



TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

BAIT3003 Data Warehouse Technology (202401)

2023/2024

Student 1	:	ONG WENG KAI (22WMR03309)
Student 2	:	THONG CHENG HOW (22WMR03154)
Student 3	:	RYAN KHO YUEN THIAN (22WMR04097)
Student 4	:	YONG ZEE LIN (22WMR03770)
Programme	:	Bachelor of Computer Science in Data Science
Tutorial Group	:	G2
Tutor's name	:	MR CHOONG YUN LOONG

Task No.	Task Description	Weightage	Criteria	Ratings	Marks	CLO
1	Design of Data warehouse (logical design)	5%	<ul style="list-style-type: none"> Include the relevant dimensions. Include the correct measures in the fact table. 	<ul style="list-style-type: none"> Excellent (5) Good (4) Moderate (2-3) Poor (0-1) 		1
	Design of Data warehouse (physical design)	15%	<ul style="list-style-type: none"> Create TABLE statements Appropriate data types and size of attributes Proper Integrity constraints 	<ul style="list-style-type: none"> Excellent (13-15) Good (10-12) Moderate (6-9) Poor (0-5) 		1
2	ETL (initial loading)	20%	<ul style="list-style-type: none"> VIEWS, SELECT,INSERT,PROCEDURES for each of the dimensions and fact table. Variety of techniques necessary to achieve the correct data loading 	<ul style="list-style-type: none"> Excellent (18-20) Good (14-17) Moderate (9-13) Poor (0-8) 		1
	ETL (subsequent loading)	20%	<ul style="list-style-type: none"> VIEWS, SELECT,INSERT,PROCEDURES for each of the dimensions and fact table. Logic to scrub dirty data 	<ul style="list-style-type: none"> Excellent (18-20) Good (15-17) Moderate (9-14) Poor (0-8) 		1
3	*Business Analytic queries design (Individual marks awarded))	30%	<ul style="list-style-type: none"> Clear and proper identification of information needs Flexible query to cater for variety of inputs, use of multiple tables Meaningful report handlings Data values formatted accordingly 	<ul style="list-style-type: none"> Excellent (25-30) Good (16-24) Moderate (9-15) Poor (0-8) 		3
4	Assignment Report	10%	<ul style="list-style-type: none"> Comprehensive coverage Quality of report presented All tasks numbered, header / footer used, proper formatting 	<ul style="list-style-type: none"> Excellent (9-10) Good (7-8) Moderate (4-6) Poor (0-3) 		1

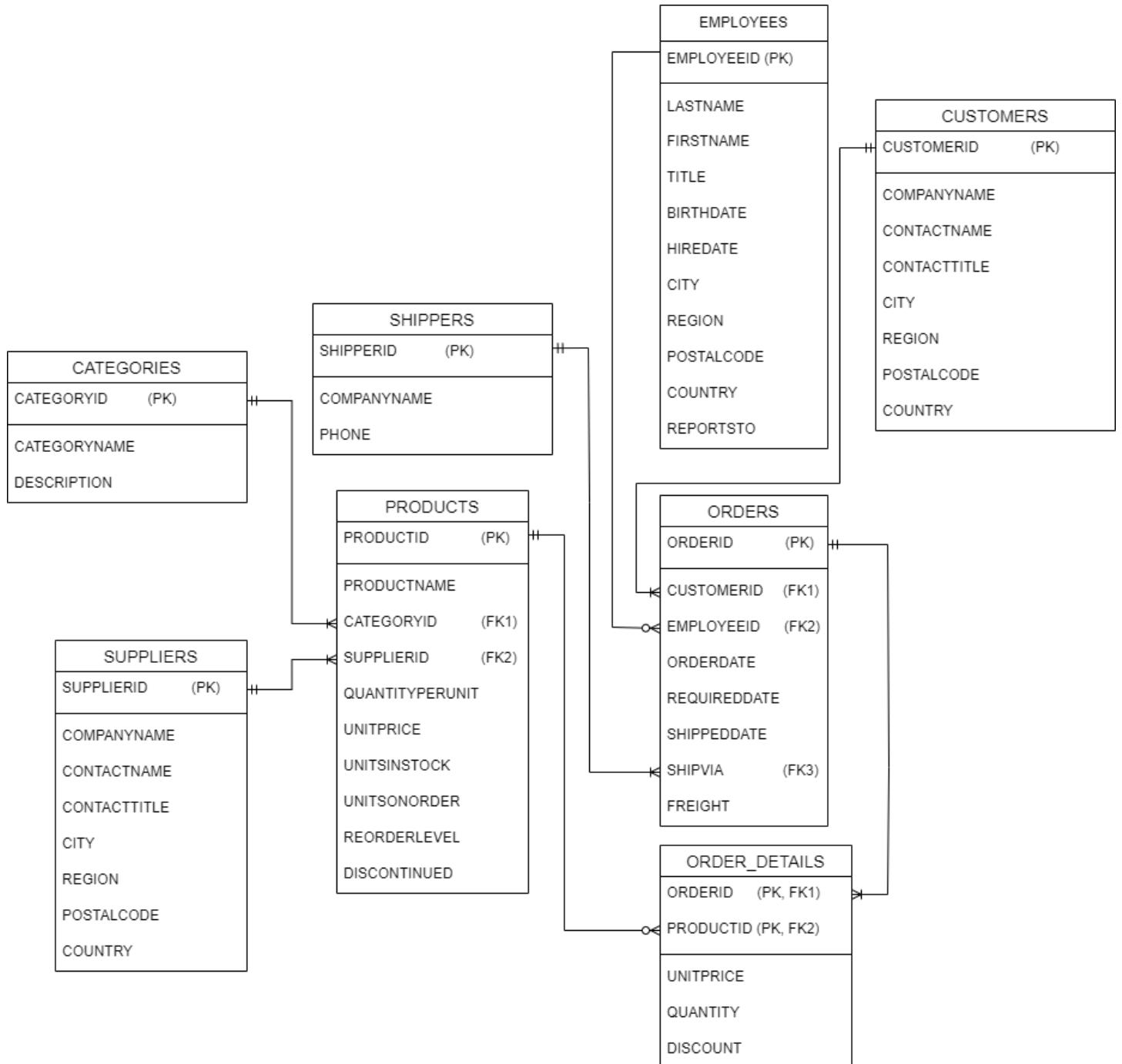
Group Member	Task 3 Marks	Total marks
Ong Weng Kai (22WMR03309)		
Thong Cheng How (22WMR03154)		
Ryan Kho Yuen Thian (22WMR04097)		
Yong Zee Lin (22WMR03770)		

Chapter 1 Design of Data Warehouse	3
1.1 Logical Design	3
1.1.1 Original Database (Entity Relationship Diagram)	3
1.1.2 Star Schema Dimension and Fact Tables	4
1.2 Physical Design	5
1.2.1 Dimension Tables	5
1.2.2 Fact Table	8
Chapter 2 Extract, Transform, Load Process	9
2.1 Script for initial loading	9
2.2 Script for Subsequent loading	14
2.3 Type 2 SCD maintenance	26
2.3.1 Update the ExpirationDate and CurrentRowIndicator	26
2.3.2 Insert New Row	29
Chapter 3 Business Analytics Reports	32
3.1 Ong Weng Kai	32
3.1.1 Generating quarterly sales for a given year and comparing it with the previous year's total sales	32
3.1.2 Customizable Annual Sales and Profitability Report	38
3.1.3 Dynamic Report: Monthly sales performance and growth analysis	44
3.2 Thong Cheng How	49
3.2.1 Top Ordering Companies Analysis By City In Year 2022 and 2023	49
3.2.2 Employee performance report comparison year 2022 vs year 2023	55
3.2.3 Best Performing Product Analysis Shipped by Suppliers In Year 2022 and comparison in year 2023	61
3.3 Ryan Kho Yuen Thian	66
3.3.1 For every month of two given years, calculate total orders, sales amount, average sales/order & differences for all or a given country	66
3.3.2 Sales Distribution by Country and Region for 2 Given Years for Comparison (includes Ranking)	72
3.3.3 Top 5 Products in each Country for 2 Given Years for comparison (includes their quarterly & yearly sales)	77
3.4 Yong Zee Lin	84
3.4.1 Yearly Shipper Performance Report	84
3.4.2 Top 10 Customer Lifetime Value in 2023	88
3.4.3 Product Category Sales in 2023	91

