A DATABASE DESIGN FOR "DAILYNEST"

DailyNest is a store that offers all your daily essentials with the best quality at affordable prices. We provide a wide range of products, from groceries and household items to personal care needs, all in one place to make your life easier. To support efficient operations, we need a well-structured database to manage product inventory, sales transactions, customer information, and stock levels. **This database is essential to ensure product availability,** minimize record-keeping errors, facilitate sales analysis, and provide a better shopping experience for our customers.

by: Nalini Mahastuti Panunjul; Doria Vika Santana; Nur Ismi Maharani

ACCESS THE CODE HERE

"Click" to go to the GitHub page.

A businessman owns **"DailyNest"** that sells daily essentials such as food, beverages, toiletries, cleaning supplies, and similar items.

10TH YEAR

In the tenth year, the store owner plans to **overhaul the business by improving product category management to make workflows more efficient and better organize warehouse** and supply chain management.

Based on the owner's experience, many customers have **complained about the slow response from the sales team in providing basic information** about products, such as stock, color options, and specifications.

This issue is caused by a poorly organized data recording system. Therefore, in addition to the physical renovation of the store, the business owner also **plans to develop a database system to improve operational efficiency in the store.**

01. PROBLEM OVERVIEW

DRAW.IO

Creating UML diagrams, flowcharts, and other visualizations, supporting process and system design, including database schemas.

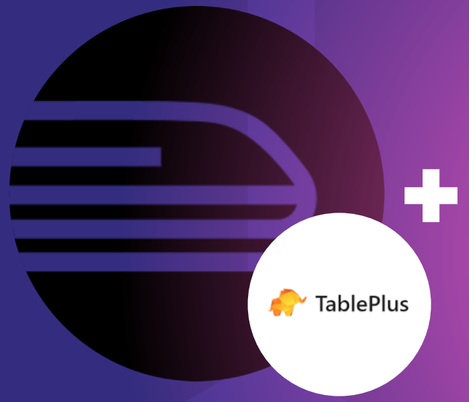
MYSQL

Open-source database system widely used in web applications and the LAMP stack for managing and manipulating data with SQL.

WORKBENCH

User-friendly **visual tool for database design,** SQL development, server management, and performance monitoring.

RAILWAY



Railway is a platform for **deploying and managing databases.**

02. TOOLS

UNF (Unnormalized)

Table 1

id_product	product	price	stock	category	id_customer	phone	name	address	email	id_transaction	transaction_date	quantity_order	total_price
...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...

- Data resides in a single table without any specific rules or structure.
- The table contains poorly defined columns, duplication, and non-atomic (non-singular) values.

1NF

Table 1

id_product	product	price	stock	category	id_customer	phone	name	address	email	id_transaction	transaction_date	quantity_order
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...

- Each column has a unique name.
- Each cell contains a single atomic value.

2NF

Table Product

id_product	product	price	stock	id_category	category
...	...	...	...	...	...
...	...	...	...	...	...

Table Customer

id_customer	phone	name	address	email
...	...	...	...	...
...	...	...	...	...

Table Transaction

id_transaction	transaction_date	id_customer	id_product	quantity_order
...	...	...	...	...
...	...	...	...	...

- The table must be in 1NF.
- Each non-primary key attribute must be fully dependent on the entire primary key (no partial dependencies).

3NF

Table Product

id_product	id_category	product	price	stock
...	...	...	...	...
...	...	...	...	...

Table Customer

id_customer	phone	name	address	email
...	...	...	...	...
...	...	...	...	...

- No non-primary key attribute should have a transitive dependency on the primary key.

