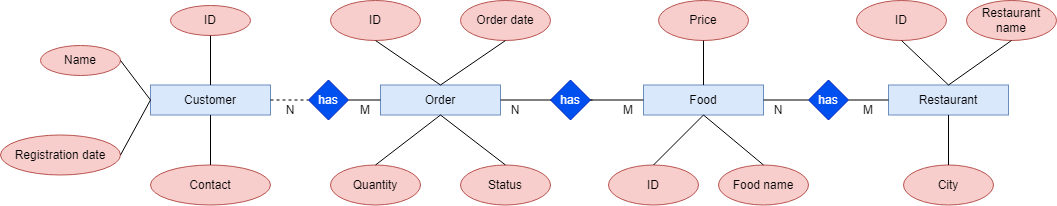
## User Manual

Dokumen ini bertujuan untuk mempermudah pembaca dalam memhami rancangan dan struktur database yang teman-teman buat.

## Entity Relationship Diagram

//Masukan desain / gambar terupdate yang dimiliki



| *List of Table* | *Description* | *Limitation* | *Notes* |
| --- | --- | --- | --- |
| Customer | Menyimpan data seluruh pengguna aplikasi OK!Food Delivery | Hanya menyimpan id, nama, tanggal registrasi, dan kontak | Tidak memiliki primary key / foreign key |
| Order | Menyimpan data transaksi customer pada aplikasi OK!Food Delivery | Hanya menyimpan id, tanggal transaksi, quantity, dan nama makanan | Memiliki foreign key ke tabel customer dan food |
| Food | Menyimpan data menu makanan pada aplikasi OK!Food Delivery | Hanya menyimpan id, nama makanan, dan harga makanan | Memiliki foreign key ke tabel restaurant |
| Restaurant | Menyimpan data restaurant pada aplikasi OK!Food Delivery | Hanya menyimpan id, nama restaurant, dan kota/lokasi restaurant | Tidak memiliki primary key / foreign key |

## 

## Data Catalog

//Berisikan semua tabel dan penjelasan masing-masing kolom yang dimiliki

### Table Customer

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| id\_customer | Integer | Id dari pengguna | Hanya terdiri dari angka saja |
| name | String | Nama dari pengguna | Hanya terdiri dari huruf saja |
| registration\_date | Date | Tanggal registrasi pengguna | Hanya terdiri dari tanggal saja |
| contact | String | Kontak pengguna yang dapat dihubungi | Hanya terdiri dari string saja (angka berformat string) |

### Table Order

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| id\_order | Integer | Id dari order/pesanan | Hanya terdiri dari angka saja |
| fk\_customer\_id | Integer | Id dari pengguna | Hanya terdiri dari angka saja |
| order\_date | Date | Tanggal transaksi pengguna | Hanya terdiri dari tanggal saja |
| fk\_food\_id | Integer | Id dari menu makanan | Hanya terdiri dari angka saja |
| quantity | Integer | Banyaknya pesanan | Hanya terdiri dari angka saja |
| status | String | Status pesanan | Hanya terdiri dari huruf saja (pending atau sukses) |

### Table Food

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| id\_food | Integer | Id dari menu makanan | Hanya terdiri dari angka saja |
| food\_name | String | Nama dari menu makanan | Hanya terdiri dari huruf saja |
| fk\_resto\_id | Integer | Id dari restaurant | Hanya terdiri dari angka saja |
| price | Integer | Harga menu makanan persatuan | Hanya terdiri dari angka saja |

### Table Restaurant

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| id\_restaurant | Integer | Id dari pengguna | Hanya terdiri dari angka saja |
| restaurant\_name | String | Nama dari restaurant | Hanya terdiri dari huruf saja |
| city | String | Kota/lokasi dari restaurant | Hanya terdiri dari huruf saja |

## Report Catalog

// berisikin semua report yang teman-teman buat / table aggregasi

### **Part 2**

### Table Daily Registration Customer

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| registration\_date | Date | Tanggal dari pengguna registrasi | Hanya terdiri dari tanggal saja |
| Jumlah\_Customer\_Registrasi | Integer | Jumlah customer | Hanya terdiri dari angka saja |