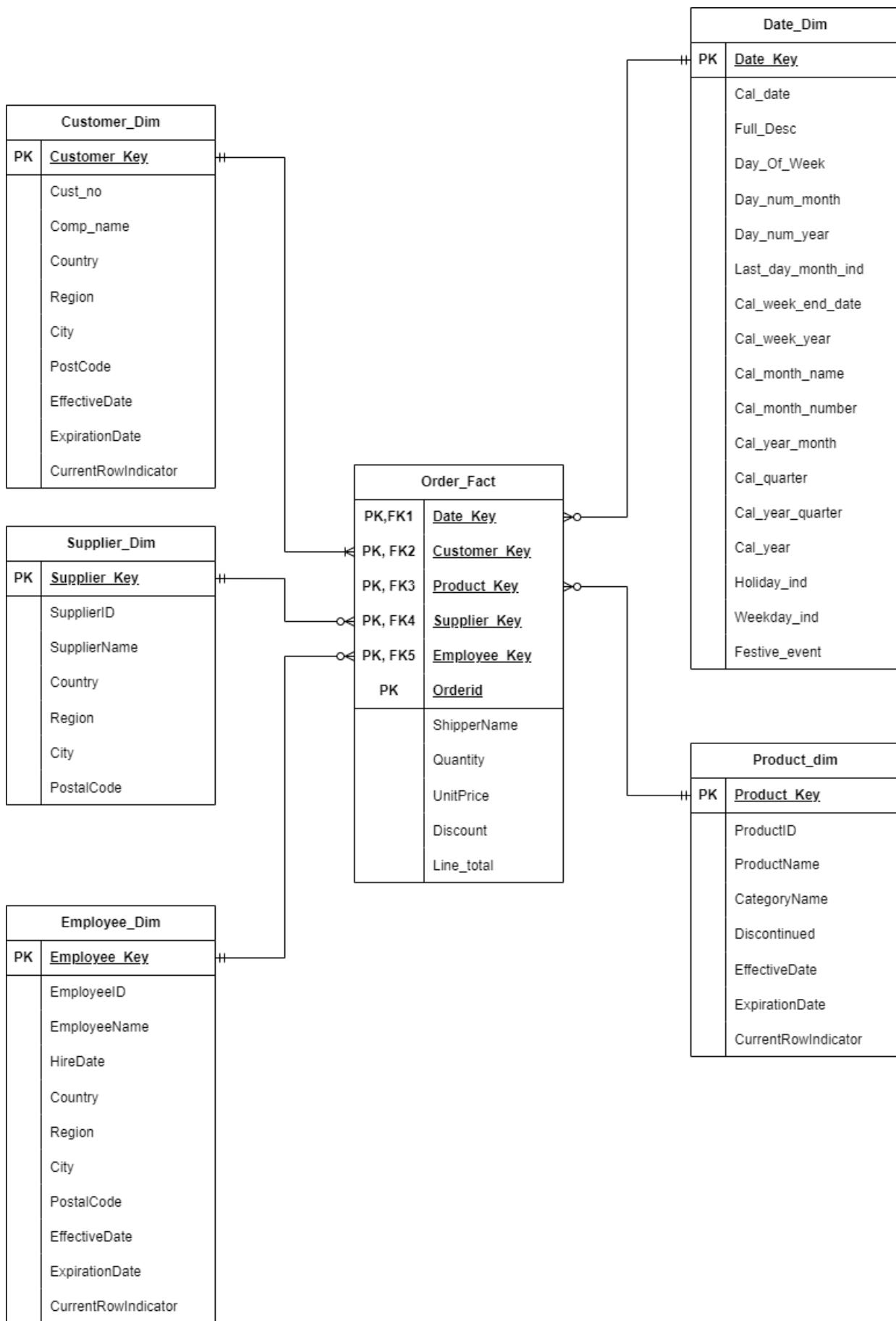
Chapter 1 Design of Data Warehouse

1.1 Logical Design

1.1.1 Original Database (Entity Relationship Diagram)



1.1.2 Star Schema Dimension and Fact Tables



1.2 Physical Design

1.2.1 Dimension Tables

i) Date Dimension Table

```

create table Date_Dim
(date_key          number not null check(date_key >= 100001),
cal_date          date,
full_desc         varchar(45),
day_of_week       number(1),
day_num_month    number(2),
day_num_year     number(3),
last_day_month_ind char(1),
cal_week_end_date date,
cal_week_year    number(2),
cal_month_name   varchar(9),
cal_month_number number(2), -- 1 to 12
cal_year_month   char(7), -- e.g '2024-01'
cal_quarter      char(2),
cal_year_quarter char(7), -- e.g '2024-Q1'
cal_year          number(4),
holiday_ind      char(1),
weekday_ind      char(1),
festive_event    varchar(45), -- 'CNY', 'RAYA', etc.
primary key(date_key),
CONSTRAINT check_day_of_week CHECK (day_of_week BETWEEN 1 AND 7),
CONSTRAINT check_day_num_month CHECK (day_num_month BETWEEN 1 AND 31),
CONSTRAINT check_day_num_year CHECK (day_num_year BETWEEN 1 AND 366),
CONSTRAINT check_last_day_month_ind CHECK (last_day_month_ind IN ('Y', 'N')),
CONSTRAINT check_cal_month_number CHECK (cal_month_number BETWEEN 1 AND 12),
CONSTRAINT check_cal_week_year CHECK (cal_week_year BETWEEN 1 AND 53),
CONSTRAINT check_cal_quarter CHECK (cal_quarter IN ('Q1', 'Q2', 'Q3', 'Q4')),
CONSTRAINT check_cal_month_name CHECK (TRIM(UPPER(cal_month_name)) IN
('JANUARY', 'FEBRUARY', 'MARCH', 'APRIL', 'MAY', 'JUNE', 'JULY', 'AUGUST',
'SEPTEMBER', 'OCTOBER', 'NOVEMBER', 'DECEMBER')),
CONSTRAINT check_holiday_ind CHECK (holiday_ind IN ('Y', 'N')),
CONSTRAINT check_weekday_ind CHECK (weekday_ind IN ('Y', 'N')),
CONSTRAINT check_cal_year_quarter CHECK (REGEXP_LIKE(cal_year_quarter,
'^[0-9]{4}-Q[1-4]$')),
CONSTRAINT check_cal_year_month CHECK (REGEXP_LIKE(TRIM(cal_year_month),
'^[0-9]{4}-(0?[1-9]|1[0-2])$'))
);

```

ii) Product Dimension Table (Type 2 SCD)

```

CREATE table Product_dim
(
    product_key          number not null, -- for type 2 SCD
    productID           number not null,
    productName          varchar(40) not null,
    categoryName         varchar(15) not null,
    discontinued        number(1) not null check(discontinued IN (0,1)),
    effectiveDate       date Default '01/01/2015', -- for type 2 SCD
    expirationDate      date Default '31/12/9999', -- for type 2 SCD
    currentRowIndicator CHAR(1) DEFAULT 'Y' NOT NULL CHECK
    (UPPER(currentRowIndicator) IN ('Y', 'N')), -- for type 2 SCD
    primary key(product_key),
    CONSTRAINT check_productKey CHECK (product_key >= 1001),
    CONSTRAINT check_endDateProduct CHECK (expirationDate >= effectiveDate)
);

```

iii) Supplier Dimension Table

```

CREATE TABLE Supplier_dim (
    Supplier_key          number not null,
    SupplierID            number NOT NULL,
    supplierName          VARCHAR(40) NOT NULL,
    Country               VARCHAR(15) CHECK (NOT REGEXP_LIKE(Country,
    '[[[:digit:]]'))),
    Region                VARCHAR(15),
    City                  VARCHAR(15),
    PostalCode             VARCHAR(10),
    PRIMARY KEY (Supplier_key),
    CONSTRAINT check_supplierKey CHECK (supplier_key >= 1001)
);

```

iv) Employee Dimension Table (Type 2 SCD)

```

CREATE TABLE Employee_dim (
    Employee_key          number not null,
    EmployeeID            number not null,
    EmployeeName          VARCHAR(40) not null CHECK (NOT
    REGEXP_LIKE(EmployeeName, '[[:digit:]]')),
    HireDate               DATE,
    Country                VARCHAR(15) CHECK (NOT REGEXP_LIKE(Country,
    '[[[:digit:]]'))),
    Region                 VARCHAR(15),
    City                   VARCHAR(15),
    PostalCode              VARCHAR(10),
    EffectiveDate          Date Default '01/01/2015',
    ExpirationDate         Date Default '31/12/9999',
    currentRowIndicator CHAR(1) DEFAULT 'Y' NOT NULL CHECK
    (UPPER(currentRowIndicator) IN ('Y', 'N')),
    PRIMARY KEY (Employee_key),
    CONSTRAINT check_employeeKey CHECK (employee_key >= 1001),
    CONSTRAINT check_endDateEmployee CHECK (expirationDate >= effectiveDate)
);

```

v) Customer Dimension Table (Type 2 SCD)

```

create table customer_dim
(customer_key          number not null CHECK (Customer_Key > 1000) ,
cust_no               number not null,
comp_name              varchar(40) not null,
country                VARCHAR(15) CHECK (NOT REGEXP_LIKE(Country,
'[[[:digit:]]'))),
region                 varchar(15),
city                   varchar(15),
postcode                VARCHAR(10),
effectiveDate          date not null,
expirationDate         date Default '31/12/9999',
currentRowIndicator CHAR(1) DEFAULT 'Y' NOT NULL CHECK
(UPPER(currentRowIndicator) IN ('Y', 'N')),
primary key(customer_key),
CONSTRAINT check_customerKey CHECK (customer_key >= 1001),
CONSTRAINT check_endDateCustomer CHECK (expirationDate >= effectiveDate)
);

```

1.2.2 Fact Table

i) Fact Table (the grain is an individual product in an order transaction)

```
create table order_fact
(date_key      number not null,
customer_key  number not null,
product_key   number not null,
supplier_key  number not null,
employee_key  number not null,
orderid       number not null,
shipperName   varchar(40) not null,
quantity      number(3) not null,
unitPrice     number(6,2) not null,
discount      number(3,2) not null,
line_total    number(9,2) not null,
primary key (date_key, customer_key, product_key, supplier_key, employee_key,
orderid),
foreign key (date_key) references Date_dim(date_key),
foreign key (customer_key) references Customer_dim(customer_key),
foreign key (product_key) references Product_dim(product_key),
foreign key (supplier_key) references Supplier_dim(supplier_key),
foreign key (employee_key) references Employee_dim(employee_key)
);
```

Chapter 2 Extract, Transform, Load Process

2.1 Script for initial loading

i) For Date_Dim

```

drop sequence date_seq;

create sequence date_seq
  start with 100001
  increment by 1;

declare
  StartDate date:=to_date('01/01/2015','dd/mm/yyyy');
  EndDate   date:=to_date('31/12/2024','dd/mm/yyyy');
  v_cal_date          date;
  v_full_desc         varchar(45);
  v_day_of_week        number(1);
  v_day_num_month      number(2);
  v_day_num_year       number(3);
  v_last_day_month_ind char(1);
  v_cal_week_end_date date;
  v_cal_week_year      number(2);
  v_cal_month_name     varchar(9);
  v_cal_month_number   number(2);
  v_cal_year_month     char(7);
  v_cal_quarter        char(2);
  v_cal_year_quarter   char(7);
  v_cal_year           number(4);
  v_holiday_ind        char(1);
  v_weekday_ind         char(1);
  v_festive_event       varchar(45);

-- counter number:=0;
begin
  while (StartDate<=EndDate) loop
    -- counter:=counter+1;
    v_cal_date:=StartDate;
    v_full_desc:=to_char(StartDate,'Year Month Day DD');
    v_day_of_week:=to_char(StartDate,'D');
    v_day_num_month:=to_char(StartDate,'DD');
    v_day_num_year:=to_char(StartDate,'DDD');

    if (StartDate=last_day(StartDate)) then
      v_last_day_month_ind:='Y';
    else
      v_last_day_month_ind:='N';
    end if;
  end loop;
end;
  
```

```
v_cal_week_end_date:= StartDate+(7-v_day_of_week);

v_cal_week_year:=to_char(StartDate,'IW');
v_cal_month_name:=to_char(StartDate,'MONTH');
v_cal_month_number:=to_char(StartDate,'MM');
v_cal_year:=to_char(StartDate,'YYYY');
v_cal_year_month:=v_cal_year||'-'||v_cal_month_number;

if (v_cal_month_number<=3) then
    v_cal_quarter:='Q1';
elsif (v_cal_month_number<=6) then
    v_cal_quarter:='Q2';
elsif (v_cal_month_number<=9) then
    v_cal_quarter:='Q3';
else
    v_cal_quarter:='Q4';
end if;

v_cal_year_quarter:=v_cal_year||'-'||v_cal_quarter;
v_holiday_ind:='N';

if (v_day_of_week BETWEEN 2 AND 6) then
    v_weekday_ind:='Y';
else
    v_weekday_ind:='N';
end if;

v_festive_event:=NULL;

insert into Date_Dim values(
    date_seq.nextval,
    v_cal_date,
    v_full_desc,
    v_day_of_week,
    v_day_num_month,
    v_day_num_year,
    v_last_day_month_ind,
    v_cal_week_end_date,
    v_cal_week_year,
    v_cal_month_name,
    v_cal_month_number,
    v_cal_year_month,
    v_cal_quarter,
    v_cal_year_quarter,
    v_cal_year,
    v_holiday_ind,
    v_weekday_ind,
    v_festive_event);
```

```
    StartDate:=StartDate+1;
  end loop;
-- dbms_output.put_line('Count is '||counter);
end;
/

update date_dim
set holiday_ind='Y'
where cal_date = to_date('01/01/2024', 'dd/mm/yyyy');

update date_dim
set festive_event='CNY'
where cal_date = to_date('10/02/2024', 'dd/mm/yyyy');
```

ii) For Product_Dim

```
drop sequence prod_dim_seq;

create sequence prod_dim_seq
  start with 1001
  increment by 1;

insert into Product_dim (product_key, productID, productName, categoryName,
discontinued)
select prod_dim_seq.nextval,
       P.productID,
       UPPER(P.productName),
       UPPER(C.categoryName),
       P.discontinued
  from products      P
 JOIN categories C on P.categoryID=C.categoryID;

commit;
```

iii) For Supplier_Dim

```
drop sequence supplier_dim_seq;

create sequence supplier_dim_seq
    start with 1001
    increment by 1;

insert into supplier_dim (Supplier_key, SupplierID, SupplierName, Country,
Region, City, PostalCode)
select supplier_dim_seq.nextval,
       supplierid,
       UPPER(companyname),
       UPPER(country),
       UPPER(region),
       UPPER(City),
       UPPER(postalcode)
from suppliers;

commit;
```

iv) For Employee_Dim

```
drop sequence employee_dim_seq;

create sequence employee_dim_seq
    start with 1001
    increment by 1;

insert into Employee_dim (Employee_key, EmployeeID, EmployeeName, HireDate,
Country, Region, City, PostalCode)
select employee_dim_seq.nextval,
       employeeid,
       firstname || ' ' || lastname,
       hiredate,
       UPPER(country),
       UPPER(region),
       UPPER(city),
       UPPER(postalcode)
from employees;

commit;
```

v) For Customer_Dim

```

drop sequence cust_dim_seq;

create sequence cust_dim_seq
  start with 1001
  increment by 1;

INSERT INTO customer_dim (customer_key, cust_no, comp_name, country, region,
city, postcode, effectiveDate)
SELECT cust_dim_seq.nextval,
       cust_no,
       comp_name,
       country,
       region,
       city,
       postcode,
       (SELECT MIN(OrderDate)
        FROM n_orders no
        WHERE no.Cust_ID = nc.cust_no)
FROM n_cust nc;

commit;

