DigitalSkola

DATABASE AND SQL

DATA ENGINEER BATCH 9 - WEEK #3

DWI HANDOYO

DATABASE

A database is an organized collection of structured information, or data, typically stored electronically in a computer system.

Data is crucial for responding Business Questions.

However spreadsheet, eg. MS Excel csv, got drawbacks in data storing :

1. Data storage limitation. The data are stored in many excel files due to max file size is 10 GB.
2. Difficult in getting insight.

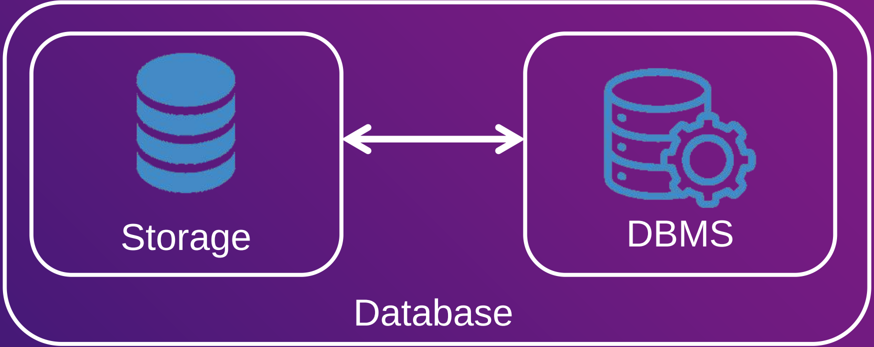
Database then replaced the spreadsheet, to cater those issues. Benefits of using database :

1. Optimize data storage.
2. Easy for data retrieval and processing.

Businesses use data stored in databases to make informed business decisions.

Basic Database Components

1. **Storage.** The memory, where the data stored
2. **Database Management System (DBMS).**



Types of Database

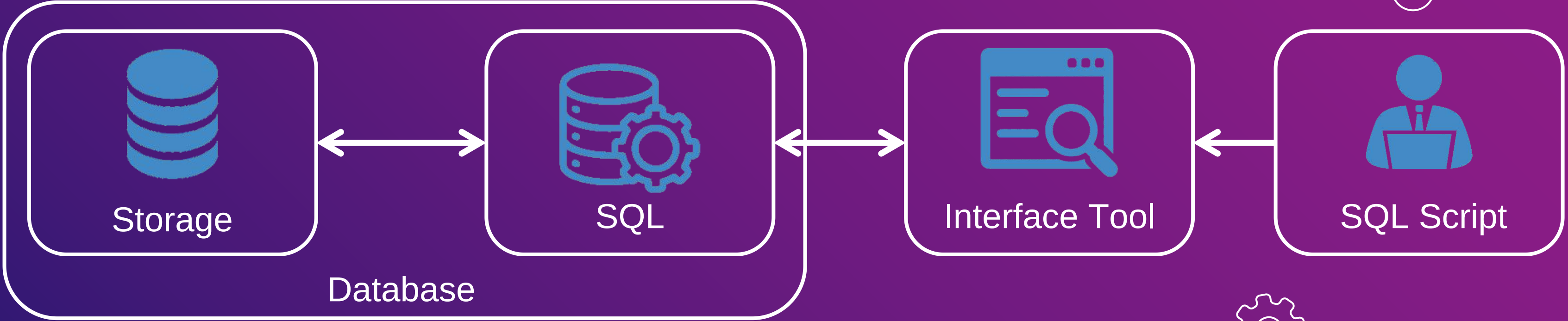
1. **SQL Database.** Relational Database Management System (RDBMS) using Structured Query Language.
2. **NoSQL Database.** Non relational database or Not Only SQL.

SQL and NoSQL Comparison

SQL	NoSQL
Relational	Non Relational
Structured	Unstructured or semistructured
Vertically scalable	Horizontally scalable
Tabular format with columns (fields) and rows (records)	Document format (JSON)
Table	Collection
Example: MySQL, PostgreSQL, MS SQL	Example: MongoDB

SQL BASICS

Structured Query Language or SQL is a standard Database language which is used to create, manipulating, and retrieve data in Relational Database Systems (RDBMS).



- **PostgresSQL** is RDBMS used in this course.
- **DBeaver Community Edition** software is used as database Interface Tool.

PostgreSQL

1. Object-oriented database management system, which means it’s a hybrid SQL/NoSQL database solution.
2. Free and open-source.
3. Compatibility with a wide range of operating systems.
4. Active community and many third-party service providers.
5. High ACID compliance. ACID is the database storage engine.
6. Uses pure SQL.
7. It also works well for extra-large databases and running complicated queries.

DBeaver

1. **Dbeaver** is a database Graphic User Interface tool for developers and database administrators.
2. Free and open-source.
3. Universal and multiplatform.

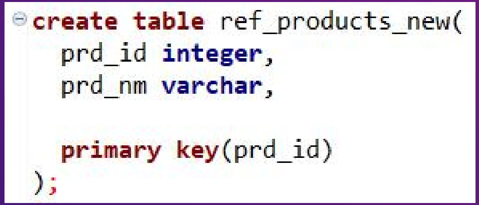
SQL Syntax Types

1. Data Definition Language (DDL)

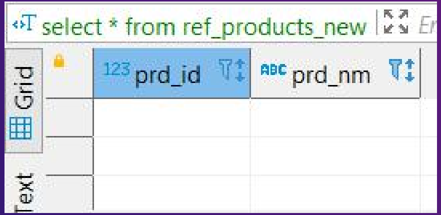
DDL is a set of SQL commands used to create, modify, and delete database structures but not for the data itself. List of DDL commands:

DDL Command	Description
CREATE	This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
DROP	This command is used to delete objects from the database.
ALTER	This is used to alter the structure of the database.
TRUNCATE	This is used to remove all records from a table, including all spaces allocated for the records are removed.

CREATE



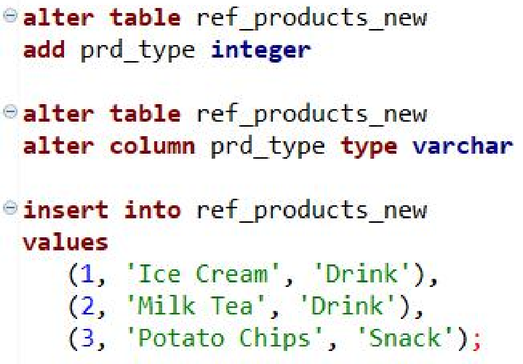
```
create table ref_products_new(
   prd_id integer,
   prd_nm varchar,

   primary key(prd_id)
);
```



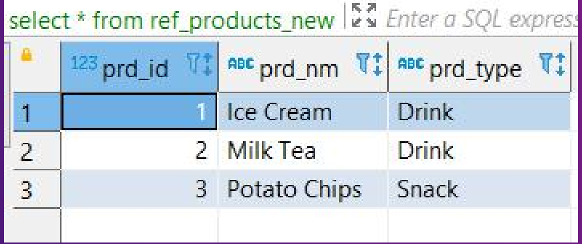
select * from ref_products_new

	prd_id	prd_nm

ALTER



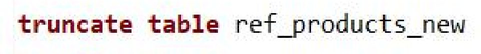
```
alter table ref_products_new
add prd_type integer

alter table ref_products_new
alter column prd_type type varchar

insert into ref_products_new
values
    (1, 'Ice Cream', 'Drink'),
    (2, 'Milk Tea', 'Drink'),
    (3, 'Potato Chips', 'Snack');
```