Table Category

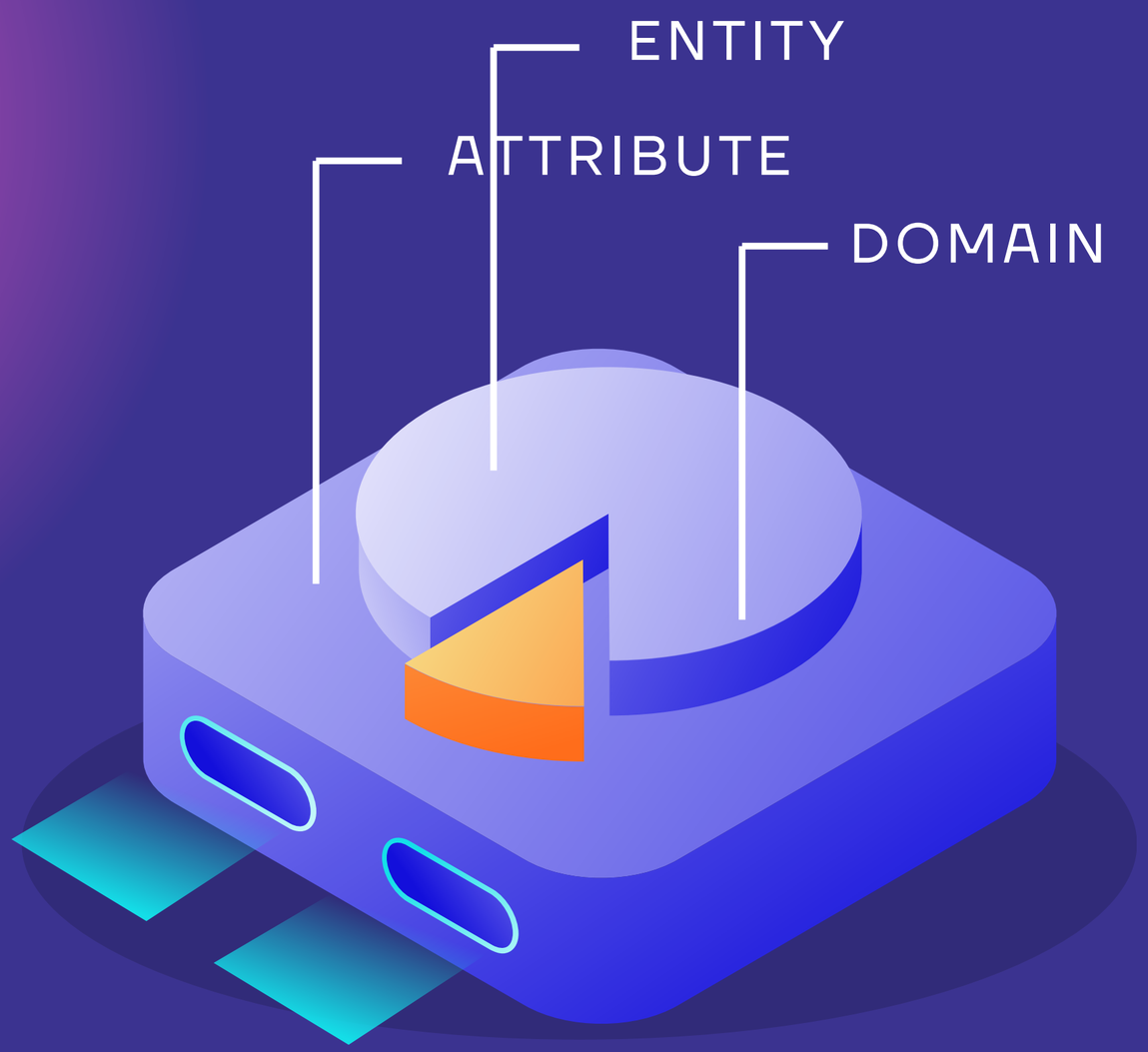
id_category	category
...	...
...	...

Table Transaction

id_transaction	transaction_date	id_customer
...	...	...
...	...	...

Table Detail Transaction

id_transaction	id_product	quantity_order
...	...	...
...	...	...



Entity: Category

Attribute:
- id_category
- category

Domain:
- varchar(225)
- varchar(225)

Entity: Product

Attribute:
- id_product
- id_category
- product
- price
- stock

Domain:
- varchar(225)
- varchar(225)
- varchar(225)
- int
- int

Entity: Customer

Attribute:
- id_customer
- phone
- name
- address
- email

Domain:
- varchar(225)
- varchar(15)
- varchar(225)
- varchar(225)
- varchar(225)

Entity: Transaction

Attribute:
- id_transaction
- id_customer
- transaction_date

Domain:
- varchar(225)
- date
- varchar(225)