```

vi) For Order_Fact

```

INSERT INTO order_fact
SELECT D.date_key,
       C.customer_key,
       P.product_key,
       SP.supplier_key,
       E.employee_key,
       ORD.order_id,
       SH.companyName,
       OD.quantity,
       OD.unitPrice,
       OD.discount,
       OD.quantity * OD.unitprice * (1 - OD.discount)
FROM n_orders ORD
JOIN n_order_details OD ON ORD.order_ID = OD.orderID
JOIN Products Prod ON OD.productID = Prod.productID
JOIN Date_dim D ON ORD.orderDate = D.cal_date
JOIN (
       SELECT cust_no, customer_key, effectiveDate, expirationDate
       FROM Customer_dim
      ) C ON ORD.cust_id = C.cust_no
      AND ORD.orderDate BETWEEN C.effectiveDate AND C.expirationDate
JOIN (
       SELECT employeeID, employee_key, effectiveDate, expirationDate
       FROM Employee_dim
      )

```

```

) E ON ORD.staffid = E.employeeID
    AND ORD.orderDate BETWEEN E.effectiveDate AND E.expirationDate
JOIN Supplier_dim SP on Prod.supplierID=SP.supplierID
JOIN (
    SELECT productID, product_key, effectiveDate, expirationDate
    FROM Product_dim
) P ON OD.productID = P.productID
    AND ORD.orderDate BETWEEN P.effectiveDate AND P.expirationDate
JOIN Shippers SH ON ORD.shipvia = SH.shipperID;

```

2.2 Script for Subsequent loading

i) For Date_Dim (Subsequent loading logic)

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY';
```

```

declare
    StartDate date;
    EndDate   date;
    v_cal_date          date;
    v_full_desc        varchar(45);
    v_day_of_week       number(1);
    v_day_num_month    number(2);
    v_day_num_year     number(3);
    v_last_day_month_ind char(1);
    v_cal_week_end_date date;
    v_cal_week_year    number(2);
    v_cal_month_name   varchar(9);
    v_cal_month_number number(2);
    v_cal_year_month   char(7);
    v_cal_quarter      char(2);
    v_cal_year_quarter char(7);
    v_cal_year          number(4);
    v_holiday_ind      char(1);
    v_weekday_ind       char(1);
    v_festive_event     varchar(45);

-- counter number:=0;
begin
    select TRUNC(max(cal_date))+1
    into StartDate
    from Date_Dim;
    EndDate := ADD_MONTHS(StartDate, 108);
    while (StartDate<=EndDate) loop
-- counter:=counter+1;
        v_cal_date:=StartDate;
        v_full_desc:=to_char(StartDate,'Year Month Day DD');
        v_day_of_week:=to_char(StartDate,'D');

```

```
v_day_num_month:=to_char(StartDate,'DD');
v_day_num_year:=to_char(StartDate,'DDD');

if (StartDate=last_day(StartDate)) then
    v_last_day_month_ind:='Y';
else
    v_last_day_month_ind:='N';
end if;

v_cal_week_end_date:= StartDate+(7-v_day_of_week);

v_cal_week_year:=to_char(StartDate,'IW');
v_cal_month_name:=to_char(StartDate,'MONTH');
v_cal_month_number:=to_char(StartDate,'MM');
v_cal_year:=to_char(StartDate,'YYYY');
v_cal_year_month:=v_cal_year||'-'||v_cal_month_number;

if (v_cal_month_number<=3) then
    v_cal_quarter:='Q1';
elsif (v_cal_month_number<=6) then
    v_cal_quarter:='Q2';
elsif (v_cal_month_number<=9) then
    v_cal_quarter:='Q3';
else
    v_cal_quarter:='Q4';
end if;

v_cal_year_quarter:=v_cal_year||'-'||v_cal_quarter;
v_holiday_ind:='N';

if (v_day_of_week BETWEEN 2 AND 6) then
    v_weekday_ind:='Y';
else
    v_weekday_ind:='N';
end if;

v_festive_event:=NULL;

insert into Date_Dim values (
    date_seq.nextval,
    v_cal_date,
    v_full_desc,
    v_day_of_week,
    v_day_num_month,
    v_day_num_year,
    v_last_day_month_ind,
    v_cal_week_end_date,
    v_cal_week_year,
    v_cal_month_name,
```

```

        v_cal_month_number,
        v_cal_year_month,
        v_cal_quarter,
        v_cal_year_quarter,
        v_cal_year,
        v_holiday_ind,
        v_weekday_ind,
        v_festive_event);

        StartDate:=StartDate+1;
    end loop;
-- dbms_output.put_line('Count is '||counter);
end;
/

```

ii) For Product_Dim (Subsequent Loading + Dirty Data Scrubbing)

```

INSERT INTO Products VALUES (78, 'Simon', 1, 1, '10 boxes x 20 bags', 18, 39, 0, 10, 0);
INSERT INTO Products VALUES (79, 'Guava', 1, 1, '10 boxes x 20 bags', 18, 39, 0, 10, 0);

```

```

-- 1) Staging Area
drop table product_staging_area;

CREATE TABLE Product_Staging_Area AS
SELECT P.productID,
       UPPER(P.productName) as ProductName,
       UPPER(C.categoryName) as CategoryName,
       P.discontinued
FROM products P
JOIN categories C on P.categoryID=C.categoryID;

```

```

--2) VIEW
create or replace view count_product_data as
select count(*) as Num_Records_Before_Scrub
from product_staging_area;

```

```
select * from count_product_data;
```

```

-- 3) Scrubbing Dirty Data
CREATE OR REPLACE PROCEDURE Scrub_Product_Staging_Area AS
BEGIN
    -- Remove records where either productName or categoryName is NULL
    DELETE FROM Product_Staging_Area
    WHERE productName IS NULL OR categoryName IS NULL OR discontinued IS NULL;

```

```
FOR rec IN (SELECT * FROM Product_Staging_Area) LOOP
    IF rec.discontinued NOT IN (0, 1) THEN
        rec.discontinued := 0;
    END IF;

    UPDATE Product_Staging_Area
    SET productName = UPPER(productName),
        categoryName = UPPER(categoryName),
        discontinued = rec.discontinued;
END LOOP;
COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/

exec Scrub_Product_Staging_Area;

-- 4) For View
create or replace view count_product_data as
select count(*) as Num_Records_After_Scrub
from product_staging_area;

select * from count_product_data;

-- 5) For adding cleaned new records into Dimension table)
CREATE OR REPLACE PROCEDURE Insert_CProduct_Dim AS
BEGIN
    insert into Product_Dim (product_key, productID, productName, categoryName,
discontinued)
    select prod_dim_seq.nextval,C.*
    from Product_Staging_Area C
    where C.productID NOT IN (select distinct productID from product_dim);
END;
/

exec insert_Cproduct_dim;
```

iii) For Supplier_Dim (Subsequent Loading + Dirty Data Scrubbing)

```

INSERT INTO Suppliers VALUES (30, 'KongTiao', 'Eliot Brandt', 'Sales Representative', 'Annecy', NULL, '74000', 'France');
INSERT INTO Suppliers VALUES (31, 'KGB', 'Consola Ray', 'Accounting Manager', 'Ste-Hyacinthe', 'Quebec', 'J2S 7S8', 'Canada');

-- 1) Staging Area
drop table supplier_staging_area;

CREATE TABLE Supplier_Staging_Area AS
SELECT
    supplierid,
    companyname,
    country,
    region,
    city,
    postalcode
FROM suppliers;

--2) VIEW
create or replace view count_supplier_data as
select count(*) as Num_Records_Before_Scrub
from supplier_staging_area;

select * from count_supplier_data;

-- 3) Scrubbing Dirty Data
CREATE OR REPLACE PROCEDURE clean_supplier_data AS
BEGIN
    DELETE FROM supplier_staging_area
    WHERE city is null and region is null and postalcode is null and country is null;
    FOR supplier_rec in (SELECT * FROM supplier_staging_area) LOOP

        -- Check for numbers in Country, Region, and City
        IF REGEXP_LIKE(supplier_rec.country, '[0-9]') THEN
            DBMS_OUTPUT.PUT_LINE('Error: Country cannot contain numbers');
            delete from supplier_staging_area
            where supplierid = supplier_rec.supplierid;
            continue;
        END IF;

        IF REGEXP_LIKE(supplier_rec.region, '[0-9]') THEN
            DBMS_OUTPUT.PUT_LINE('Error: Region cannot contain numbers');
            delete from supplier_staging_area
            where supplierid = supplier_rec.supplierid;
            continue;
        END IF;
    END IF;

```

```
END IF;

IF REGEXP_LIKE(supplier_rec.city, '[0-9]') THEN
    DBMS_OUTPUT.PUT_LINE('Error: City cannot contain numbers');
    delete from supplier_staging_area
    where supplierid = supplier_rec.supplierid;
    continue;
END IF;

-- Clean Country, Region, City, and Postal Code (convert to uppercase)
UPDATE supplier_staging_area
Set country = UPPER(supplier_rec.country),
region = UPPER(supplier_rec.region),
city = UPPER(supplier_rec.city),
postalcode = UPPER(supplier_rec.postalcode),
companynname = INITCAP(REGEXP_REPLACE(supplier_rec.companynname,
'[^a-zA-Z ]', ''))
WHERE supplierid = supplier_rec.supplierid;

END LOOP;
COMMIT;
EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/

exec clean_supplier_data;

-- 4) For View
create or replace view count_supplier_data as
select count(*) as Num_Records_After_Scrub
from supplier_staging_area;

select * from count_supplier_data;

-- 5) For adding cleaned new records into Dimension table
CREATE OR REPLACE PROCEDURE Insert_CSupplier_Dim AS
BEGIN
    insert into Supplier_Dim (Supplier_key, SupplierID, SupplierName, Country,
Region, City, PostalCode)
    select supplier_dim_seq.nextval, S.*
    from supplier_staging_area S
    where S.supplierid not in (select distinct supplierid from supplier_dim);
END;
/
```

```
exec Insert_CSupplier_Dim;
```

iv) For Employee_Dim (Subsequent Loading + Dirty Data Scrubbing)