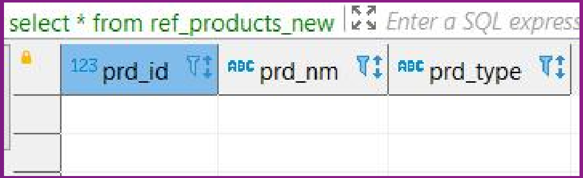
select * from ref_products_new

	prd_id	prd_nm	prd_type
1	1	Ice Cream	Drink
2	2	Milk Tea	Drink
3	3	Potato Chips	Snack

TRUNCATE



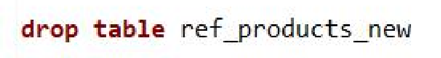
```
truncate table ref_products_new
```



select * from ref_products_new

	prd_id	prd_nm	prd_type

DROP



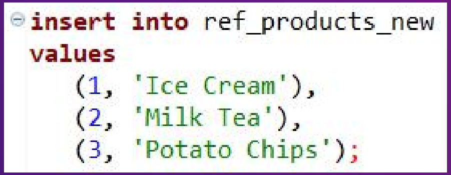
```
drop table ref_products_new
```

2. Data Manipulation Language (DML)

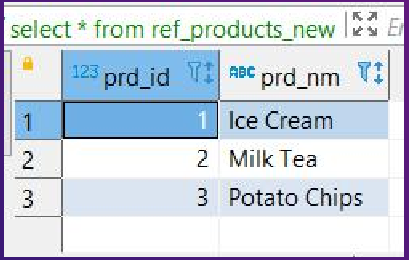
The SQL commands that deals with the manipulation/change of data present in the database.
List of DML commands:

DML Command	Description
INSERT	It is used to insert data into a table.
UPDATE	It is used to update existing data within a table.
DELETE	It is used to delete records from a database table.

INSERT



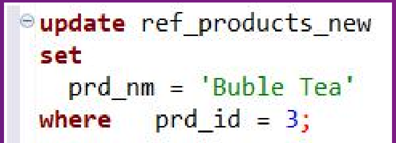
```
insert into ref_products_new
values
    (1, 'Ice Cream'),
    (2, 'Milk Tea'),
    (3, 'Potato Chips');
```



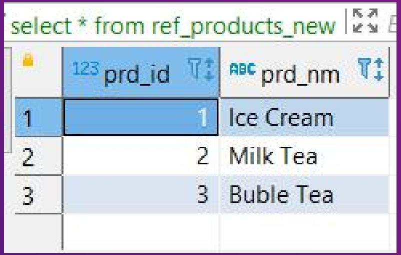
select * from ref_products_new

	123 prd_id	ABC prd_nm
1	1	Ice Cream
2	2	Milk Tea
3	3	Potato Chips

UPDATE



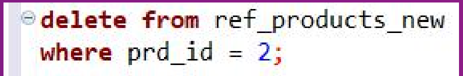
```
update ref_products_new
set
   prd_nm = 'Buble Tea'
where    prd_id = 3;
```



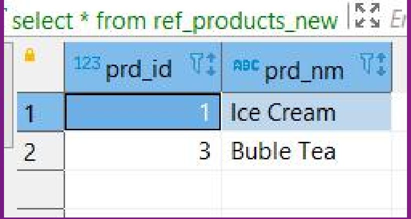
select * from ref_products_new

	123 prd_id	ABC prd_nm
1	1	Ice Cream
2	2	Milk Tea
3	3	Buble Tea

DELETE



```
delete from ref_products_new
where prd_id = 2;
```



select * from ref_products_new

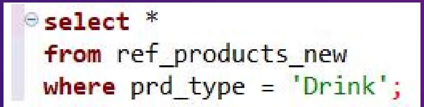
	123 prd_id	ABC prd_nm
1	1	Ice Cream
2	3	Buble Tea

3. Data Query Language (DQL)

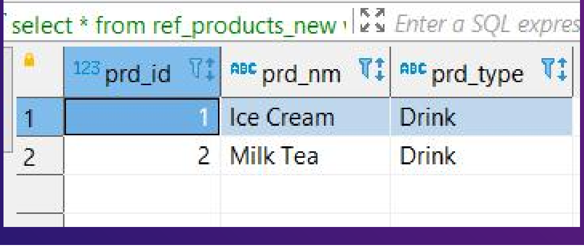
DQL is a set of SQL commands that allows in getting the data out of the database to perform operations with it. List of DQL commands:

DQL Command	Description
SELECT:	It is used to retrieve data from the database.

SELECT



```sql
select *
from ref_products_new
where prd_type = 'Drink';
```

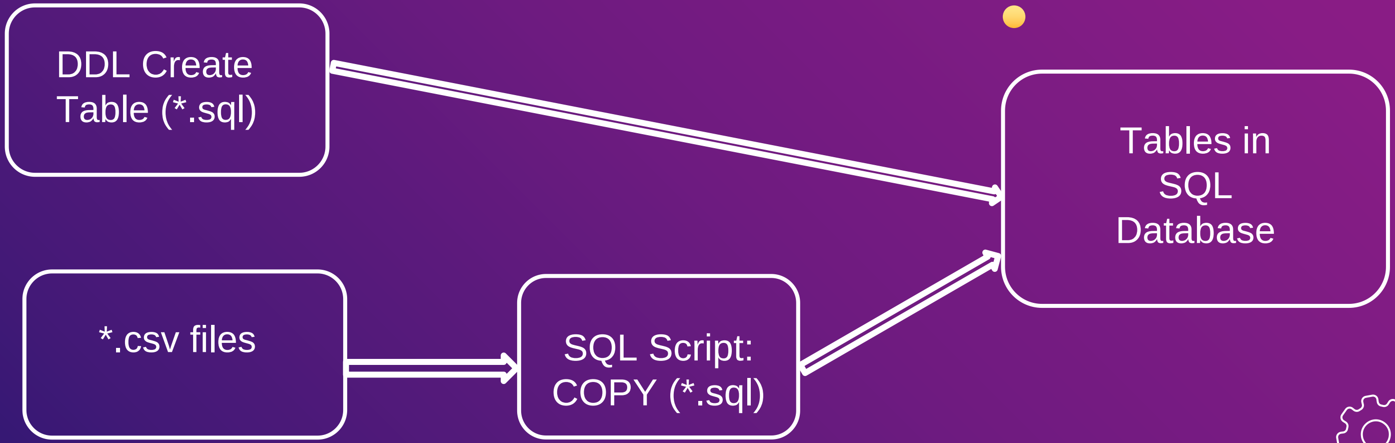


```
select * from ref_products_new
```

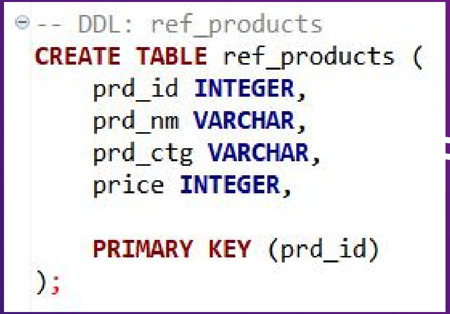
	prd_id	prd_nm	prd_type
1	1	Ice Cream	Drink
2	2	Milk Tea	Drink

Data Loading

If available data is in other format, for example in csv format, the csv format files can be loaded into SQL database.