Entity: Detail Transaction

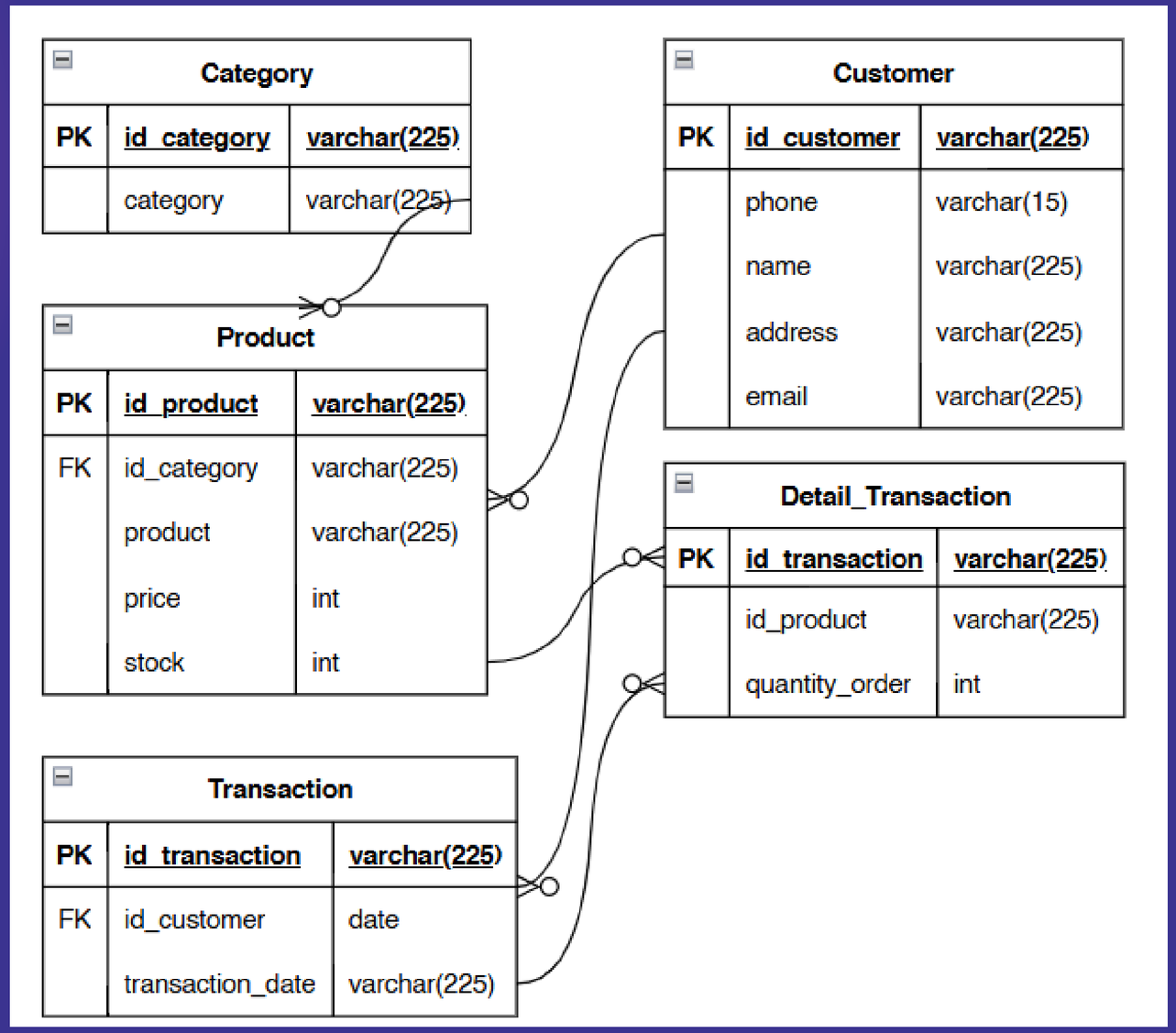
Attribute:
- id_transaction
- id_product
- quantity_order

Domain:
- varchar(225)
- varchar(225)
- int

Primary Key

Foreign Key

03. DATABASE DESIGN | DATABASE COMPONENTS



Customer → Product
Relationship: One-to-Many
Reason: A customer can purchase multiple products.

Category → Product
Relationship: One-to-Many
Reason: A single category can contain multiple products

Transaction → Detail_Transaction
Relationship: One-to-Many
Reason: A single transaction can contain multiple detailed product entries.

Customer → Transaction
Relationship: One-to-Many
Reason: A single customer can make multiple transactions over time.

