%'
   OR nama_pelanggan LIKE '%Drs.%';
```

Menampilkan nama pelanggan dan urutkan hasilnya berdasarkan kolom nama_pelanggan dari yang terkecil ke yang terbesar (A ke Z).

```
SELECT  
nama_pelanggan  
FROM  
pelanggan  
ORDER BY  
nama_pelanggan ASC;
```

	nama_pelanggan character varying (33) 
1	Agus Cahyono
2	Djoko Wardoyo, Drs.
3	Eva Novianti, S.H.
4	Heidi Goh
5	Ir. Ita Nugraha
6	Irwan Setianto
7	Jokolono Sukarman
8	Maria Sirait
9	Tommy Sinaga
10	Unang Handoko

Menampilkan urutan nama pelanggan dari yang terkecil ke yang terbesar (A ke Z), namun gelar tidak boleh menjadi bagian dari urutan.

```
SELECT
    nama_pelanggan
FROM
    pelanggan
ORDER BY
    TRIM (REGEXP_REPLACE
(nama_pelanggan,
'^([Drs\.|Ir\.|S\.H\.,?]\s*)', '')) ASC;
```

	nama_pelanggan
1	Agus Cahyono
2	Djoko Wardoyo, Drs.
3	Eva Novianti, S.H.
4	Heidi Goh
5	Irwan Setianto
6	Ir. Ita Nugraha
7	Jokolono Sukarman
8	Maria Sirait
9	Tommy Sinaga
10	Unang Handoko

Tampilkan nama pelanggan yang memiliki nama paling panjang.

	nama_pelanggan character varying (33)	
1	Djoko Wardoyo, Drs.	

```
SELECT
    nama_pelanggan
FROM
    pelanggan
WHERE
    LENGTH (nama_pelanggan) =
        (SELECT MAX (LENGTH
    (nama_pelanggan) )
        FROM
    pelanggan);
```

Menampilkan nama paling panjang pada row atas,
dan nama paling pendek pada row setelahnya

```
SELECT nama_pelanggan FROM pelanggan WHERE
LENGTH(nama_pelanggan) IN (
(SELECT MAX(LENGTH(nama_pelanggan)) FROM
pelanggan),
(SELECT MIN(LENGTH(nama_pelanggan)) FROM
pelanggan)
)
ORDER BY LENGTH(nama_pelanggan) DESC;
```

	kode_produk [PK] character varying (11)	nama_produk character varying (55)	harga integer
1	prod-002	Flashdisk DQLab 64 GB	55000
2	prod-004	Flashdisk DQLab 32 GB	40000

Menampilkan produk yang paling banyak terjual dari segi kuantitas

```
SELECT
    produk.kode_produkt,
    produk.nama_produkt,
    SUM(penjualan_detail.qyt) AS total_qty
FROM
    produk
INNER JOIN penjualan_detail
    ON produk.kode_produkt = penjualan_detail.kode_produkt
GROUP BY
    produk.kode_produkt,
    produk.nama_produkt
HAVING
    SUM(penjualan_detail.qyt) > 2;
```

kode_produkt	nama_produkt	total_qty
[PK] character varying (11)	character varying (55)	bigint
prod-004	Flashdisk DQLab 32 GB	6
prod-008	Gantungan Kunci DQLab	6

Menampilkan pelanggan yang paling banyak menghabiskan uangnya untuk belanja

```
SELECT
    penjualan.kode_pelanggan,
    pelanggan.nama_pelanggan,
    SUM(penjualan_detail.qyt * penjualan_detail.harga_satuan) AS total_harga
FROM
    pelanggan
INNER JOIN penjualan USING (kode_pelanggan)
INNER JOIN penjualan_detail USING (kode_transaksi)
GROUP BY
    penjualan.kode_pelanggan,
    pelanggan.nama_pelanggan
ORDER BY
    total_harga DESC
LIMIT 1 ;
```

kode_pelanggan	nama_pelanggan	total_harga
character varying (11)	character varying (33)	bigint
cust0007	Agus Cahyono	480000

Menampilkan daftar pelanggan yang belum pernah melakukan transaksi