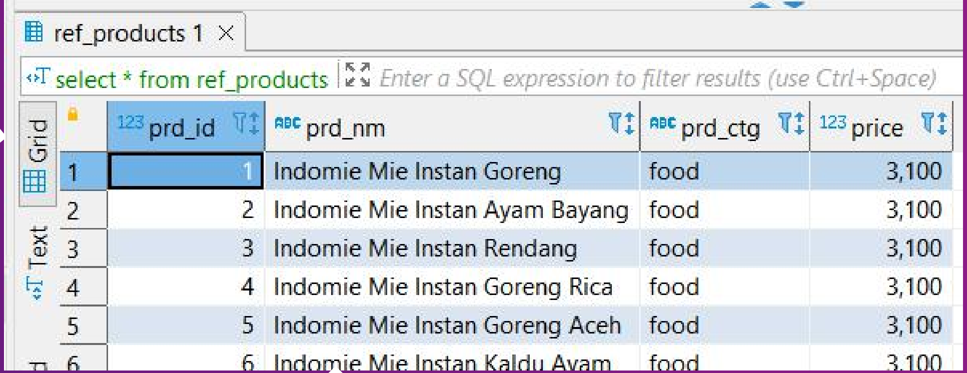


Data Loading of ref_products.csv



```
-- DDL: ref_products
CREATE TABLE ref_products (
    prd_id INTEGER,
    prd_nm VARCHAR,
    prd_ctg VARCHAR,
    price INTEGER,

    PRIMARY KEY (prd_id)
);
```

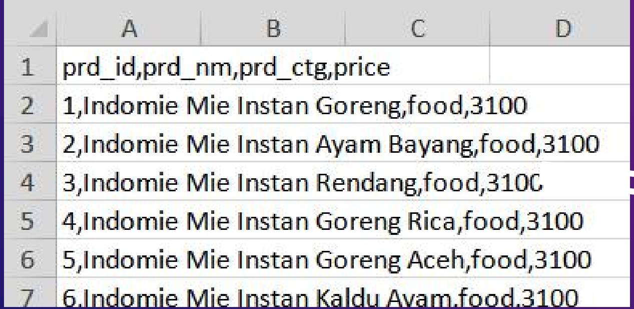
Create (DDL) *.sql file



```
select * from ref_products
```

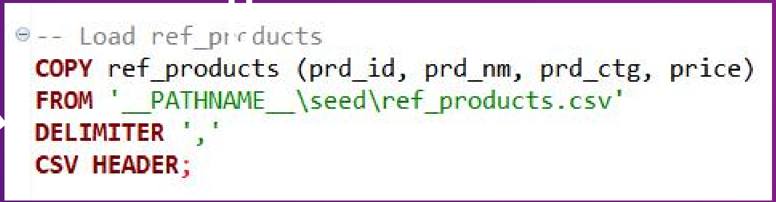
	prd_id	prd_nm	prd_ctg	price
1	1	Indomie Mie Instan Goreng	food	3,100
2	2	Indomie Mie Instan Ayam Bayang	food	3,100
3	3	Indomie Mie Instan Rendang	food	3,100
4	4	Indomie Mie Instan Goreng Rica	food	3,100
5	5	Indomie Mie Instan Goreng Aceh	food	3,100
6	6	Indomie Mie Instan Kaldu Ayam	food	3,100

Data copying is completed



	A	B	C	D
1	prd_id,prd_nm,prd_ctg,price			
2	1,Indomie Mie Instan Goreng,food,3100			
3	2,Indomie Mie Instan Ayam Bayang,food,3100			
4	3,Indomie Mie Instan Rendang,food,3100			
5	4,Indomie Mie Instan Goreng Rica,food,3100			
6	5,Indomie Mie Instan Goreng Aceh,food,3100			
7	6,Indomie Mie Instan Kaldu Ayam,food,3100			

input data *.csv file



```
-- Load ref_products
COPY ref_products (prd_id, prd_nm, prd_ctg, price)
FROM '__PATHNAME__\seed\ref_products.csv'
DELIMITER ','
CSV HEADER;
```

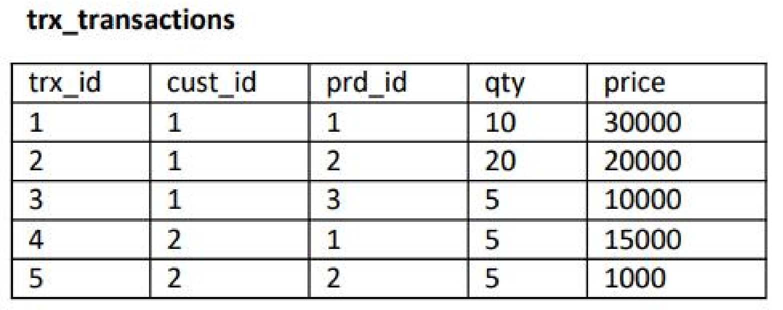
COPY csv into database *.sql file

SQL SCRIPTS

Relational Tables

In relational database, tables can be combined (join). As an example, there are three tables in a database

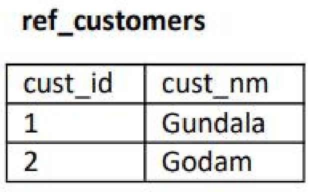
1. Transactions record table



trx_transactions

trx_id	cust_id	prd_id	qty	price
1	1	1	10	30000
2	1	2	20	20000
3	1	3	5	10000
4	2	1	5	15000
5	2	2	5	1000

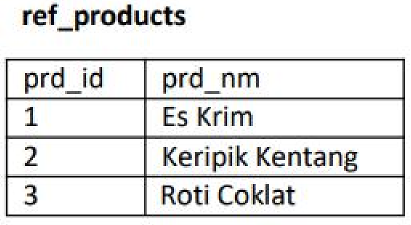
2. Customers reference table



ref_customers

cust_id	cust_nm
1	Gundala
2	Godam

3. Products reference table



ref_products

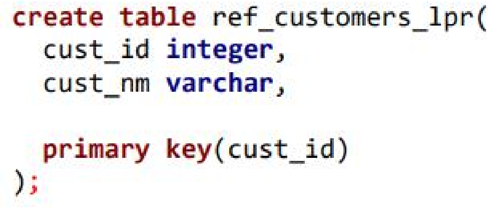
prd_id	prd_nm
1	Es Krim
2	Keripik Kentang
3	Roti Coklat

Business Question:

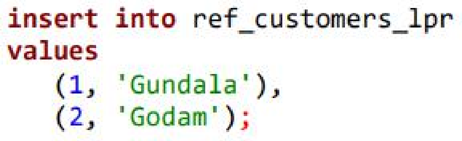
How many products did each person buy in total?