#### Query

// query untuk membentuk table tersebut

**SELECT**

registration\_date,

**COUNT**(id\_customer) **AS** Jumlah\_Customer\_Regristrasi

**FROM** `aqueous-freedom-398208.OFD\_Team\_1.Customer`

**GROUP** **BY** registration\_date

**ORDER** **BY** registration\_date **ASC**;

### Table Daily Order

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| order\_date | Date | Tanggal dari pengguna transaksi/order | Hanya terdiri dari tanggal saja |
| Jumlah\_Order | Integer | Jumlah order | Hanya terdiri dari angka saja |

#### Query

// query untuk membentuk table tersebut

**SELECT**

Order\_date,

COUNT(id\_order) **AS** Jumlah\_Order

**FROM** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Order`

**GROUP BY** order\_date

**ORDER BY** order\_date **ASC**;

### Table Daily Revenue (Keuntungan Harian)

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| Tanggal | Date | Tanggal dari keuntungan harian | Hanya terdiri dari tanggal saja |
| Keuntungan | Integer | Total keuntungan | Hanya terdiri dari angka saja |

#### Query

// query untuk membentuk table tersebut

**SELECT**

o.order\_date **AS** Tanggal,

SUM(o.quantity \* f.price) **AS** Keuntungan

**FROM** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Order` o

**INNER** **JOIN** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Food` f **ON** o.fk\_food\_id = f.id\_food

**GROUP BY** o.order\_date

**ORDER BY** o.order\_date **ASC**;

### Table Daily Average Spending Customer Per-Month

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| nama\_customer | String | Nama dari customer/pengguna | Hanya terdiri dari huruf saja |
| Bulan | Date | Bulan | Hanya terdiri dari tanggal saja |
| rata\_rata\_pengeluaran\_per\_transaksi | Integer | Rata-rata pengeluaran 1 customer dalam 1 bulan | Hanya terdiri dari angka saja |

#### Query

// query untuk membentuk table tersebut

**WITH** MonthlyTransactions **AS** (

**SELECT**

DATE\_TRUNC(o.order\_date, **MONTH**) **AS** bulan,

o.fk\_customer\_id,

COUNT(\*) **AS** jumlah\_transaksi

**FROM** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Order` o

**GROUP BY** bulan, o.fk\_customer\_id

),

MonthlyExpenditures **AS** (

**SELECT**

DATE\_TRUNC(o.order\_date, **MONTH**) **AS** bulan,

o.fk\_customer\_id,

SUM(f.price \* o.quantity) **AS** total\_pengeluaran

**FROM** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Order` o

**JOIN** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Food` f **ON** o.fk\_food\_id = f.id\_food

**GROUP BY**

bulan, o.fk\_customer\_id

),

CustomerData **AS** (

**SELECT**

c.id\_customer,

c.name **AS** nama\_customer

**FROM**

`aqueous-freedom-398208.OFD**\_**Team**\_**1.Customer` c

)

**SELECT**

cd.nama\_customer,

mt.bulan **AS** Bulan,

COALESCE(ROUND(me.total\_pengeluaran / mt.jumlah\_transaksi, 2), 0) **AS** rata\_rata\_pengeluaran\_per\_transaksi

**FROM**

MonthlyTransactions mt

LEFT **JOIN**

MonthlyExpenditures me **ON** mt.bulan = me.bulan **AND** mt.fk\_customer\_id = me.fk\_customer\_id

**JOIN**

CustomerData cd **ON** mt.fk\_customer\_id = cd.id\_customer

**ORDER BY**

Bulan **ASC**, cd.nama\_customer **ASC**;

### 

### **Part 4**

### Table Segmentasi Customer

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| name | String | Nama customer/pengguna | Hanya terdiri dari huruf saja |
| total\_order | Integer | Total transaksi/order | Hanya terdiri dari angka saja |
| total\_spend | Integer | Total uang yang dikeluarkan customer | Hanya terdiri dari angka saja |
| level\_segmentasi | String | Segmentasi/kategori dari customer | Hanya terdiri dari huruf saja |

#### Query

// query untuk membentuk table tersebut

**WITH** kategori **AS** (

**SELECT**

c.name,

COUNT(\*) **AS** total\_order,

SUM(f.price\*o.quantity) **AS** total\_spend

**FROM** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Order` o

**INNER** **JOIN** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Food` f **ON** f.id\_food = o.fk\_food\_id

**INNER** **JOIN** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Customer` c **ON** c.id\_customer = o.fk\_customer\_id

**GROUP BY** 1

)

**SELECT**

k.name,

k.total\_order,

k.total\_spend,

CASE

WHEN k.total\_order > 3 **AND** k.total\_spend >= 150000 THEN 'Big Spender'