```
alter session set NLS_DATE_FORMAT='YYYY-MM-DD';
```

```
INSERT INTO Employees VALUES (10, 'Keanu', 'Reeve', 'Inside Sales Coordinator',
'1958-01-09', '1994-03-05', 'Seattle', 'WA', '98105', 'USA', 2);
INSERT INTO Employees VALUES (11, 'Kean', 'Wong', 'Sales Representative',
'1966-01-27', '1994-11-15', 'London', NULL, 'WG2 7LT', 'UK', 5);
```

--1) Staging Area

```
drop table employee_staging_area;
```

```
create table employee_staging_area as
select Employeeid,
       firstName || ' ' || lastName as EmployeeName,
       hiredate,
       upper(country) as Country,
       upper(region) as Region,
       upper(city) as City,
       upper(postalcode) as PostalCode
  from employees;
```

--2) VIEW

```
create or replace view count_emp_data as
select count(*) as Num_Records_Before_Scrub
  from employee_staging_area;
```

```
select * from count_emp_data;
```

```
set serveroutput on
```

--3) Scrubbing Dirty Data

```
CREATE OR REPLACE PROCEDURE clean_employee_data AS
BEGIN
```

```
    DELETE FROM employee_staging_area
```

```
    WHERE country is null and region is null and city is null and postalcode is
null;
```

```
    FOR emp_rec in (SELECT * FROM employee_staging_area) LOOP
```

```
        IF REGEXP_LIKE(emp_rec.country, '[0-9]') THEN
```

```
            DBMS_OUTPUT.PUT_LINE('Error: Country cannot contain numbers');
```

```
            delete from employee_staging_area
```

```
            where employeeid = emp_rec.employeeid;
```

```
            continue;
```

```
        END IF;
```

```

IF REGEXP_LIKE(emp_rec.region, '[0-9]') THEN
    DBMS_OUTPUT.PUT_LINE('Error: Region cannot contain numbers');
    delete from employee_staging_area
    where employeeid = emp_rec.employeeid;
    continue;
END IF;

IF REGEXP_LIKE(emp_rec.city, '[0-9]') THEN
    DBMS_OUTPUT.PUT_LINE('Error: City cannot contain numbers');
    delete from employee_staging_area
    where employeeid = emp_rec.employeeid;
    continue;
END IF;

-- Clean country, region, city and postalcode
UPDATE employee_staging_area
set country = upper(emp_rec.country),
    region = upper(emp_rec.region),
    city = upper(emp_rec.city),
    postalcode = upper(emp_rec.postalcode),
    employeename = INITCAP(REGEXP_REPLACE(emp_rec.employeename, '[^a-zA-Z]', '')) )
    where employeeid = emp_rec.employeeid;

END LOOP;
COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/

exec clean_employee_data;

--4) VIEW
create or replace view count_emp_data as
select count(*) as Num_Records_After_Scrub
from employee_staging_area;

select * from count_emp_data;

--5)
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY';
CREATE OR REPLACE PROCEDURE Insert_CEmp_Dim AS
BEGIN

```

```

    insert into Employee_dim (Employee_key, EmployeeID, EmployeeName, HireDate,
Country, Region, City, PostalCode)
    select employee_dim_seq.nextval, E.*
    from employee_staging_area E
    where E.employeeid not in (select distinct employeeid from employee_dim);
END;
/
exec Insert_CEmp_Dim;

```

v) For Customer_Dim (Subsequent Loading + Dirty Data Scrubbing)

```

INSERT INTO N_cust VALUES (105825, 'ATE', 'Simon Mark', 'Worker', 'Berlin',
NULL, '12209', 'Germany');
INSERT INTO N_cust VALUES (105826, 'WHO', 'Popeye', 'Worker', 'Berlin', NULL,
'12209', 'Germany');
INSERT INTO N_cust VALUES (105827, 'WAKANDA', NULL, NULL, NULL, NULL, NULL,
NULL);
INSERT INTO N_cust VALUES (105827, 'SIMON', NULL, NULL, NULL, NULL, NULL,
NULL);

--1) Staging Area
drop table customer_staging_area;

CREATE TABLE customer_staging_area AS
SELECT cust_no,
       comp_name,
       country,
       region,
       city,
       postcode
FROM n_cust;

--2) VIEW
create or replace view count_cust_data as
select count(*) as Num_Records_Before_Scrub
from customer_staging_area;

select * from count_cust_data;

```

```
--3) Scrubbing Dirty Data
CREATE OR REPLACE PROCEDURE Scrub_Customer_Staging AS
BEGIN
    DELETE FROM Customer_Staging_Area
    WHERE city is null and region is null and postcode is null and country is
null;
    FOR rec IN (SELECT * FROM Customer_Staging_Area) LOOP
        IF rec.comp_name is NOT NULL THEN
            rec.comp_name := UPPER(TRIM(rec.comp_name));
        END IF;

        IF rec.city IS NOT NULL THEN
            rec.city := UPPER(TRIM(rec.city));
        END IF;

        IF rec.region IS NOT NULL THEN
            rec.region := UPPER(TRIM(rec.region));
        END IF;

        IF rec.postcode IS NOT NULL THEN
            rec.postcode := UPPER(TRIM(rec.postcode));
        END IF;

        IF rec.country IS NOT NULL THEN
            rec.country := UPPER(TRIM(rec.country));
        END IF;

        -- Update the row with cleansed data
        UPDATE Customer_Staging_Area
        SET
            comp_name = rec.comp_name,
            city = rec.city,
            region = rec.region,
            postcode = rec.postcode,
            country = rec.country
        WHERE cust_no = rec.cust_no;
    END LOOP;
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
exec Scrub_Customer_Staging;
```

```
-- 4) For View
create or replace view count_cust_data as
select count(*) as Num_Records_After_Scrub
from customer_staging_area;

select * from count_cust_data;
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY';

-- 5) For adding cleaned new records into Dimension table
CREATE OR REPLACE PROCEDURE Insert_CCustomer_Dim AS
BEGIN
    insert into Customer_Dim(customer_key, cust_no, comp_name, country, region,
city, postcode, effectiveDate)
    select cust_dim_seq.nextval,
           C.*,
           SYSDATE
      from customer_staging_area C
     where c.cust_no not in (select distinct cust_no from customer_dim);
END;
/
exec insert_ccustomer_dim;
```

vi) For Order_Fact (Subsequent Loading + Dirty Data Scrubbing)

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY';

insert into n_orders values (order_seq.nextval, 100001, 1, '01/03/2024',
'10/03/2024', '05/03/2024', 4, 50);
insert into n_orders values (order_seq.nextval, 100002, 2, '03/03/2024',
'12/03/2024', '07/03/2024', 5, 70);

insert into n_order_details values (192037, 2, 19, 2, 0.1);
insert into n_order_details values (192038, 3, 10, 3, 0.1);

-- 1) Insert new records into order_fact table.
-- Only include records with valid values for freight (>0), unitprice(>0) and
quantity (>0)
-- A new record must not have the same orderid as any record currently in the
DWH
-- But the orderdate can be greater than or equal to latest orderdate in the
DWH

INSERT INTO order_fact
SELECT D.date_key,
       C.customer_key,
       P.product_key,
```

```
SP.supplier_key,
E.employee_key,
ORD.order_id,
SH.companyName,
OD.quantity,
OD.unitPrice,
OD.discount,
OD.quantity * OD.unitprice * (1 - OD.discount) AS total_price
FROM n_orders ORD
JOIN n_order_details OD ON ORD.order_ID = OD.orderID
JOIN Products Prod ON OD.productID = Prod.productID
JOIN Date_dim D ON ORD.orderDate = D.cal_date
JOIN (
    SELECT cust_no, customer_key, effectiveDate, expirationDate
    FROM Customer_dim
) C ON ORD.cust_id = C.cust_no
    AND ORD.orderDate BETWEEN C.effectiveDate AND C.expirationDate
JOIN (
    SELECT employeeID, employee_key, effectiveDate, expirationDate
    FROM Employee_dim
) E ON ORD.staffid = E.employeeID
    AND ORD.orderDate BETWEEN E.effectiveDate AND E.expirationDate
JOIN Supplier_dim SP on Prod.supplierID=SP.supplierID
JOIN (
    SELECT productID, product_key, effectiveDate, expirationDate
    FROM Product_dim
) P ON OD.productID = P.productID
    AND ORD.orderDate BETWEEN P.effectiveDate AND P.expirationDate
JOIN Shippers SH ON ORD.shipvia = SH.shipperID
WHERE ORD.order_id NOT IN (select distinct orderid from order_fact)
AND ORD.orderDate >= (SELECT MAX(cal_date)
    FROM order_fact a, date_dim b
    where a.date_key = b.date_key)
AND freight > 0
AND OD.unitprice > 0
AND OD.quantity > 0;
```

2.3 Type 2 SCD maintenance

The physical database design consists of 3 Type 2 SCD tables, which are Product_Dim, Employee_Dim and Customer_Dim.

2.3.1 Update the ExpirationDate and CurrentRowIndicator

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY';
set serveroutput on;
```

i) For Product_Dim

```
CREATE OR REPLACE PROCEDURE Insert_Product_Dim_Record(
    p_productID NUMBER,
    p_productName VARCHAR,
    p_categoryName VARCHAR,
    p_discontinued NUMBER
)
IS
BEGIN
    -- Update the expiration date and current row indicator of the previous
    record (if exists)
    UPDATE Product_dim
    SET expirationDate = TRUNC(SYSDATE)-1,      -- To expire the previous record
    1 day before the effective date of the new record
        currentRowIndicator = 'N'      -- To mark the previous record as expired
    WHERE productID = p_productID
    AND currentRowIndicator = 'Y';      -- To update only the current active
    record

    -- Insert the new record
    INSERT INTO Product_dim (product_key, productID, productName, categoryName,
    discontinued, effectiveDate, expirationDate, currentRowIndicator)
    VALUES (prod_dim_seq.nextval, p_productID, p_productName, p_categoryName,
    p_discontinued, SYSDATE, '31/12/9999', 'Y');

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Record inserted successfully');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

ii) For Employee Dim

```
CREATE OR REPLACE PROCEDURE Insert_Emp_Dim_Record(
    e_employeeid      number,
    e_employeename    varchar,
    e_hiredate        date,
    e_country         varchar,
    e_region          varchar,
    e_city            varchar,
    e_postalCode      varchar
)
IS
BEGIN
    UPDATE Employee_dim
    SET expirationDate = TRUNC(SYSDATE)-1,
        currentRowIndicator = 'N'
    WHERE employeeid = e_employeeid
    AND currentRowIndicator = 'Y';

    INSERT INTO employee_dim (Employee_key, EmployeeID, EmployeeName, HireDate,
Country,     Region,     City,     PostalCode,     EffectiveDate,     ExpirationDate,
currentRowIndicator)
    VALUES (employee_dim_seq.nextval, e_employeeid, e_employeename, e_hiredate,
e_country, e_region, e_city, e_postalCode, SYSDATE, '31/12/9999', 'Y');

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Record inserted successfully');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

iii) Customer_Dim

```
CREATE OR REPLACE PROCEDURE Insert_Cust_Dim_Record(
    c_cust_no      NUMBER,
    c_comp_name    varchar,
    c_country      varchar,
    c_region       varchar,
    c_city         varchar,
    c_postcode     varchar
)
IS
BEGIN
    UPDATE Customer_dim
    SET expirationDate = TRUNC(SYSDATE)-1,
        currentRowIndicator = 'N'
    WHERE cust_no = c_cust_no
    AND currentRowIndicator = 'Y';