Product → Detail_Transaction
Relationship: One-to-Many
Reason: A single product can appear in multiple detailed transaction entries.

03. DATABASE DESIGN | ERD & RELATIONSHIP

QUERY

```
-- Feature 1: Show Information --
select * from category;
select * from product;
select * from customer;
select * from transaction;
select * from detail_transaction;
```

The first feature is designed using an SQL query to **display information** about items, buyers, and the transactions of item purchases.

DETAIL TRANSACTION

id_transaction	id_product	quantity_orde
TR6	MK0004	2
TR6	LN0004	1
TR7	PKM0002	1
TR7	PKM0003	1
TR7	PKM0007	1
TR8	ATK0001	1
TR8	ATK0002	2
TR8	ATK0003	4
TR8	ATK0004	2
TR8	ATK0005	2
TR9	MK0006	1
TR10	OB0005	1
TR10	OB0002	1

id_transaction	id_product	quantity_order
TR1	ATK0005	2
TR1	ATK0006	3
TR1	ATK0004	3
TR2	SB0003	1
TR2	SB0004	1
TR2	SB0002	5
TR2	SB0001	1
TR3	ATK0005	10
TR4	PBB0003	2
TR4	PKM0001	2
TR4	PD0002	1
TR4	PKM0005	2
TR4	PBB0005	6

TRANSACTION

id_transaction	id_customer	transaction_date
TR1	P4	2023-12-05
TR10	P1	2023-12-10
TR2	P11	2023-12-06
TR3	P13	2023-12-06
TR4	P14	2023-12-06
TR5	P6	2023-12-07
TR6	P3	2023-12-07
TR7	P6	2023-12-08
TR8	P6	2023-12-09
TR9	P5	2023-12-09

CUSTOMER

id_customer	phone	name	address	email
P12	082851840730	Utami Fitriyani	Jl. Resi No. 143	utamifitriyani143@gmail.com
P13	089273068952	Zaqi Brahmayudha	Jl. Meliala No. 157	zaqibrahmayudha157@gmail.com
P14	088017407386	Afghan Sandyka	Jl. Wijawati No. 162	afghansandyka162@gmail.com
P15	083810683306	Nazhif Titis	Jl. Faiza No. 170	nazhiftitis170@gmail.com
P2	082345678901	Gita Anwar	Jl. Lukman No. 181	gitaanwar181@gmail.com
P3	083456789012	Ayuningtyas Octaviana	Jl. Diva No. 188	ayuningtyasoctaviana188@gmail.com
P4	084567890123	Meilisa Nurhayati	Jl. Aidifa No. 196	meilisanurhayati196@gmail.com
P5	085678901234	Sri Syahirah	Jl. Putri No. 08	srisyahirah08@gmail.com
P6	086789012345	Aflah Seto	Jl. Anindita No. 15	aflahseto15@gmail.com
P7	087890123456	Salwa Rahmannia	Jl. Razan No. 23	salwarahmannia23@gmail.com
P8	088901234567	Bramantika Nataniel	Jl. Alfahima No. 37	bramantikanataniel37@gmail.com
P9	089012345678	Nasaruddin Haq	Jl. Mirzadiniyah No....	nasaruddinhaq42@gmail.com

PRODUCT

id_product	id_category	product	price	stock
ATK0001	ATK	Kertas Folio	47000	42
ATK0002	ATK	Kertas Kado	1500	42
ATK0003	ATK	Joyko Eraser EB 30	1800	49
ATK0004	ATK	Tip-x	25889	90
ATK0005	ATK	Buku Tulis SIDU 58 Lembar	39000	71
ATK0006	ATK	Pena	2750	52
LN0001	LN	LPG 3 kg	12750	86
LN0002	LN	SWALLOW SERI	15000	47
LN0003	LN	Baygon cair reffil 175 ml	13506	60
LN0004	LN	PASEO TISSUE	10650	41
LN0005	LN	LILIN CAP PAUS	6750	86
LN0006	LN	TUSUK GIGI	20500	70
MK0001	MK	Biskuat Coklat 20 gr	3120	78
MK0002	MK	Biskuat Energi 22,5 gr	1500	84
MK0003	MK	Coki Coki	20000	49
MK0004	MK	DUA KELINCI KACANG 25...	23000	103
MK0005	MK	HELLO PANDA	6995	85
MK0006	MK	KWACI BUNGA MATAHARI	9995	34