WHEN k.total\_spend < 50000 THEN 'Regular'

WHEN k.total\_spend < 100000 THEN 'Silver'

WHEN k.total\_spend < 150000 THEN 'VIP'

ELSE 'VVIP' END level\_segmentasi

**FROM** kategori **AS** k

**ORDER BY** 2, 3, 4;

### Table Transaksi Pending

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| name | String | Nama dari customer/pengguna | Hanya terdiri dari huruf saja |
| order\_date | Date | Tanggal transaksi/order | Hanya terdiri dari tanggal saja |
| status | String | Status transaksi/order | Hanya terdiri dari huruf saja |
| contact | String | Kontak customer yang dapat dihubungi | Hanya terdiri dari angka saja tetapi berstatus string |

#### Query

// query untuk membentuk table tersebut

**SELECT**

c.name,

o.order\_date,

o.**status**,

c.contact

**FROM** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Customer` c

**INNER** **JOIN** `aqueous-freedom-398208.OFD**\_**Team**\_**1.Order` o **ON** c.id\_customer = o.fk\_customer\_id

**WHERE** o.**status** = 'pending'

**ORDER BY** o.order\_date **ASC**;

### 

### **Part 5 (Dataset Tim-1)**

### Table Revenue Harian Perkota

#### Table Structure

| Column Name | Data Type | Description | Notes |
| --- | --- | --- | --- |
| order\_date | Date | Tanggal transaksi/order | Hanya terdiri dari tanggal saja |
| city | String | Kota/lokasi dari restaurant | Hanya terdiri dari huruf saja |
| daily\_revenue | Integer | Data pendapatan harian keseluruhan restaurant di setiap kota | Hanya terdiri dari angka saja |

#### Query

// query untuk membentuk table tersebut

SELECT

o.order\_date,

r.city,

SUM(f.price \* o.quantity) AS daily\_revenue

FROM `aqueous-freedom-398208.OFD\_Team\_1.Order` o

INNER JOIN`aqueous-freedom-398208.OFD\_Team\_1.Food` f ON o.fk\_food\_id = f.id\_food

INNER JOIN `aqueous-freedom-398208.OFD\_Team\_1.Restaurant` r ON f.fk\_resto\_id = r.id\_restaurant

GROUP BY order\_date,city

ORDER BY order\_date,city;

### Table Segmentation Report Harian

#### Table Structure

| Column Name | Data Type | Description | Notes |
| --- | --- | --- | --- |
| id\_customer | Integer | Id dari pelanggan | Hanya terdiri dari angka saja |
| name | String | Nama dari pelanggan | Hanya terdiri dari huruf saja |
| today\_date | Date | Tanggal hari ini | Hanya terdiri dari tanggal saja |
| last\_order | Date | Tanggal terakhir pemesanan | Hanya terdiri dari tanggal saja |
| total\_transaction | Integer | Jumlah transaksi pada masing masing pelanggan | Hanya terdiri dari angka saja |
| total\_spending | Integer | Total harga belanja pada masing masing pelanggan | Hanya terdiri dari angka saja |
| segmentasi | String | Penggolongan pelanggan berdasarkan total transaction dan total spending | Hanya terdiri dari huruf saja |

#### Query

// query untuk membentuk table tersebut

drop table if exists`aqueous-freedom-398208.OFD\_Team\_1.costumer\_segmentation`;

create table `aqueous-freedom-398208.OFD\_Team\_1.costumer\_segmentation` AS

WITH tabel AS (

SELECT

id\_customer,

c.name,

DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY) AS last\_order\_date,

DATE\_DIFF(o.order\_date, c.registration\_date, DAY) AS days\_since\_registration,

COUNT(\*) OVER (PARTITION BY c.name) AS total\_transaction,

SUM(f.price \* o.quantity) OVER (PARTITION BY c.name) AS total\_spending,

ROW\_NUMBER() OVER (PARTITION BY c.name ORDER BY o.order\_date DESC) AS row\_num,

LAG(o.order\_date) OVER (PARTITION BY c.name ORDER BY o.order\_date) AS prev\_order\_date

FROM aqueous-freedom-398208.OFD\_Team\_1.Customer c

LEFT JOIN aqueous-freedom-398208.OFD\_Team\_1.Order o ON c.id\_customer = o.fk\_customer\_id