Create tables (DDL) and insert data (DML) in the PostgreSQL database by running SQL script via DBeaver interface.

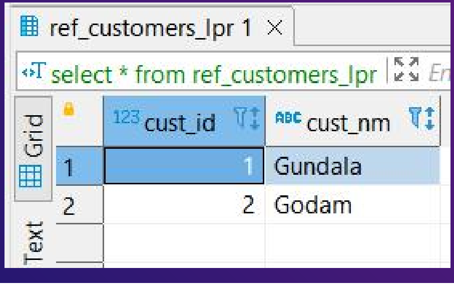
Customers reference table



```
create table ref_customers_lpr(
  cust_id integer,
  cust_nm varchar,

  primary key(cust_id)
);
```



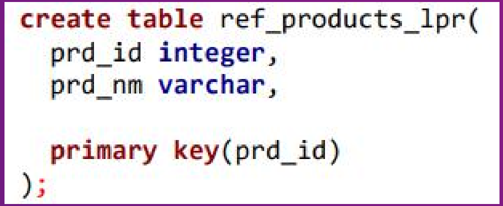
```
insert into ref_customers_lpr
values
   (1, 'Gundala'),
   (2, 'Godam');
```



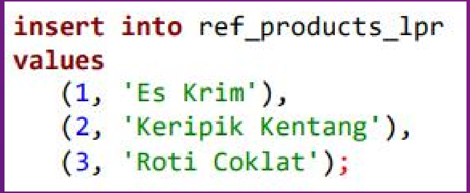
select * from ref_customers_lpr

	cust_id	cust_nm
1	1	Gundala
2	2	Godam

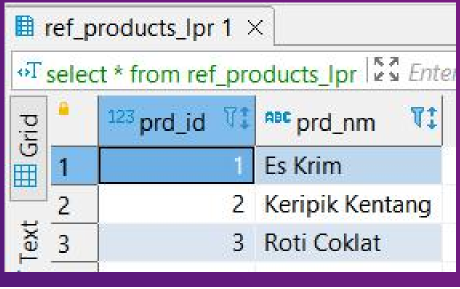
Products reference table



```
create table ref_products_lpr(
  prd_id integer,
  prd_nm varchar,

  primary key(prd_id)
);
```



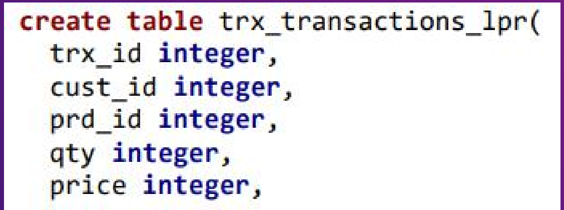
```
insert into ref_products_lpr
values
   (1, 'Es Krim'),
   (2, 'Keripik Kentang'),
   (3, 'Roti Coklat');
```



select * from ref_products_lpr

	prd_id	prd_nm
1	1	Es Krim
2	2	Keripik Kentang
3	3	Roti Coklat

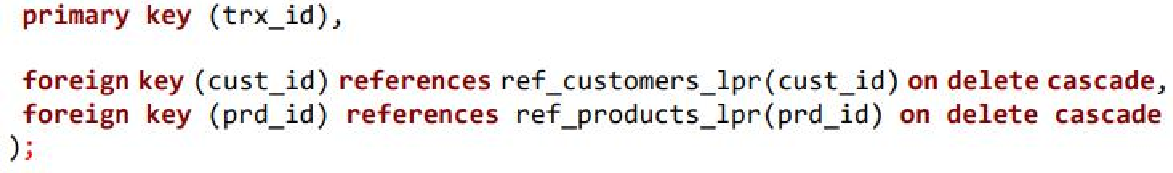
Create transactions record table complete with Primary Key and Foreign Key.



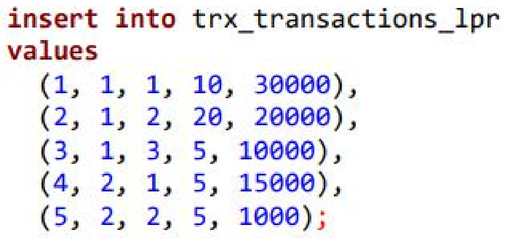
```
create table trx_transactions_lpr(
  trx_id integer,
  cust_id integer,
  prd_id integer,
  qty integer,
  price integer,
```

Primary Key is a unique key that distinguishes each row of data.

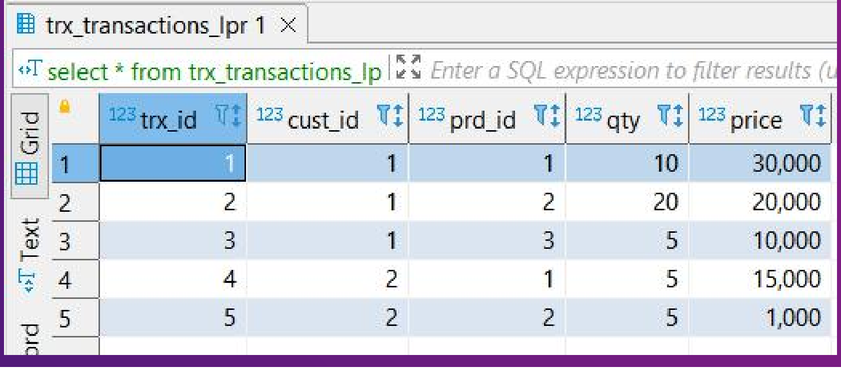
Foreign Key is the primary key of another table that also describes the data row and can be used to combine data (join tables).



```
 primary key (trx_id),

 foreign key (cust_id) references ref_customers_lpr(cust_id) on delete cascade,
 foreign key (prd_id) references ref_products_lpr(prd_id) on delete cascade
);
```



```
insert into trx_transactions_lpr
values
   (1, 1, 1, 10, 30000),
   (2, 1, 2, 20, 20000),
   (3, 1, 3, 5, 10000),
   (4, 2, 1, 5, 15000),
   (5, 2, 2, 5, 1000);
```