CATEGORY

id_category	category
ATK	Alat Tulis
LN	Lainnya
MK	Makanan
MN	Minuman
OB	Obat-obatan
PBB	Perlengkapan Bayi dan Balita
PD	Perlengkapan Dapur
PKM	Perlengkapan Kamar Mandi
SB	Sembako

04. PROTOTYPE | FEATURE 1: SHOW INFORMATIONS

QUERY

```
-- Feature 2: Update Stok [Customer Beli] --
delimiter //
create procedure beli_barang (in input_id_product varchar(225), in input_beli int)
begin
    declare stok_terakhir int;

    -- dapetin current stock dari item yang dipilih --
    select stock
    into stok_terakhir
    from product
    where id_product = input_id_product;

    -- update stock setelah beli --
    update product
    set stock = stok_terakhir - input_beli
    where id_product = input_id_product;

    -- show info setelah update stock --
    select
        product.id_product,
        product.product,
        product.price,
        product.stock
    from detail_transaction
    join product on detail_transaction.id_product = product.id_product
    order by product.id_product;
end //
delimiter ;
call beli_barang('ATK0002', 10);
```

FUNCTIONALITY & OUTPUT

```
select * from product;
```

id_product	id_category	product	price	stock
ATK0001	ATK	Kertas Folio	47000	42
ATK0002	ATK	Kertas Kado	1500	14
ATK0003	ATK	Joyko Eraser EB 30	1800	49
ATK0004	ATK	Tip-x	25889	90
ATK0005	ATK	Buku Tulis SIDU 58 Lembar	39000	71

```
call beli_barang('ATK0002', 10);
```

id_product	product	price	stock
ATK0001	Kertas Folio	47000	42
ATK0002	Kertas Kado	1500	4
ATK0003	Joyko Eraser EB 30	1800	49
ATK0004	Tip-x	25889	90
ATK0004	Tip-x	25889	90

04. PROTOTYPE | FEATURE 2: UPDATE STOCK [CUSTOMER]

QUERY

```
-- Feature 3: Update Stok [Admin Add Barang] --
delimiter //
create procedure nambah_barang(in input_id_product varchar(225), in input_update int)
begin
    declare stok_terakhir int;

    -- dapetin current stock dari item yang dipilih --
    select stock
    into stok_terakhir
    from product
    where id_product = input_id_product;

    -- update stock setelah beli --
    update product
    set stock = stok_terakhir + input_update
    where id_product = input_id_product;

    -- show info setelah update stock --
    select
        product.id_product,
        product.product,
        product.price,
        product.stock
    from detail_transaction
    join product on detail_transaction.id_product = product.id_product
    order by product.id_product;
end //
delimiter ;
call nambah_barang('ATK0002', 2);
```

FUNCTIONALITY & OUTPUT

```
select * from product;
```

id_product	id_category	product	price	stock
ATK0001	ATK	Kertas Folio	47000	42
ATK0002	ATK	Kertas Kado	1500	4
ATK0003	ATK	Joyko Eraser EB 30	1800	49
ATK0004	ATK	Tip-x	25889	90
ATK0005	ATK	Buku Tulis SIDU 58 Lembar	39000	71

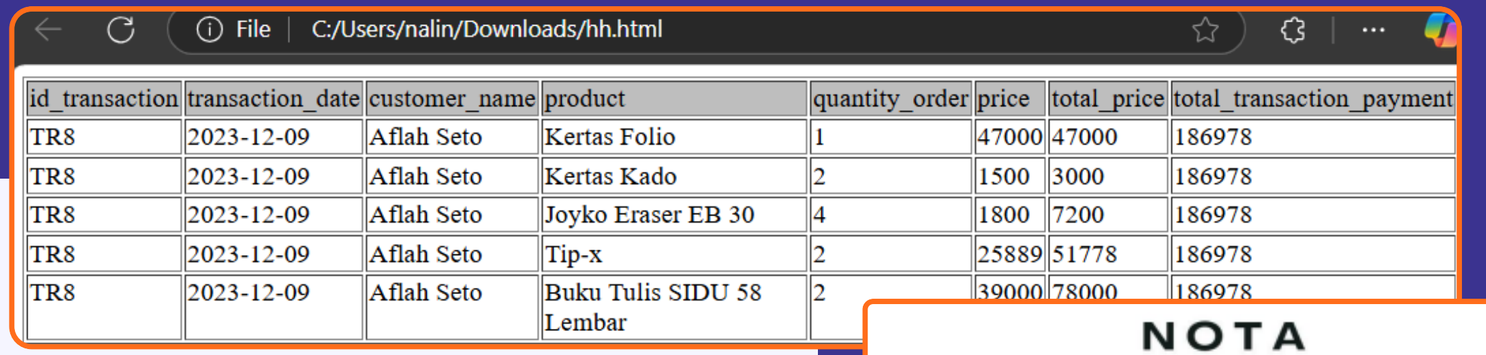
```
call nambah_barang('ATK0002', 2);
```