    INSERT INTO customer_dim (customer_key, cust_no, comp_name, country,
region, city, postcode, effectiveDate, expirationDate, currentRowIndicator)
    VALUES (cust_dim_seq.nextval, c_cust_no, c_comp_name, c_country, c_region,
c_city, c_postcode, SYSDATE, '31/12/9999', 'Y');

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Record inserted successfully');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

2.3.2 Insert New Row

i) For Product Dim

For example, I would like to change the ProductName for ProductID 1 from "CHAI" to "CHAI BERRIES"

PRODUCT_KEY	PRODUCTID	PRODUCTNAME	CATEGORYNAME	DISCONTINUED	EFFECTIVE	EXPIRATION	C
1001	1	CHAI	BEVERAGES	0	01-JAN-15	31-DEC-99	Y
1002	2	CHANG	BEVERAGES	0	01-JAN-15	31-DEC-99	Y
1003	3	ANISEED SYRUP	CONDIMENTS	0	01-JAN-15	31-DEC-99	Y
1004	4	CHEF ANTON'S CAJUN SEASONING	CONDIMENTS	0	01-JAN-15	31-DEC-99	Y

After executing the below command:

A New Record is inserted and the ExpirationDate, EffectiveDate and CurrentRowIndicator are updated accordingly.

```
exec Insert_Product_Dim_Record(1, 'CHAI BERRIES', 'BEVERAGES', 0);
```

PRODUCT_KEY	PRODUCTID	PRODUCTNAME	CATEGORYNAME	DISCONTINUED	EFFECTIVED	EXPIRATION	C
1001	1	CHAI	BEVERAGES	0	01/01/2015	04/05/2024	N
1002	2	CHANG	BEVERAGES	0	01/01/2015	31/12/9999	Y
1003	3	ANISEED SYRUP	CONDIMENTS	0	01/01/2015	31/12/9999	Y
1004	4	CHEF ANTON'S CAJUN SEASONING	CONDIMENTS	0	01/01/2015	31/12/9999	Y
1005	5	CHEF ANTON'S GUMBO MIX	CONDIMENTS	1	01/01/2015	31/12/9999	Y
1006	6	GRANDMA'S BOYSENBERRY SPREAD	CONDIMENTS	0	01/01/2015	31/12/9999	Y
1007	7	UNCLE BOB'S ORGANIC DRIED PEARS	PRODUCE	0	01/01/2015	31/12/9999	Y
1008	8	NORTHWOODS CRANBERRY SAUCE	CONDIMENTS	0	01/01/2015	31/12/9999	Y
1009	9	MISHI KOBE NIKU	MEAT/POULTRY	1	01/01/2015	31/12/9999	Y
1010	10	IKURA	SEAFOOD	0	01/01/2015	31/12/9999	Y
1011	11	QUESO CABRALES	DAIRY PRODUCTS	0	01/01/2015	31/12/9999	Y

PRODUCT_KEY	PRODUCTID	PRODUCTNAME	CATEGORYNAME	DISCONTINUED	EFFECTIVED	EXPIRATION	C
1078	1	CHAI BERRIES	BEVERAGES	0	05/05/2024	31/12/9999	Y

.....

PRODUCT_KEY	PRODUCTID	PRODUCTNAME	CATEGORYNAME	DISCONTINUED	EFFECTIVED	EXPIRATION	C
1078	1	CHAI BERRIES	BEVERAGES	0	05/05/2024	31/12/9999	Y

ii) For Employee_Dim

For example, I would like to change the name of EmployeeID 9.

EMPLOYEE_KEY	EMPLOYEEID	EMPLOYEENAME	HIREDATE	COUNTRY	REGION	CITY	POSTALCODE	EFFECTIVED	EXPIRATION C
1001	1	Nancy Davolio	01/05/1992	USA	WA	SEATTLE	98122	01/01/2015	31/12/9999 Y
1002	2	Andrew Fuller	14/08/1992	USA	WA	TACOMA	98401	01/01/2015	31/12/9999 Y
1003	3	Janet Leverling	01/04/1992	USA	WA	KIRKLAND	98033	01/01/2015	31/12/9999 Y
1004	4	Margaret Peacock	03/05/1993	USA	WA	REDMOND	98052	01/01/2015	31/12/9999 Y
1005	5	Steven Buchanan	17/10/1993	UK		LONDON	SW1 8JR	01/01/2015	31/12/9999 Y
1006	6	Michael Suyama	17/10/1993	UK		LONDON	EC2 7JR	01/01/2015	31/12/9999 Y
1007	7	Robert King	02/01/1994	UK		LONDON	RG1 9SP	01/01/2015	31/12/9999 Y
1008	8	Laura Callahan	05/03/1994	USA	WA	SEATTLE	98105	01/01/2015	31/12/9999 Y
1009	9	Anne Dodsworth	15/11/1994	UK		LONDON	WG2 7LT	01/01/2015	31/12/9999 Y

9 rows selected.

After executing the below command:

A New Record is inserted and the ExpirationDate, EffectiveDate and CurrentRowIndicator are updated accordingly.

```
exec Insert_Emp_Dim_Record(9, 'Annie Dodsworth', '15/11/1994', 'UK', NULL,
'LONDON', 'WG2 7LT');
```

EMPLOYEE_KEY	EMPLOYEEID	EMPLOYEENAME	HIREDATE	COUNTRY	REGION	CITY	POSTALCODE	EFFECTIVED	EXPIRATION C
1001	1	Nancy Davolio	01/05/1992	USA	WA	SEATTLE	98122	01/01/2015	31/12/9999 Y
1002	2	Andrew Fuller	14/08/1992	USA	WA	TACOMA	98401	01/01/2015	31/12/9999 Y
1003	3	Janet Leverling	01/04/1992	USA	WA	KIRKLAND	98033	01/01/2015	31/12/9999 Y
1004	4	Margaret Peacock	03/05/1993	USA	WA	REDMOND	98052	01/01/2015	31/12/9999 Y
1005	5	Steven Buchanan	17/10/1993	UK		LONDON	SW1 8JR	01/01/2015	31/12/9999 Y
1006	6	Michael Suyama	17/10/1993	UK		LONDON	EC2 7JR	01/01/2015	31/12/9999 Y
1007	7	Robert King	02/01/1994	UK		LONDON	RG1 9SP	01/01/2015	31/12/9999 Y
1008	8	Laura Callahan	05/03/1994	USA	WA	SEATTLE	98105	01/01/2015	31/12/9999 Y
1009	9	Anne Dodsworth	15/11/1994	UK		LONDON	WG2 7LT	01/01/2015	04/05/2024 N
1010	9	Annie Dodsworth	15/11/1994	UK		LONDON	WG2 7LT	05/05/2024	31/12/9999 Y

iii) For Customer_Dim

For example, I would like to change the name of the customer with cust_no 105824

CUSTOMER_KEY	CUST_NO	COMP_NAME	COUNTRY	REGION	CITY	POSTCODE	EFFECTIVED	EXPIRATION	C
6820	105820	Wartian Herkku	Finland		Oulu	90110	26/03/2016	31/12/9999	Y
6821	105821	Wellington Importado	Brazil	SP	Resende	08737-363	21/02/2016	31/12/9999	Y
6822	105822	White Clover Markets	USA	WA	Seattle	98128	03/01/2016	31/12/9999	Y
6823	105823	Wilman Kala	Finland		Helsinki	21240	07/02/2015	31/12/9999	Y
6824	105824	Wolski Zajazd	Poland		Warszawa	01-012	26/09/2015	31/12/9999	Y

After executing the below command:

A New Record is inserted and the ExpirationDate, EffectiveDate and CurrentRowIndicator are updated accordingly.

```
exec Insert_Cust_Dim_Record(105824, 'Aston Martin', 'Poland', NULL, 'Warszawa', '01-012');
```

CUSTOMER_KEY	CUST_NO	COMP_NAME	COUNTRY	REGION	CITY	POSTCODE	EFFECTIVED	EXPIRATION	C
6820	105820	Wartian Herkku	Finland		Oulu	90110	26/03/2016	31/12/9999	Y
6821	105821	Wellington Importado	Brazil	SP	Resende	08737-363	21/02/2016	31/12/9999	Y
6822	105822	White Clover Markets	USA	WA	Seattle	98128	03/01/2016	31/12/9999	Y
6823	105823	Wilman Kala	Finland		Helsinki	21240	07/02/2015	31/12/9999	Y
6824	105824	Wolski Zajazd	Poland		Warszawa	01-012	26/09/2015	04/05/2024	N
6825	105824	Aston Martin	Poland		Warszawa	01-012	05/05/2024	31/12/9999	Y

Chapter 3 Business Analytics Reports

3.1 Ong Weng Kai

3.1.1 Generating quarterly sales for a given year and comparing it with the previous year's total sales

i) Purpose:

This query and the resulting data are essential for several reasons:

1. **Performance Evaluation:** The data provides insight into individual employee performance over quarters and across years. It helps assess how each employee's sales figures have evolved, which is critical for performance reviews and appraisals.
2. **Strategic Planning:** The quarterly and yearly sales data, when carefully analyzed, can reveal trends, such as which quarters are more profitable or show higher sales activity. This invaluable information can steer strategic decisions regarding sales targets, resource allocation, and planning for future periods.
3. **Incentive and Compensation Planning:** Companies often use such data to structure compensation plans, including bonuses and incentives. Employees who show substantial growth or consistently high performance might be rewarded, which can motivate further productivity and loyalty to the company.
4. **Identifying Training Needs:** The data can help identify underperforming employees, suggesting a need for additional training or support to improve their sales skills.
5. **Forecasting and Budgeting:** Historical sales data is crucial for forecasting future sales and setting realistic budgets. It helps in understanding market dynamics and the effectiveness of sales strategies.
6. **Market and Economic Impact Analysis:** Fluctuations in sales across the quarters can also reflect market conditions or economic impacts, providing insights beyond individual performance that are useful for broader business strategy adjustments.

ii) SQL Code:

```

set linesize 3000
set pagesize 30
DEFINE page_count = 1

ACCEPT year_input CHAR PROMPT 'Please enter the year for analysis (e.g. 2015-2024) : '

BREAK ON REPORT
COMPUTE SUM LABEL 'Total' OF "Total_Sales_&year_input.($)" "Previous Year Sales ($)" ON REPORT
COMPUTE AVG LABEL 'Average' OF "Q1_&year_input.($)" "Q2_&year_input.($)" "Q3_&year_input.($)" "Q4_&year_input.($)" "Growth(%)" ON REPORT

REPFOOTER '
END OF REPORT ) '

```

TTITLE

```
'=====
====='
SKIP 1-
'                               &year_input   Employee
Sales Performance and Year-over-Year Growth Analysis           ' SQL.PNO
" / &page_count" SKIP 1-
'=====
====='
-
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'=====
====='
-
```

BTITLE

```
'=====
====='
SKIP 1-
'                               End of
Page' FORMAT 9 SQL.PNO SKIP 1-
'=====
====='
SKIP 2-
```

```
COLUMN "ID" FORMAT A10
COLUMN "Employee Name" FORMAT A17
COLUMN "Q1_&year_input.($)" FORMAT $999,999,999.00
COLUMN "Q2_&year_input.($)" FORMAT $999,999,999.00
COLUMN "Q3_&year_input.($)" FORMAT $999,999,999.00
COLUMN "Q4_&year_input.($)" FORMAT $999,999,999.00
COLUMN "Total_Sales_&year_input.($)" FORMAT $999,999,999.00
COLUMN "Previous Year Sales($)" FORMAT $999,999,999.00
COLUMN "Growth(%)" FORMAT 999.99
```