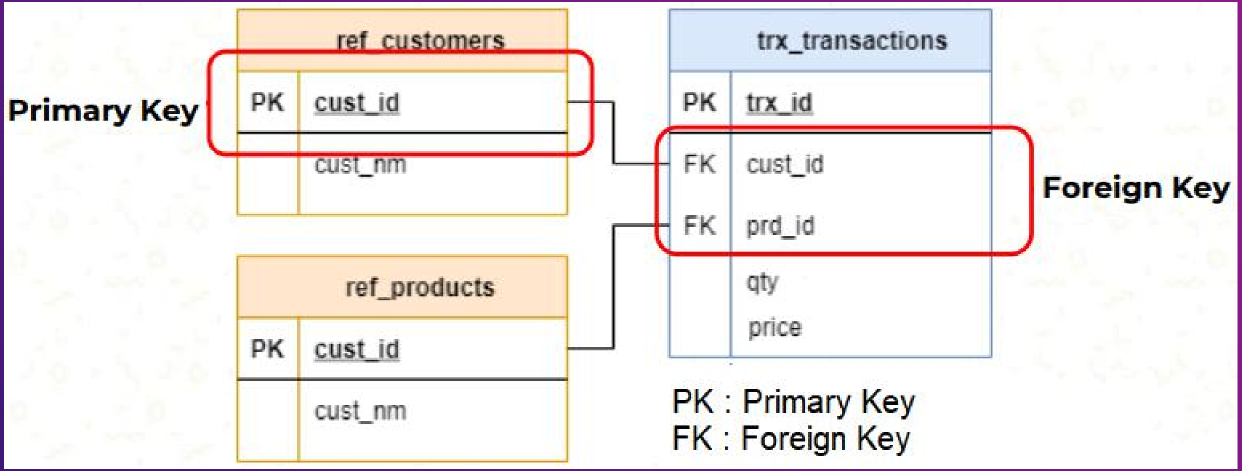


```
select * from trx_transactions_lp
```

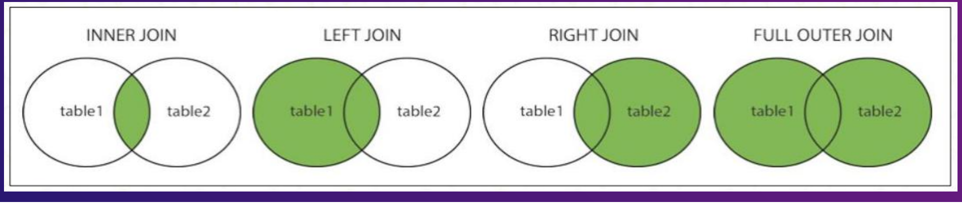
Enter a SQL expression to filter results (u

	trx_id	cust_id	prd_id	qty	price
1	1	1	1	10	30,000
2	2	1	2	20	20,000
3	3	1	3	5	10,000
4	4	2	1	5	15,000
5	5	2	2	5	1,000

Join the three tables by using Primary Key and Foreign Key

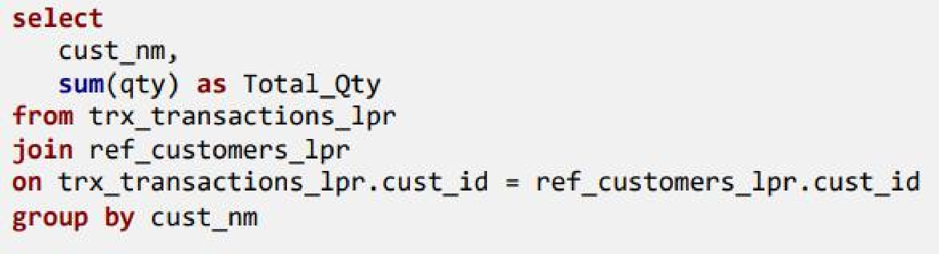


Types of Join



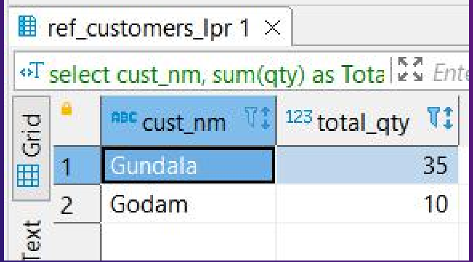
SQL SCRIPTS

To answer the Business Question, select and join the tables by using below script.



```
select
   cust_nm,
   sum(qty) as Total_Qty
from trx_transactions_lpr
join ref_customers_lpr
on trx_transactions_lpr.cust_id = ref_customers_lpr.cust_id
group by cust_nm
```

Here is the result in DBeaver interface.



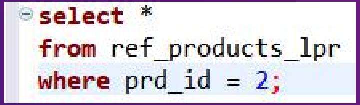
```
select cust_nm, sum(qty) as Tota
```

	cust_nm	total_qty
1	Gundala	35
2	Godam	10

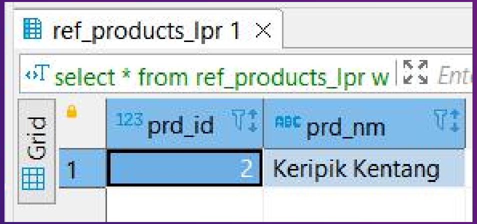
All scripts can be saved by DBeaver in *.sql format.

Filter Query Scripts

Simple filter using where

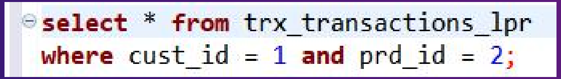


```
select *
from ref_products_lpr
where prd_id = 2;
```

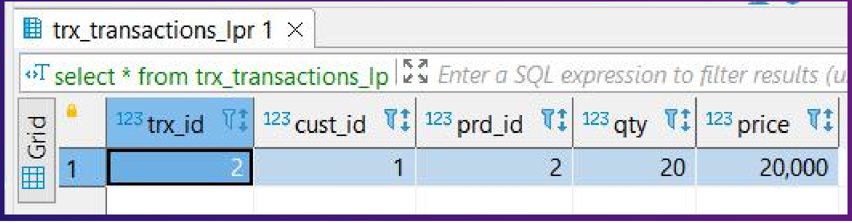


	prd_id	prd_nm
1	2	Keripik Kentang

Query filter using 'and'



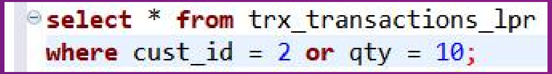
```
select * from trx_transactions_lpr
where cust_id = 1 and prd_id = 2;
```



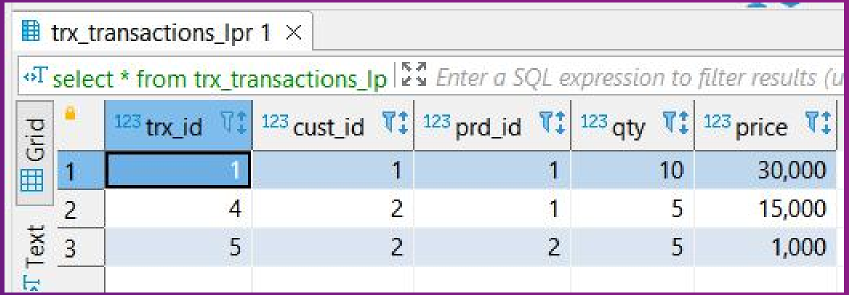
trx_transactions_lpr 1
select * from trx_transactions_lp
Enter a SQL expression to filter results (u
Grid

	trx_id	cust_id	prd_id	qty	price
1	2	1	2	20	20,000

Query filter using 'or'



```
select * from trx_transactions_lpr
where cust_id = 2 or qty = 10;
```



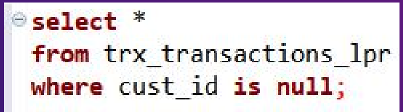
trx_transactions_lpr 1
select * from trx_transactions_lp
Enter a SQL expression to filter results (u
Grid
Text

	trx_id	cust_id	prd_id	qty	price
1	1	1	1	10	30,000
2	4	2	1	5	15,000
3	5	2	2	5	1,000

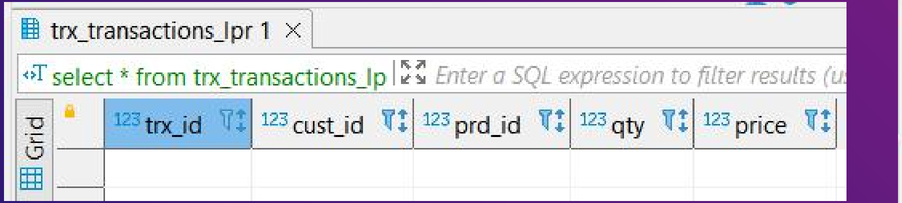
Conditions Scripts

Null Checking

NULL is the condition when the column has no values. NULL value can be checked only with IS or IS NOT.