id_product	product	price	stock
ATK0001	Kertas Folio	47000	42
ATK0002	Kertas Kado	1500	6
ATK0003	Joyko Eraser EB 30	1800	49
ATK0004	Tip-x	25889	90
ATK0004	Tip-x	25889	90

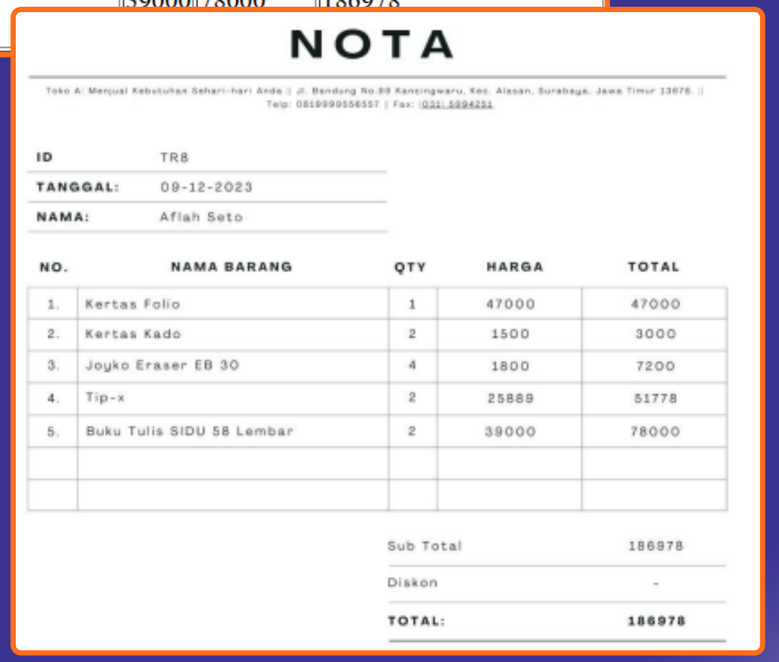
QUERY

```sql
-- Feature 4: Print Bill --
select
    transaction.id_transaction,
    transaction.transaction_date,
    customer.name as customer_name,
    product.product,
    detail_transaction.quantity_order,
    product.price,
    (detail_transaction.quantity_order * product.price) as total_price,
    sum(detail_transaction.quantity_order * product.price) over () as total_transaction_payment
from detail_transaction
join product on detail_transaction.id_product = product.id_product
join transaction on detail_transaction.id_transaction = transaction.id_transaction
join customer on transaction.id_customer = customer.id_customer
where transaction.id_transaction = 'TR2';
```

OUTPUT



id_transaction	transaction_date	customer_name	product	quantity_order	price	total_price	total_transaction_payment
TR8	2023-12-09	Aflah Seto	Kertas Folio	1	47000	47000	186978
TR8	2023-12-09	Aflah Seto	Kertas Kado	2	1500	3000	186978
TR8	2023-12-09	Aflah Seto	Joyko Eraser EB 30	4	1800	7200	186978
TR8	2023-12-09	Aflah Seto	Tip-x	2	25889	51778	186978
TR8	2023-12-09	Aflah Seto	Buku Tulis SIDU 58 Lembar	2	39000	78000	186978



NOTA

Toko A: Menjual Kebutuhan Sehari-hari Anda || Jl. Bandung No.89 Kensingwaru, Kec. Alasan, Surabaya, Jawa Timur 13678. ||
Telp: 0819999556557 | Fax: (031) 5994251