SELECT

```
    TO_CHAR(e.employeeID) AS "ID",
    e.employeeName AS "Employee Name",
    SUM(CASE WHEN d.cal_year = &year_input AND d.cal_quarter = 'Q1' THEN
f.line_total END) AS "Q1_&year_input.($)",
    SUM(CASE WHEN d.cal_year = &year_input AND d.cal_quarter = 'Q2' THEN
f.line_total END) AS "Q2_&year_input.($)",
    SUM(CASE WHEN d.cal_year = &year_input AND d.cal_quarter = 'Q3' THEN
f.line_total END) AS "Q3_&year_input.($)",
    SUM(CASE WHEN d.cal_year = &year_input AND d.cal_quarter = 'Q4' THEN
f.line_total END) AS "Q4_&year_input.($)",
    SUM(CASE WHEN d.cal_year = &year_input THEN f.line_total END) AS
"Total_Sales_&year_input.($)",
```

```

        SUM(CASE WHEN d.cal_year = &year_input-1 THEN f.line_total END) AS
"Previous Year Sales($)" ,
        ROUND((SUM(CASE WHEN d.cal_year = &year_input THEN f.line_total END) -
                SUM(CASE WHEN d.cal_year = &year_input-1 THEN f.line_total END)) /
                NULLIF(SUM(CASE WHEN d.cal_year = &year_input-1 THEN f.line_total
END), 0) * 100, 2) AS "Growth(%)"
FROM
    Order_Fact f
JOIN
    Employee_dim e ON f.employee_key = e.employee_key
JOIN
    Date_dim d ON f.date_key = d.date_key
GROUP BY
    e.employeeID, e.employeeName
ORDER BY
    "Total_Sales_&year_input.($)" DESC;

TTITLE OFF
BTITLE OFF

```

iii) Output:

```

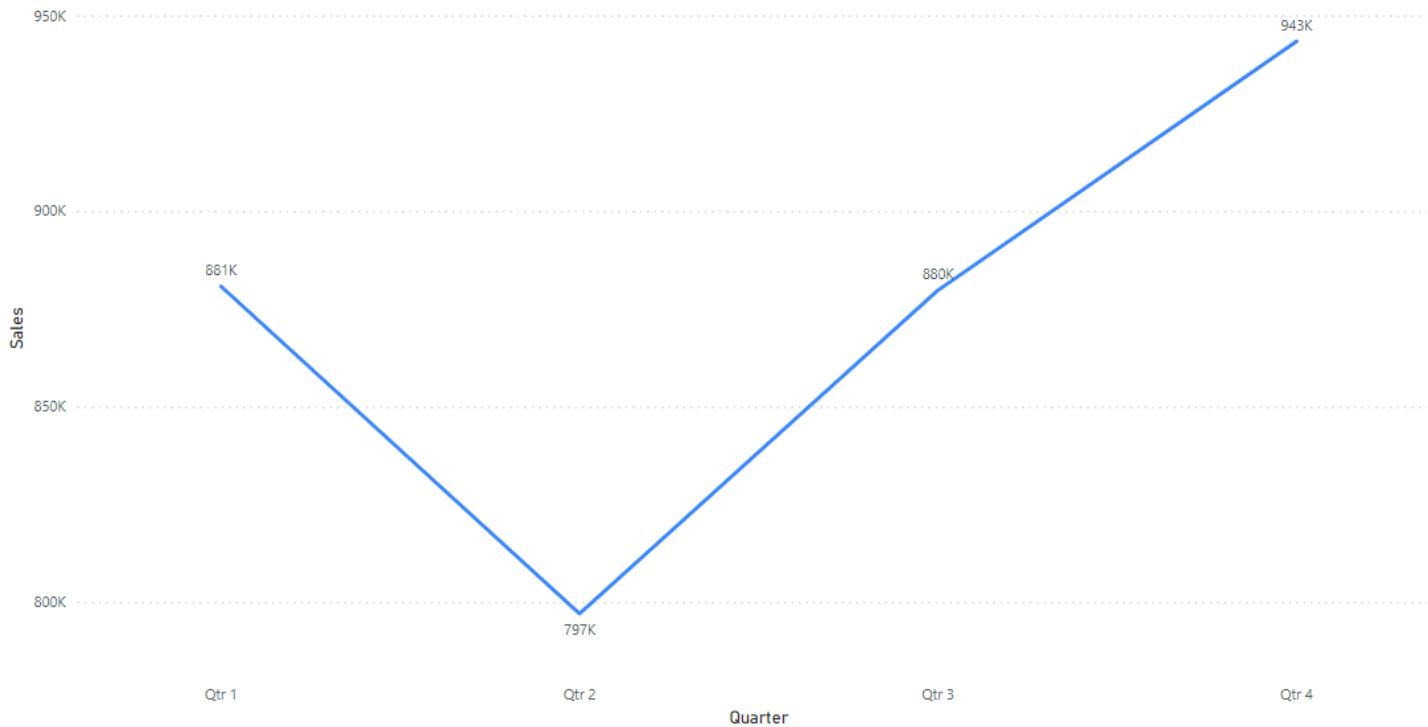
-----+
      start C:\DataWarehouseAssignment\Q1Trial(3).txt;
Please enter the year for analysis (e.g. 2015-2024): 2023
-----+

```

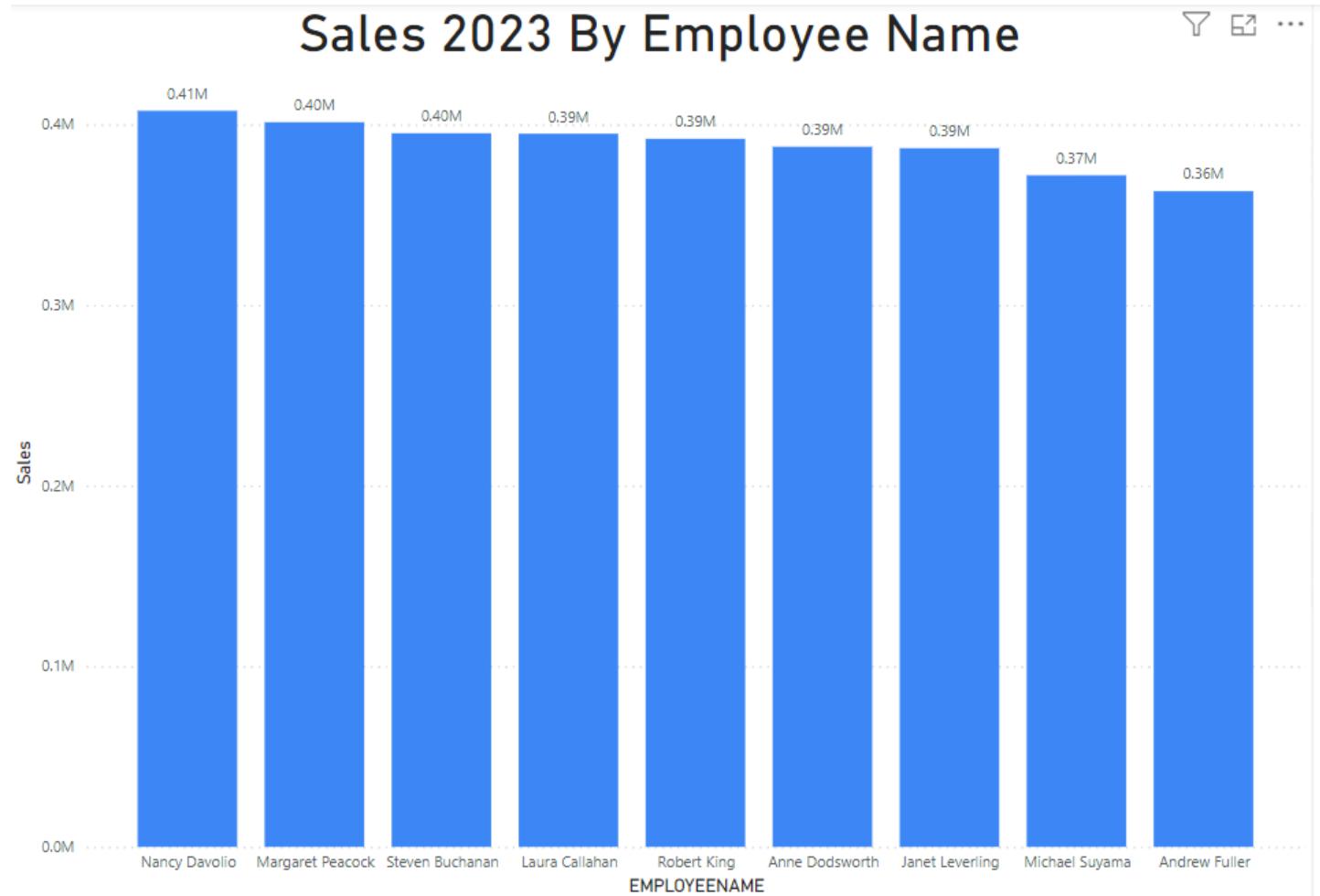
2023 Employee Sales Performance and Year-over-Year Growth Analysis								1 / 1
ID	Employee Name	Q1_2023(\$)	Q2_2023(\$)	Q3_2023(\$)	Q4_2023(\$)	Total_Sales_2023(\$)	Previous Year Sales(\$)	Growth(%)
4	Margaret Peacock	\$115,081.50	\$110,006.24	\$87,341.70	\$97,185.63	\$409,615.07	\$424,291.54	-3.46
2	Andrew Fuller	\$110,558.38	\$101,783.09	\$94,861.03	\$92,150.85	\$399,353.35	\$409,799.71	-2.55
3	Janet Leverling	\$91,377.02	\$105,634.80	\$104,220.77	\$88,069.25	\$389,301.84	\$382,567.37	1.76
9	Anne Dodsworth	\$96,793.25	\$107,848.12	\$102,930.71	\$78,621.66	\$386,193.74	\$385,628.09	.15
1	Nancy Davolio	\$100,169.11	\$98,430.98	\$90,705.79	\$94,323.92	\$383,629.80	\$395,395.62	-2.98
5	Steven Buchanan	\$92,739.79	\$91,768.32	\$79,061.02	\$106,594.06	\$370,163.19	\$381,310.17	-2.92
7	Robert King	\$97,846.06	\$96,667.20	\$91,062.69	\$80,339.71	\$365,915.66	\$390,291.20	-6.25
8	Laura Callahan	\$80,774.59	\$99,288.04	\$95,432.66	\$89,568.15	\$365,063.44	\$396,970.22	-8.04
6	Michael Suyama	\$89,719.67	\$95,676.88	\$87,368.35	\$90,688.70	\$363,453.60	\$358,098.91	1.50
Average Total		\$97,228.82	\$100,789.30	\$92,553.86	\$90,837.99	\$3,432,689.69	\$3,524,352.83	-2.53
(END OF REPORT)								

iv) Visualization For Query 1:

Sales by 2023 Quarter

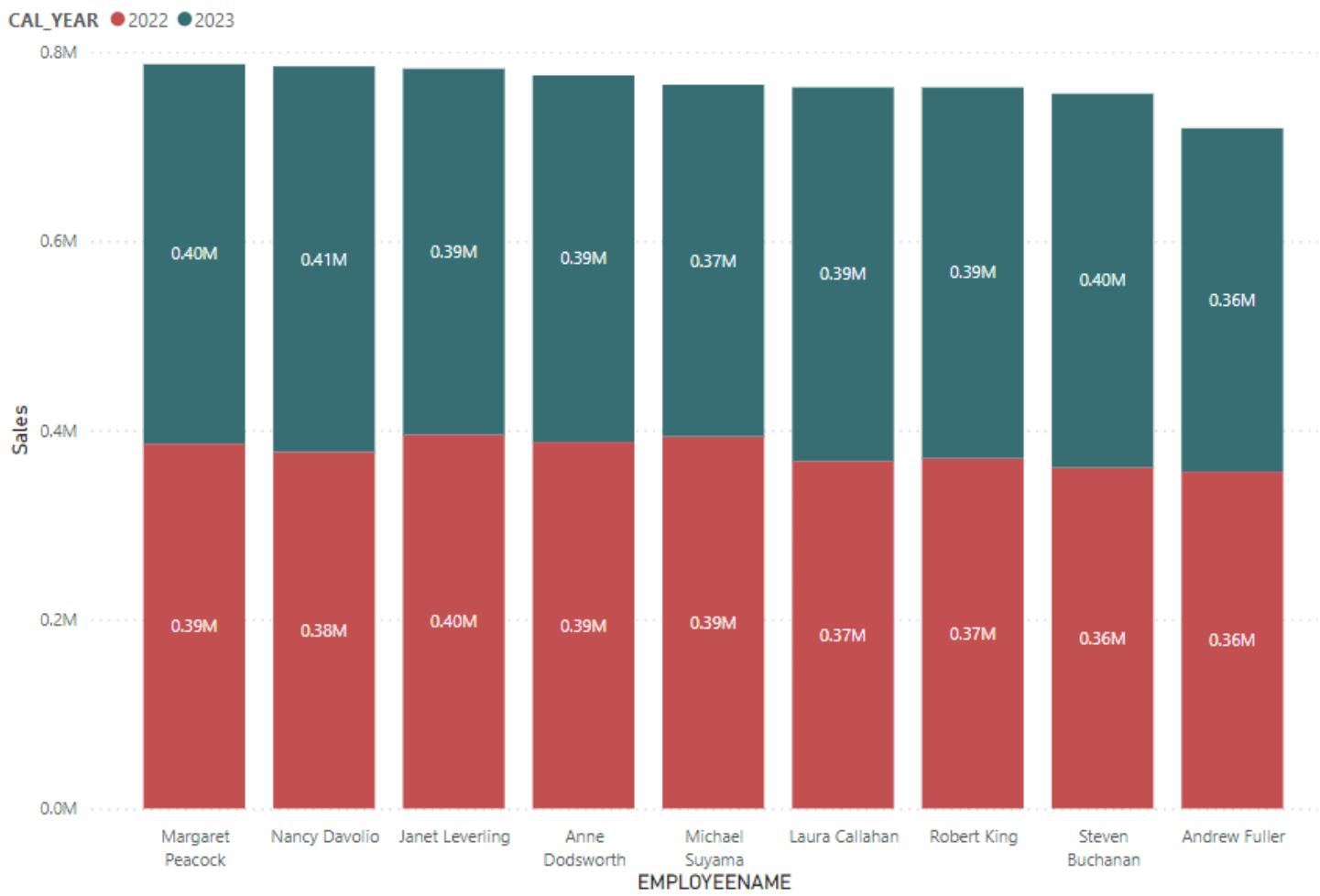


Observation: Based on the graph, we can observe that sales dropped significantly in Quarter 2 of 2023 but rebounded to the level of Quarter 1 in Quarter 3 and continued to grow in the following quarter. This information can help the business investigate the factors that caused the dip in Quarter 2, such as market conditions, product availability, or marketing efforts, and implement strategies to prevent similar drops in the future. The rebound and subsequent growth indicate successful recovery strategies or seasonal trends the business can capitalize on. By analyzing what contributed to Quarters 3 and 4 growth, the business can replicate these strategies in other quarters to maintain steady sales growth throughout the year.



Observation: Nancy Davolio had the highest sales in 2023, reaching 0.41 million. Conversely, Michael Suyama and Andrew Fuller were the year's worst performers, with sales of only 0.37 million and 0.36 million, respectively. This information can help the business identify and analyze the strategies, customer interactions, and sales techniques that led to Nancy Davolio's success, which can be shared and implemented across the team to boost overall performance. Additionally, understanding the challenges faced by Michael Suyama and Andrew Fuller can inform targeted training and support initiatives to address their specific needs, ultimately enhancing their sales outcomes and contributing to a more balanced and effective sales team.

Sales by EMPLOYEE NAME 2022 vs 2023



Observation: Based on the graph above, we can see that Steven Buchanan's sales grew the most, increasing from 0.36 million (M) to 0.40 million, which is 0.04 million more than in the previous year, indicating a significant positive trend in his sales performance. Conversely, Michael Suyama experienced the most significant decline in sales in 2023. This information can be used by the business to identify and analyze the strategies and practices that contributed to Steven Buchanan's success, which can be replicated or adapted by other sales team members. Additionally, understanding the reasons behind Michael Suyama's sales drop can help in providing targeted support, training, or resources to improve his performance, thereby enhancing the overall effectiveness and productivity of the sales team.

3.1.2 Customizable Annual Sales and Profitability Report

i) Purpose:

The query is significant because it provides a detailed breakdown of product sales, which is fundamental for several critical business functions:

1. **Pricing and Discount Analysis:** By examining the unit price, discount rate, and actual sale price of products, businesses can assess the effectiveness of their pricing strategies and discount policies. This helps determine if the current pricing is optimal for maximizing revenue and if discounts drive sufficient sales volume without eroding profit margins.
2. **Inventory and Supply Chain Management:** Quantity and sales data are not just helpful but essential. They help businesses understand which products are in high demand, allowing for efficient inventory optimization and effective supply chain management. This ensures that popular products are sufficiently stocked and helps reduce overstock of less popular items.
3. **Revenue Forecasting:** Understanding past sales performance at the product level enables more accurate predictions of future sales, which is essential for financial planning and budgeting. It helps forecast revenue streams and plan for potential expansions or reductions in product lines.
4. **Market Strategy Adjustment:** Detailed sales data per product provide a strategic advantage, allowing companies to identify market trends and consumer preferences. Products with solid sales and growth percentages might warrant additional marketing efforts or be positioned as flagship offerings, while underperforming products might need strategic reassessment or discontinuation.
5. **Performance Metrics for Sales Teams:** Providing insights into which products are selling well and which are not can help sales teams focus on promoting high-margin or strategically essential products.

ii) SQL Code:

```
set linesize 5000
set pagesize 95
DEFINE page_count = 1

ACCEPT year_input CHAR PROMPT 'Please enter the year for analysis (e.g., 2015-2024): '
ACCEPT product_input NUMBER PROMPT 'How many records of products are to be displayed? (Enter a number): '

BREAK ON REPORT
COMPUTE SUM LABEL 'Total' OF "Sales($)" "Quantity" ON REPORT
COMPUTE AVG LABEL 'Average' OF "Sale_Price($)" "Discount_Rate(%)" "Unit_Price($)" ON REPORT

REPFOOTER '
END OF REPORT ) '
```

TTITLE

```
'=====
===== SKIP 1-
'
&year_input Annual Sales
and Profitability Report      ' SQL.PNO " / &page_count" SKIP 1-
'=====
===== -
```

SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -

```
'=====
===== -
```

BTITLE

```
'=====
===== SKIP 1-
'
End of
Page' FORMAT 9 SQL.PNO SKIP 1-
'=====
===== SKIP 2-
```

```
COLUMN "Unit_Price($)" FORMAT $999,999.00
COLUMN "Discount_Rate(%)" FORMAT 99.00
COLUMN "Sale_Price($)" FORMAT $999,999.00
COLUMN "Quantity" FORMAT 999,999
COLUMN "Sales($)" FORMAT $999,999,999.00
COLUMN "SalesPct(%)" FORMAT 999.00
```

SELECT

```
ProductID,
ProductName,
"Unit_Price($)",
"Discount_Rate(%)",
"Sale_Price($)",
"Quantity",
"Sales($)",
"SalesPct(%)"
FROM (
  SELECT
    p.ProductID,
    p.ProductName,
    AVG(f.UnitPrice) AS "Unit_Price($)",
    AVG(f.Discount) * 100 AS "Discount_Rate(%)",
    AVG(f.Line_Total / f.Quantity) AS "Sale_Price($)",
    SUM(f.Quantity) AS "Quantity",
    SUM(f.Line_Total) AS "Sales($)",
    ROUND((SUM(f.Line_Total) /
      SELECT SUM(of2.Line_Total)
      FROM Order_Fact of2
      JOIN Date_dim od2 ON of2.date_key = od2.date_key)
```

```
        WHERE od2.cal_year = &year_input
      ) * 100), 2) AS "SalesPct(%)",
      ROW_NUMBER() OVER (ORDER BY SUM(f.Line_Total) DESC) AS rn
  FROM
    Order_Fact f
  JOIN
    Product_Dim p ON f.product_key = p.Product_key
  JOIN
    Date_dim d ON f.date_key = d.date_key
  WHERE
    d.cal_year = &year_input
  GROUP BY
    p.ProductID, p.ProductName
)
WHERE rn <= &product_input;

TTITLE OFF
BTITLE OFF
```

iii) Output:

SQL> start C:\DataWarehouseAssignment\OWKQ2.txt;
 Please enter the year for analysis (e.g., 2015-2024): 2023
 How many records of products are to be displayed? (Enter a number): 5000