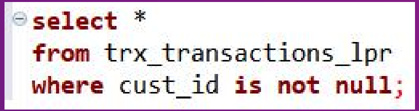


```
select *
from trx_transactions_lpr
where cust_id is null;
```

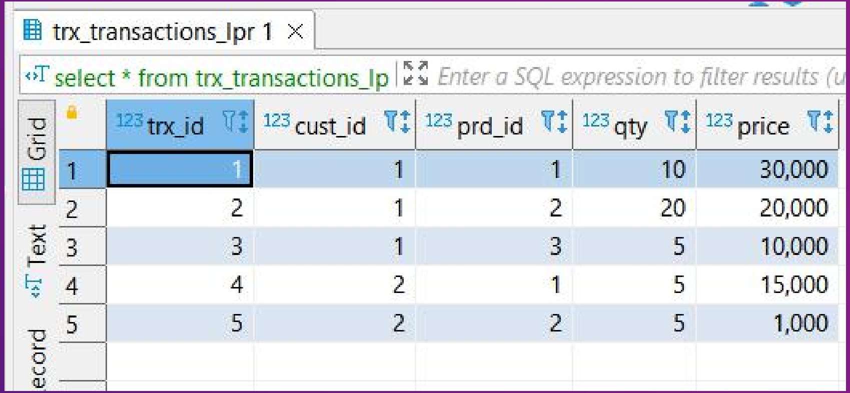


trx_transactions_lpr 1
select * from trx_transactions_lp
Enter a SQL expression to filter results (u
Grid

trx_id	cust_id	prd_id	qty	price



```
select *
from trx_transactions_lpr
where cust_id is not null;
```

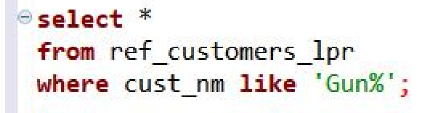


trx_transactions_lpr 1
select * from trx_transactions_lp
Enter a SQL expression to filter results (u
Grid
Text
ecord

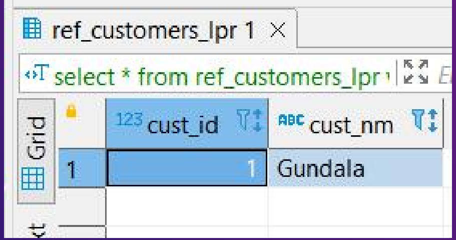
	trx_id	cust_id	prd_id	qty	price
1	1	1	1	10	30,000
2	2	1	2	20	20,000
3	3	1	3	5	10,000
4	4	2	1	5	15,000
5	5	2	2	5	1,000

Like

LIKE is used when you want to find substring of a data. % in LIKE syntax interpreted as a wildcard (any character).

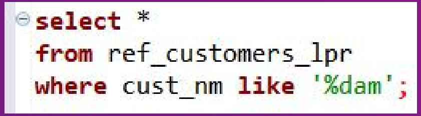


```
select *
from ref_customers_lpr
where cust_nm like 'Gun%';
```

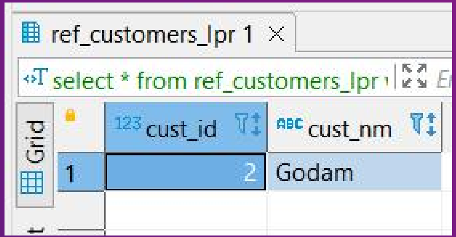


```
select * from ref_customers_lpr
```

	cust_id	cust_nm
1	1	Gundala



```
select *
from ref_customers_lpr
where cust_nm like '%dam';
```

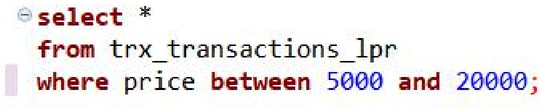


```
select * from ref_customers_lpr
```

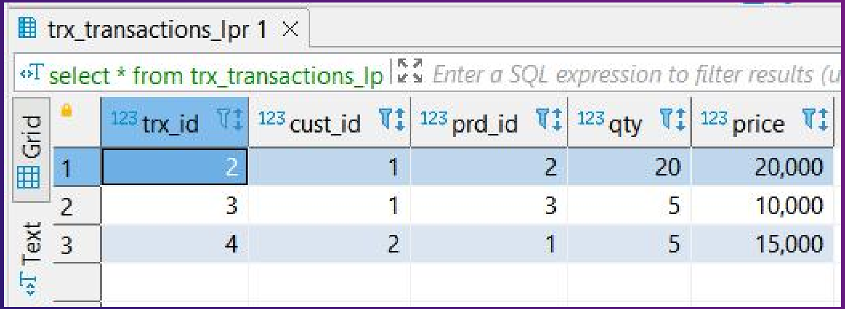
	cust_id	cust_nm
1	2	Godam

Between

BETWEEN is used if you want to find data with the value of a column is located on a range.



```sql
select *
from trx_transactions_lpr
where price between 5000 and 20000;
```

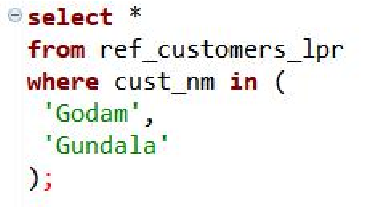


select * from trx_transactions_lp | Enter a SQL expression to filter results (u

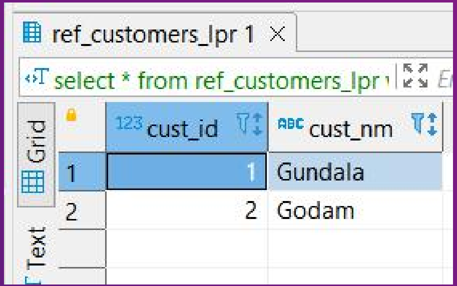
	trx_id	cust_id	prd_id	qty	price
1	2	1	2	20	20,000
2	3	1	3	5	10,000
3	4	2	1	5	15,000

In

IN is used if you want to find data with a certain value based on a list values.



```sql
select *
from ref_customers_lpr
where cust_nm in (
  'Godam',
  'Gundala'
);
```



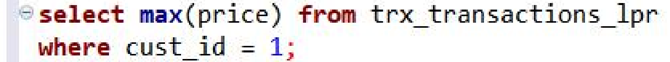
select * from ref_customers_lpr | E

	cust_id	cust_nm
1	1	Gundala
2	2	Godam

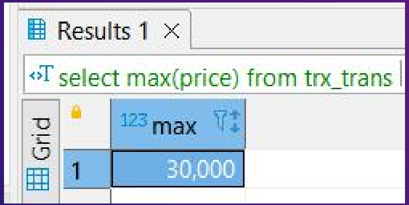
Aggregate Functions

Converts a group of values to a single value. Combined with GROUP BY to get the aggregation result for each group. If there is no GROUP BY, then the aggregation function will aggregate all the data.