LEFT JOIN aqueous-freedom-398208.OFD\_Team\_1.Food f ON f.id\_food = o.fk\_food\_id AND o.order\_date >= DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY)

WHERE c.registration\_date <= DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY)

AND o.order\_date >= DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY) -- Tambahkan kondisi untuk memfilter tanggal pesanan

)

SELECT

id\_customer,

name,

DATE\_SUB(CURRENT\_DATE(), INTERVAL 0 DAY) AS today\_date,

DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY) AS last\_order,

total\_transaction,

total\_spending,

CASE

WHEN total\_spending >= 500001 THEN 'Big Spender'

WHEN total\_transaction <= 1 THEN 'New'

WHEN total\_transaction <= 5 THEN 'Good'

WHEN total\_transaction <= 10 THEN 'Potential'

ELSE 'Loyalist'

END AS segmentation

FROM tabel

ORDER BY last\_order\_date DESC;

### Table Growth harian

#### Table Structure

| Column Name | Data Type | Description | Notes |
| --- | --- | --- | --- |
| kota | String | Kota/lokasi dari restaurant | Hanya terdiri dari angka saja |
| tanggal | Date | Daftar tanggal | Hanya terdiri dari tanggal saja |
| revenue\_hari\_kemarin | Integer | Pendapatan hari sebelumnya pada setiap kota | Hanya terdiri dari angka saja |
| revenue | Integer | Pendapatan hari berikutnya pada setiap kota | Hanya terdiri dari angka saja |
| growth\_rate\_precentage | String | Pertumbuhan pendapatan dalam persen | Terdiri dari angka untuk perhitungan yang bersifat integer serta string untuk menulis kode persen |

#### Query

// query untuk membentuk table tersebut

drop table if exists `aqueous-freedom-398208.OFD\_Team\_1.coding\_id\_\_growth\_harian`;

CREATE TABLE `aqueous-freedom-398208.OFD\_Team\_1.coding\_id\_\_growth\_harian`as

WITH DailyRevenue AS (

SELECT

r.kota AS Kota,

t.transaction\_date AS Tanggal,

SUM(t.price) AS Revenue,

LAG(SUM(t.price), 1) OVER (PARTITION BY r.kota ORDER BY t.transaction\_date) AS Revenue\_Hari\_Kemarin,

LAG(SUM(t.price), 2) OVER (PARTITION BY r.kota ORDER BY t.transaction\_date) AS Revenue\_Dua\_Hari\_Sebelumnya

FROM

`aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t

INNER JOIN

`aqueous-freedom-398208.ok\_food\_dataset.resto` r ON t.resto\_id = r.id

GROUP BY

r.kota, t.transaction\_date

)

SELECT

Kota,

Tanggal,

COALESCE(Revenue\_Hari\_Kemarin, Revenue\_Dua\_Hari\_Sebelumnya, 0) AS Revenue\_Hari\_Kemarin,

Revenue,

CASE

WHEN COALESCE(Revenue\_Hari\_Kemarin, Revenue\_Dua\_Hari\_Sebelumnya) IS NULL THEN NULL

ELSE CONCAT(ROUND(((Revenue - COALESCE(Revenue\_Hari\_Kemarin, Revenue\_Dua\_Hari\_Sebelumnya)) / NULLIF(COALESCE(Revenue\_Hari\_Kemarin, Revenue\_Dua\_Hari\_Sebelumnya), 0)) \* 100, 2), '%')

END AS Growth\_Rate\_Percentage

FROM

DailyRevenue

ORDER BY

Kota, Tanggal;

### Table Kontribusi harian

#### Table Structure

| Column Name | Data Type | Description | Notes |
| --- | --- | --- | --- |
| date | Date | Daftar tanggal harian | Hanya terdiri dari tanggal saja |
| city | String | Kota/lokasi dari restaurant | Hanya terdiri dari huruf saja |
| total\_revenue | Integer | Menghitung total pendapatan | Hanya terdiri dari angka saja |
| contribution | Float | Menghitung kontribusi harian perkota | Hanya terdiri dari bilangan desimal saja |

#### Query

// query untuk membentuk table tersebut

drop table if exists `aqueous-freedom-398208.OFD\_Team\_1.kontribusi\_harian\_perkota`;

CREATE TABLE `aqueous-freedom-398208.OFD\_Team\_1.kontribusi\_harian\_perkota` AS