```
SELECT  
kode_pelanggan,  
nama_pelanggan,  
alamat  
FROM  
pelanggan  
WHERE  
kode_pelanggan  
NOT IN  
(SELECT kode_pelanggan  
FROM  
penjualan);
```

kode_pelanggan [PK] character varying (11)	nama_pelanggan character varying (33)	alamat character varying (55)
cust0002	Heidi Goh	Ruko Sawit Permai 72 No. 1
cust0005	Tommy Sinaga	Avatar Village, Blok C8 No. 888
cust0006	Irwan Setianto	Rukan Gunung Seribu, Blok 01 - No. 1
cust0009	Ir. Ita Nugraha	Perumahan Sagitarius, Gang Kelapa No. 6
cust0010	Djoko Wardoyo, Drs.	Bukit Pintar Data, Blok A1 No. 1

Menampilkan transaksi-transaksi yang memiliki jumlah item produk lebih dari 1 jenis produk

```
SELECT
    pj.kode_transaksi, pj.kode_pelanggan, pl.nama_pelanggan,
    pj.tanggal_transaksi,
    COUNT(pd.qyt) AS jumlah_detail
FROM
    penjualan pj
INNER JOIN pelanggan pl ON pj.kode_pelanggan = pl.kode_pelanggan
INNER JOIN penjualan_detail pd ON pj.kode_transaksi = pd.kode_transaksi
GROUP BY
    pj.kode_transaksi,
    pj.kode_pelanggan,
    pl.nama_pelanggan,
    pj.tanggal_transaksi
HAVING
    COUNT(pd.qyt) > 1;
```

kode_transaksi character varying (11)	kode_pelanggan character varying (11)	nama_pelanggan character varying (33)	tanggal_transaksi date	jumlah_detail bigint
tr-0005	cust0003	Unang Handoko	2019-06-09	3
tr-0001	cust0007	Agus Cahyono	2019-06-07	4
tr-0006	cust0008	Maria Sirait	2019-06-09	2
tr-0002	cust0001	Eva Novianti, S.H.	2019-06-07	3

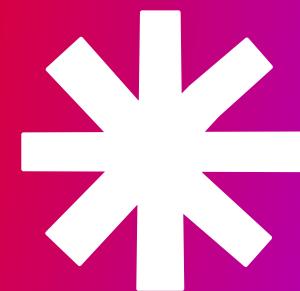
Thank You

TASK

At the core of programming are concepts like variables, loops, functions, and conditional statements. These elements form the logic that drives a program, allowing it to make decisions, process data, and perform tasks. Mastering these basics is essential for programmers to create reliable and efficient code.

Collaboration in Programming

Working
Together in Code



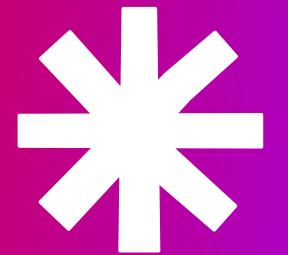
Programming often requires collaboration, especially in large projects. Developers use version control tools like Git to manage code, allowing multiple programmers to work on the same project seamlessly.



Testing and debugging are crucial steps in programming. Testing verifies that code functions as expected, while debugging identifies and resolves errors. By carefully testing and refining code, programmers ensure their applications are reliable, user-friendly, and free of bugs, enhancing the end-user experience.

Testing and Debugging

Working
Together in Code



AI, Automation, and Beyond

The Future of Programming

The future of programming is evolving with advancements in artificial intelligence, machine learning, and automation. As programming languages and tools become more sophisticated, programmers can tackle complex tasks more efficiently. These developments open up new opportunities for innovation, shaping the way we interact with technology.

Thynk Unlimited