Max



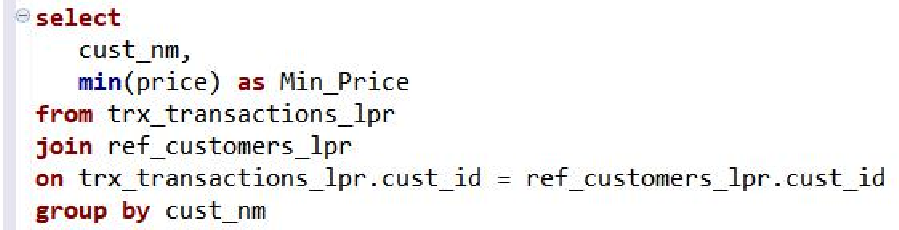
```sql
select max(price) from trx_transactions_lpr
where cust_id = 1;
```



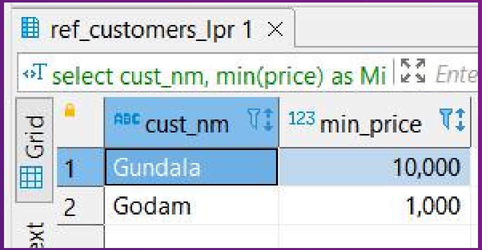
select max(price) from trx_trans

	max
1	30,000

Min



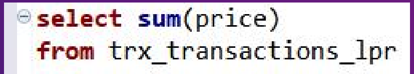
```sql
select
    cust_nm,
    min(price) as Min_Price
from trx_transactions_lpr
join ref_customers_lpr
on trx_transactions_lpr.cust_id = ref_customers_lpr.cust_id
group by cust_nm
```



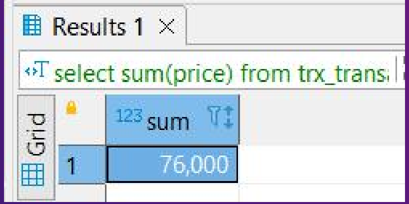
select cust_nm, min(price) as Mi

	cust_nm	min_price
1	Gundala	10,000
2	Godam	1,000

Sum

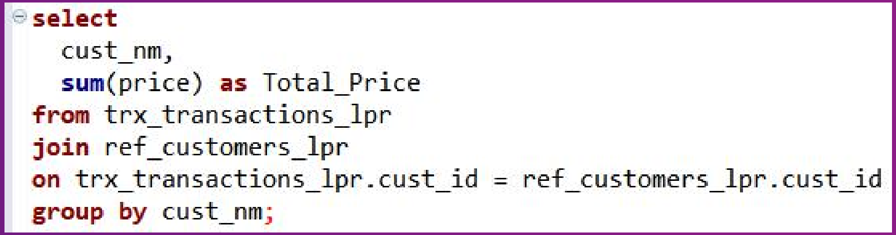


```
select sum(price)
from trx_transactions_lpr
```

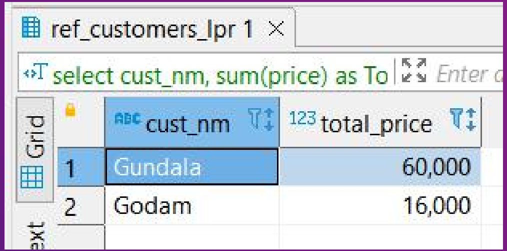


select sum(price) from trx_trans

	sum
1	76,000



```
select
   cust_nm,
   sum(price) as Total_Price
from trx_transactions_lpr
join ref_customers_lpr
on trx_transactions_lpr.cust_id = ref_customers_lpr.cust_id
group by cust_nm;
```

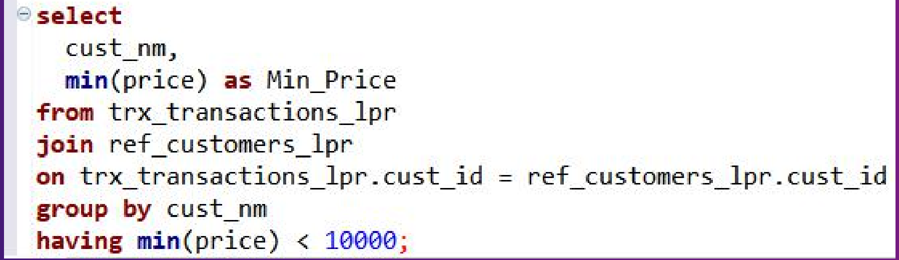


select cust_nm, sum(price) as To

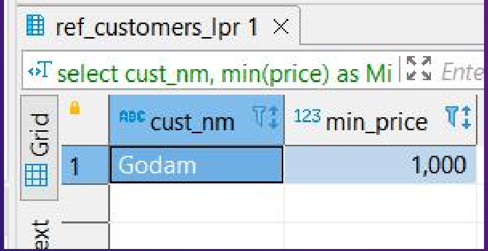
	cust_nm	total_price
1	Gundala	60,000
2	Godam	16,000

Aggregate Filter

Min/Max



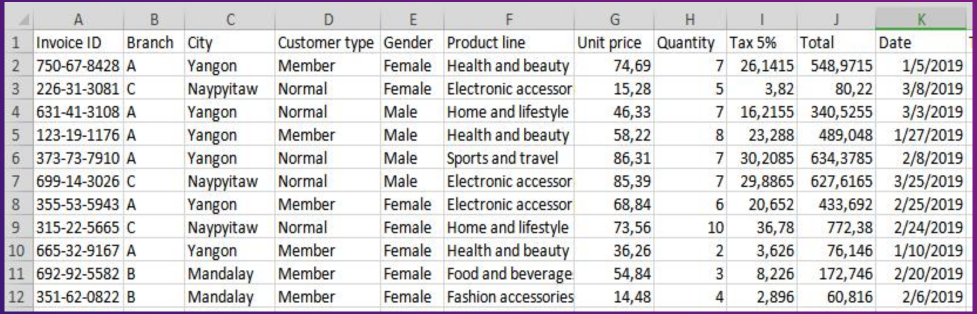
```
select
   cust_nm,
   min(price) as Min_Price
from trx_transactions_lpr
join ref_customers_lpr
on trx_transactions_lpr.cust_id = ref_customers_lpr.cust_id
group by cust_nm
having min(price) < 10000;
```



	cust_nm	min_price
1	Godam	1,000

- WHERE clause is only used to filter row data
- HAVING clause is only used to filter the aggregation results
- WHERE and HAVING clases can be used together