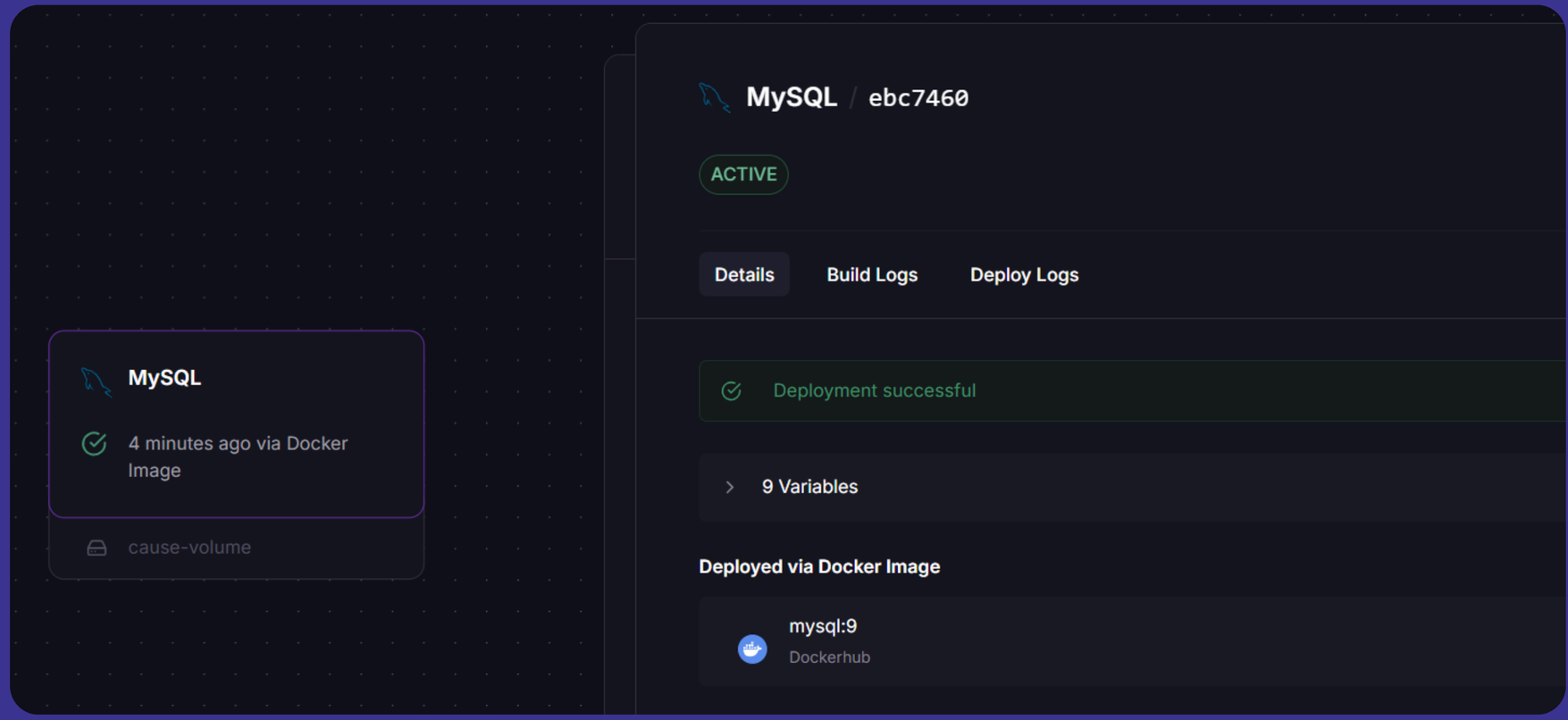
ID TR8
TANGGAL: 09-12-2023
NAMA: Aflah Seto

NO.	NAMA BARANG	QTY	HARGA	TOTAL
1.	Kertas Folio	1	47000	47000
2.	Kertas Kado	2	1500	3000
3.	Joyko Eraser EB 30	4	1800	7200
4.	Tip-x	2	25889	51778
5.	Buku Tulis SIDU 58 Lembar	2	39000	78000

Sub Total 186978
Diskon -
TOTAL: 186978

With this query, the information is retrieved to generate an invoice that can be further processed on the back-end, manually or as simple as exporting it to HTML.

don’t forget to put .gitignore file on your github :)



mysql://root:mRpU[redacted]VeJksvg@ro[redacted]roxy.rlwy.net:2[redacted]4/railway?

05. DEPLOY DATABASE

Thank You