WITH daily\_total AS (

SELECT

order\_date,

SUM(f.price \* o.quantity) AS total\_revenue\_all\_cities

FROM `aqueous-freedom-398208.OFD\_Team\_1.Order` o

JOIN `aqueous-freedom-398208.OFD\_Team\_1.Food` f ON o.fk\_food\_id = f.id\_food

JOIN `aqueous-freedom-398208.OFD\_Team\_1.Restaurant` r ON f.fk\_resto\_id = r.id\_restaurant

GROUP BY

order\_date

)

SELECT

o.order\_date AS date,

r.city,

SUM(f.price \* o.quantity) AS total\_revenue,

ROUND((SUM(f.price \* o.quantity) / dt.total\_revenue\_all\_cities) \* 100, 2) AS contribution

FROM `aqueous-freedom-398208.OFD\_Team\_1.Order` o

JOIN `aqueous-freedom-398208.OFD\_Team\_1.Food` f ON o.fk\_food\_id = f.id\_food

JOIN `aqueous-freedom-398208.OFD\_Team\_1.Restaurant` r ON f.fk\_resto\_id = r.id\_restaurant

JOIN daily\_total dt ON o.order\_date = dt.order\_date

GROUP BY

r.city, o.order\_date, dt.total\_revenue\_all\_cities

ORDER BY

o.order\_date, r.city;

### **Part 5 (Dataset Coding.ID)**

### Table Revenue Harian Perkota

#### Table Structure

| Column Name | Data Type | Description | Notes |
| --- | --- | --- | --- |
| order\_date | Date | Tanggal transaksi/order | Hanya terdiri dari tanggal saja |
| city | String | Kota/lokasi dari restaurant | Hanya terdiri dari huruf saja |
| daily\_revenue | Integer | Data pendapatan harian keseluruhan restaurant di setiap kota | Hanya terdiri dari angka saja |

#### Query

// query untuk membentuk table tersebut

SELECT

t.transaction\_date,

r.kota AS city,

SUM(t.price) AS daily\_revenue

FROM`aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t

INNER JOIN`aqueous-freedom-398208.ok\_food\_dataset.resto` r ON t.resto\_id = r.id

GROUP BY t.transaction\_date, r.kota

ORDER BY t.transaction\_date, r.kota;

### 

### **Table Segmentation Report Harian**

#### Table Structure

| Column Name | Data Type | Description | Notes |
| --- | --- | --- | --- |
| id\_customer | Integer | Id dari pelanggan | Hanya terdiri dari angka saja |
| name | String | Nama dari pelanggan | Hanya terdiri dari huruf saja |
| today\_date | Date | Tanggal hari ini | Hanya terdiri dari tanggal saja |
| last\_order | Date | Tanggal terakhir pemesanan | Hanya terdiri dari tanggal saja |
| total\_transaction | Integer | Jumlah transaksi pada masing masing pelanggan | Hanya terdiri dari angka saja |
| total\_spending | Integer | Total harga belanja pada masing masing pelanggan | Hanya terdiri dari angka saja |
| segmentasi | String | Penggolongan pelanggan berdasarkan total transaction dan total spending | Hanya terdiri dari huruf saja |

#### Query

// query untuk membentuk table tersebut

drop table if exists`aqueous-freedom-398208.OFD\_Team\_1.coding\_id\_customer\_segmentasi`;

CREATE TABLE `aqueous-freedom-398208.OFD\_Team\_1.coding\_id\_customer\_segmentasi`as

WITH tabel AS (

SELECT

c.id AS customer\_id,

c.name,

DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY) AS last\_order\_date,

COUNT(t.id) AS total\_transaction,

COALESCE(SUM(t.price), 0) AS total\_spending

FROM aqueous-freedom-398208.ok\_food\_dataset.customer c

LEFT JOIN aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024 t ON c.id = t.customer\_id

WHERE t.transaction\_date >= DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY)

GROUP BY c.id, c.name

)

SELECT

customer\_id,

name,

DATE\_SUB(CURRENT\_DATE(), INTERVAL 0 DAY) AS today\_date,

DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY) AS last\_order,

total\_transaction,

total\_spending,

CASE

WHEN total\_spending >= 500001 THEN 'Big Spender'

WHEN total\_transaction <= 1 THEN 'New'

WHEN total\_transaction <= 5 THEN 'Good'

WHEN total\_transaction <= 10 THEN 'Potential'

ELSE 'Loyalist'

END AS segmentation

FROM tabel

ORDER BY last\_order\_date DESC;