PROJECT 1 : PYTHON

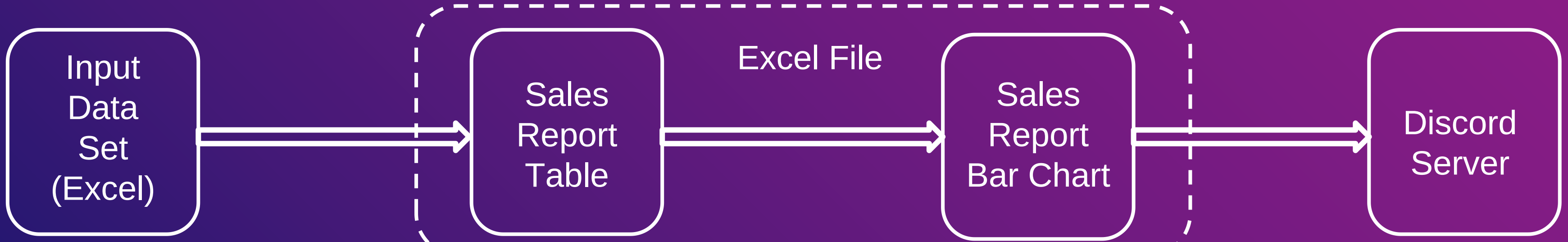


	A	B	C	D	E	F	G	H	I	J	K
1	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date
2	750-67-8428	A	Yangon	Member	Female	Health and beauty	74,69	7	26,1415	548,9715	1/5/2019
3	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessor	15,28	5	3,82	80,22	3/8/2019
4	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46,33	7	16,2155	340,5255	3/3/2019
5	123-19-1176	A	Yangon	Member	Male	Health and beauty	58,22	8	23,288	489,048	1/27/2019
6	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86,31	7	30,2085	634,3785	2/8/2019
7	699-14-3026	C	Naypyitaw	Normal	Male	Electronic accessor	85,39	7	29,8865	627,6165	3/25/2019
8	355-53-5943	A	Yangon	Member	Female	Electronic accessor	68,84	6	20,652	433,692	2/25/2019
9	315-22-5665	C	Naypyitaw	Normal	Female	Home and lifestyle	73,56	10	36,78	772,38	2/24/2019
10	665-32-9167	A	Yangon	Member	Female	Health and beauty	36,26	2	3,626	76,146	1/10/2019
11	692-92-5582	B	Mandalay	Member	Female	Food and beverage	54,84	3	8,226	172,746	2/20/2019
12	351-62-0822	B	Mandalay	Member	Female	Fashion accessories	14,48	4	2,896	60,816	2/6/2019

Supermarket Sales in 2019 as Input Data Set

Python programming for automate report of a sales data set. Applicable Python libraries:

1. pandas
2. openpyxl (interface python - excel)
3. discord

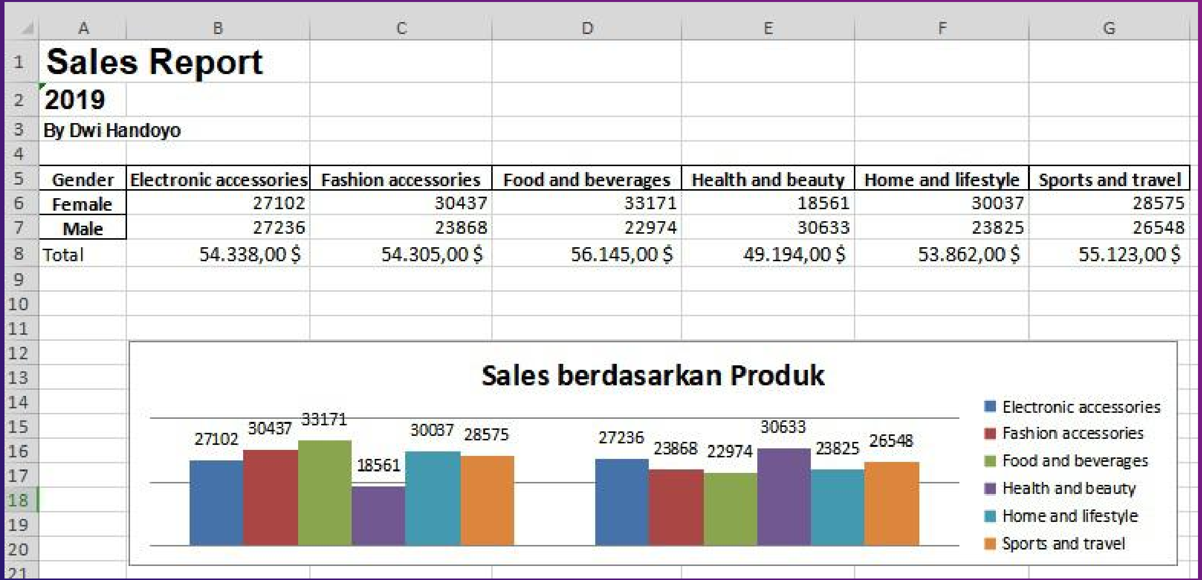


1. Extract table from input data set by using pivot table

2. Create graphic (barchart) of the report

3. Send report table and chart to Discord

Output of Automated Data Reporting by Using Python



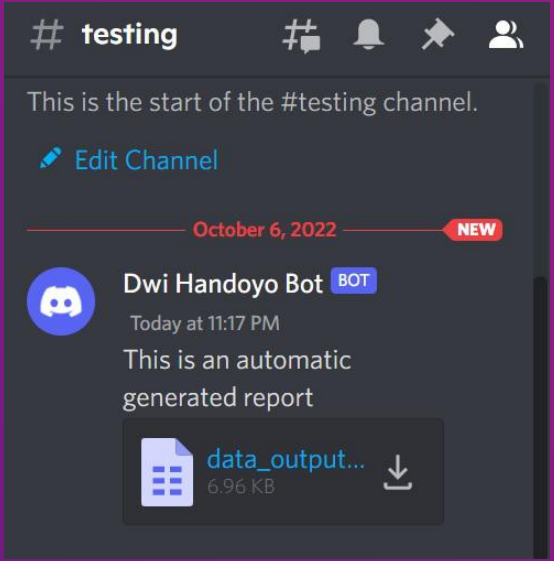
Sales Report

2019

By Dwi Handoyo

Gender	Electronic accessories	Fashion accessories	Food and beverages	Health and beauty	Home and lifestyle	Sports and travel
Female	27102	30437	33171	18561	30037	28575
Male	27236	23868	22974	30633	23825	26548
Total	54.338,00 $	54.305,00 $	56.145,00 $	49.194,00 $	53.862,00 $	55.123,00 $

Report in excel, consists of Table and Graphic (Bar Chart)



This is the start of the #testing channel.

October 6, 2022

Dwi Handoyo Bot

Today at 11:17 PM

This is an automatic generated report

data_output...
6.96 KB

Report file delivered at Discord

THANK YOU