### 

### Table Growth harian

#### Table Structure

| Column Name | Data Type | Description | Notes |
| --- | --- | --- | --- |
| kota | String | Kota/lokasi dari restaurant | Hanya terdiri dari angka saja |
| tanggal | Date | Daftar tanggal | Hanya terdiri dari tanggal saja |
| revenue\_hari\_kemarin | Integer | Pendapatan hari sebelumnya pada setiap kota | Hanya terdiri dari angka saja |
| revenue | Integer | Pendapatan hari berikutnya pada setiap kota | Hanya terdiri dari angka saja |
| growth\_rate\_precentage | String | Pertumbuhan pendapatan dalam persen | Terdiri dari angka untuk perhitungan yang bersifat integer serta string untuk menulis kode persen |

#### Query

// query untuk membentuk table tersebut

drop table if exists `aqueous-freedom-398208.OFD\_Team\_1.coding\_id\_\_growth\_harian`;

CREATE TABLE `aqueous-freedom-398208.OFD\_Team\_1.coding\_id\_\_growth\_harian`as

WITH DailyRevenue AS (

SELECT

r.kota AS Kota,

t.transaction\_date AS Tanggal,

SUM(t.price) AS Revenue,

LAG(SUM(t.price), 1) OVER (PARTITION BY r.kota ORDER BY t.transaction\_date) AS Revenue\_Hari\_Kemarin,

LAG(SUM(t.price), 2) OVER (PARTITION BY r.kota ORDER BY t.transaction\_date) AS Revenue\_Dua\_Hari\_Sebelumnya

FROM

`aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t

INNER JOIN

`aqueous-freedom-398208.ok\_food\_dataset.resto` r ON t.resto\_id = r.id

GROUP BY

r.kota, t.transaction\_date

)

SELECT

Kota,

Tanggal,

COALESCE(Revenue\_Hari\_Kemarin, Revenue\_Dua\_Hari\_Sebelumnya, 0) AS Revenue\_Hari\_Kemarin,

Revenue,

CASE

WHEN COALESCE(Revenue\_Hari\_Kemarin, Revenue\_Dua\_Hari\_Sebelumnya) IS NULL THEN NULL

ELSE CONCAT(ROUND(((Revenue - COALESCE(Revenue\_Hari\_Kemarin, Revenue\_Dua\_Hari\_Sebelumnya)) / NULLIF(COALESCE(Revenue\_Hari\_Kemarin, Revenue\_Dua\_Hari\_Sebelumnya), 0)) \* 100, 2), '%')

END AS Growth\_Rate\_Percentage

FROM

DailyRevenue

ORDER BY

Kota, Tanggal;

### 

### **Table Kontribusi harian**

#### Table Structure

| Column Name | Data Type | Description | Notes |
| --- | --- | --- | --- |
| date | Date | Daftar tanggal harian | Hanya terdiri dari tanggal saja |
| city | String | Kota/lokasi dari restaurant | Hanya terdiri dari huruf saja |
| total\_revenue | Integer | Menghitung total pendapatan | Hanya terdiri dari angka saja |
| contribution | Float | Menghitung kontribusi harian perkota | Hanya terdiri dari bilangan desimal saja |

#### Query

// query untuk membentuk table tersebut

drop table if exists

`aqueous-freedom-398208.OFD\_Team\_1.coding\_id\_kontribusi\_harian\_perkota`;

CREATE TABLE `aqueous-freedom-398208.OFD\_Team\_1.coding\_id\_kontribusi\_harian\_perkota`as

WITH city\_daily\_revenue AS (

SELECT

t.transaction\_date AS date,

r.kota AS city,

SUM(t.price) AS total\_revenue

FROM

`aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t

INNER JOIN

`aqueous-freedom-398208.ok\_food\_dataset.resto` r ON t.resto\_id = r.id

GROUP BY

t.transaction\_date, r.kota

),

daily\_total AS (

SELECT

transaction\_date AS order\_date,

SUM(price) AS total\_revenue\_all\_cities

FROM

`aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024`

GROUP BY

transaction\_date

)

SELECT

cdr.date,

cdr.city,

cdr.total\_revenue,

ROUND((cdr.total\_revenue / dt.total\_revenue\_all\_cities) \* 100, 0) AS contribution

FROM

city\_daily\_revenue AS cdr

JOIN

daily\_total AS dt ON cdr.date = dt.order\_date

ORDER BY

cdr.date, cdr.city;