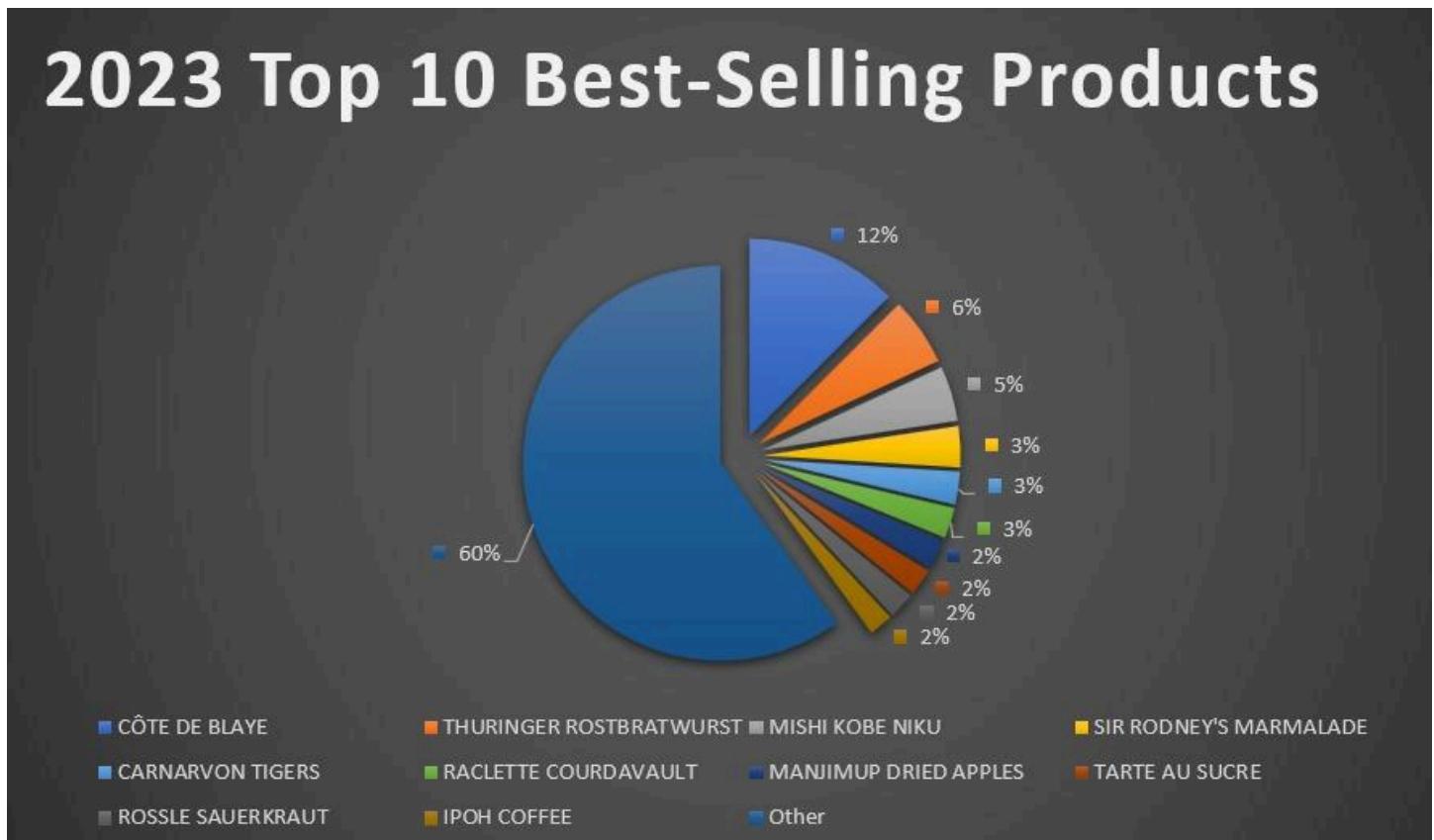
2023 Annual Sales and Profitability Report							1 / 1
PRODUCTID	PRODUCTNAME	Unit_Price(\$)	Discount_Rate(%)	Sale_Price(\$)	Quantity	Sales(\$)	SalesPct(%)
38	CÔTE DE BLAYE	\$263.50	12.48	\$230.61	1,851	\$428,749.19	12.49
29	THURINGER ROSTBRATWURST	\$123.79	12.49	\$108.33	1,782	\$192,570.12	5.61
9	MISHI KOBE NIJKU	\$97.00	12.91	\$84.48	1,841	\$155,209.70	4.52
20	SIR RODNEY'S MARMALADE	\$81.00	12.45	\$70.92	1,628	\$115,135.02	3.35
18	CARNARVON TIGERS	\$62.50	12.67	\$54.58	1,698	\$92,491.60	2.69
59	RACLETTE COURDAV AULT	\$55.00	12.24	\$48.27	1,778	\$85,732.35	2.50
51	MANJIMUP DRIED APPLES	\$53.00	12.38	\$46.44	1,824	\$84,193.68	2.45
62	TARTE AU SUCRE	\$49.30	12.14	\$43.31	1,751	\$75,798.00	2.21
28	ROSSLE SAUKRAUT	\$45.60	12.40	\$39.95	1,878	\$74,604.41	2.17
43	IPOH COFFEE	\$46.00	12.61	\$40.20	1,668	\$67,340.32	1.96
17	ALICE MUTTON	\$39.00	12.39	\$34.17	1,912	\$65,226.72	1.90
63	VEGIE-SPREAD	\$43.90	12.13	\$38.57	1,637	\$63,261.41	1.84
56	GNOCCHI DI NONNA ALICE	\$38.00	12.25	\$33.35	1,791	\$59,786.54	1.74
27	SCHOGGI SCHOKOLADE	\$43.90	13.74	\$37.87	1,571	\$59,324.86	1.73
8	NORTHWOODS CRANBERRY SAUCE	\$40.00	12.76	\$34.90	1,654	\$57,631.20	1.68
12	QUESO MANCHEGO LA PASTORA	\$38.00	12.50	\$33.25	1,711	\$56,873.84	1.66
69	GUDBRANDSDALSO ST	\$36.00	12.07	\$31.66	1,735	\$55,241.64	1.61
60	CAMEMBERT PIERROT	\$34.00	12.13	\$29.88	1,844	\$55,149.36	1.61
72	MOZZARELLA DI GIOVANNI	\$34.80	12.58	\$30.42	1,785	\$54,442.14	1.59
53	PERTH PASTIES	\$32.80	12.77	\$28.61	1,828	\$52,391.12	1.53
32	MASCARPONE FABIOLI	\$32.00	12.73	\$27.93	1,868	\$52,240.32	1.52
64	WIMMERS GUTE SEMMELKNODEL	\$33.25	12.89	\$28.96	1,748	\$50,414.77	1.47
26	GUMBAR GUMMIBARCHEN	\$31.23	12.25	\$27.40	1,777	\$48,785.40	1.42
10	IKURA	\$31.00	12.60	\$27.09	1,736	\$46,919.43	1.37
7	UNCLE BOB'S ORGANIC DRIED PEARS	\$30.00	12.24	\$26.33	1,681	\$44,331.30	1.29
61	SIROP D'ERABLE	\$28.50	12.08	\$25.06	1,657	\$41,748.31	1.22
6	GRANDMA'S BOYSENBERRY SPREAD	\$25.00	12.85	\$21.79	1,856	\$40,372.75	1.18
37	GRAVAD LAX	\$26.00	12.10	\$22.85	1,717	\$39,303.42	1.14
30	NORD-OST MATJESHERING	\$25.89	12.60	\$22.63	1,730	\$39,202.45	1.14
5	CHEF ANTON'S GUMBO MIX	\$21.35	11.93	\$18.80	1,906	\$35,787.54	1.04
71	FLOTEMYSOST	\$21.50	13.13	\$18.68	1,868	\$34,808.48	1.01
14	TOFU	\$23.25	13.49	\$20.11	1,681	\$33,618.74	.98
55	PATE CHINOIS	\$24.00	12.56	\$20.98	1,585	\$33,339.60	.97
65	LOUISIANA FIERY HOT PEPPER SAUCE	\$21.05	12.12	\$18.50	1,785	\$33,030.22	.96
4	CHEF ANTON'S CAJUN SEASONING	\$22.00	12.78	\$19.19	1,697	\$32,537.56	.95
49	MAXILAKU	\$20.00	12.35	\$17.53	1,849	\$32,340.00	.94
11	QUESO CABRALES	\$21.00	12.18	\$18.44	1,720	\$31,802.40	.93
1	CHAI	\$18.00	11.60	\$15.91	1,928	\$30,786.30	.90
36	INLAGD SILL	\$19.00	12.38	\$16.65	1,836	\$30,638.64	.89
22	GUSTAF'S KNACKEBROD	\$21.00	13.36	\$18.19	1,682	\$30,603.72	.89
57	RAVIOLI ANGELO	\$19.50	12.14	\$17.13	1,737	\$29,667.88	.86
40	BOSTON CRAB MEAT	\$18.40	12.99	\$16.01	1,851	\$29,535.48	.86
2	CHANG	\$19.00	12.56	\$16.61	1,779	\$29,484.20	.86
44	GULA MALACCA	\$19.45	12.24	\$17.07	1,699	\$28,972.30	.84
35	STEELEYE STOUT	\$18.00	12.38	\$15.77	1,755	\$27,724.68	.81
66	LOUISIANA HOT SPICED OKRA	\$17.00	12.11	\$14.94	1,794	\$26,833.99	.78
16	PAVLOVA	\$17.45	12.27	\$15.31	1,703	\$26,143.99	.76
15	GENEN SHOUYU	\$15.50	12.46	\$13.57	1,890	\$25,665.16	.75
39	CHARTREUSE VERTE	\$18.00	12.46	\$15.76	1,565	\$24,781.32	.72
50	VALKOINEN SUKLAA	\$16.25	12.77	\$14.17	1,711	\$24,205.36	.71
73	ROD KAVIAR	\$15.00	12.33	\$13.15	1,828	\$24,061.20	.70
76	LAKRALLIKOORTI	\$18.00	13.38	\$15.59	1,540	\$24,009.48	.70
70	OUTBACK LAGER	\$15.00	12.71	\$13.09	1,743	\$22,776.00	.66
34	SASQUATCH ALE	\$14.00	13.39	\$12.12	1,852	\$22,407.98	.65
42	SINGAPOREAN HOKKIEN FRIED MEE	\$14.00	12.93	\$12.19	1,779	\$21,726.18	.63
68	SCOTTISH LONGBREADS	\$12.50	12.43	\$10.95	1,958	\$21,437.80	.62
67	LAUGHING LUMBERJACK LAGER	\$14.00	12.72	\$12.22	1,743	\$21,325.08	.62
25	NUNUCA NUB-NOUGAT-CRÈME	\$14.00	12.15	\$12.30	1,700	\$20,900.32	.61
46	SPEGESILD	\$12.00	12.59	\$10.49	1,865	\$19,598.64	.57
48	CHOCOLADE	\$12.75	12.15	\$11.20	1,701	\$19,059.67	.56
58	ESCARGOTS DE BOURGOGNE	\$13.25	12.11	\$11.65	1,626	\$18,961.44	.55
77	ORIGINAL FRANKFURTER GRUNE SOBE	\$13.00	13.52	\$11.24	1,611	\$18,160.35	.53
31	GORGONZOLA TELINO	\$12.50	12.98	\$10.88	1,670	\$18,140.23	.53
19	TEATIME CHOCOLATE BISCUITS	\$9.20	12.38	\$8.06	1,963	\$15,827.50	.46
21	SIR RODNEY'S SCONES	\$10.00	12.80	\$8.72	1,795	\$15,652.70	.46
47	ZAANSE KOEKEN	\$9.50	12.66	\$8.30	1,879	\$15,563.00	.45
23	TUNNBROD	\$9.00	12.16	\$7.91	1,947	\$15,392.88	.45
41	JACK'S NEW ENGLAND CLAM CHOWDER	\$9.65	12.57	\$8.44	1,743	\$14,726.07	.43
3	ANISEED SYRUP	\$10.00	12.01	\$8.80	1,617	\$14,205.10	.41
74	LONGLIFE TOFU	\$10.00	12.03	\$8.80	1,600	\$14,055.10	.41
45	ROGEDE SILD	\$9.50	12.13	\$8.35	1,636	\$13,672.29	.40
75	RHÖNBRÄU KLOSTERBIER	\$7.75	13.30	\$6.72	1,849	\$12,431.78	.36

54 TOURTIERE	\$7.45	12.78	\$6.50	1,641	\$10,685.19	.31
52 FILO MIX	\$7.00	12.53	\$6.12	1,670	\$10,185.84	.30
13 KONBU	\$6.00	12.72	\$5.24	1,895	\$9,937.26	.29
24 GUARANA FANTASTICA	\$4.50	12.51	\$3.94	1,849	\$7,277.89	.21
33 GEITOST	\$2.50	13.15	\$2.17	1,720	\$3,733.79	.11
Average	\$28.87	12.54	\$25.25			
Total				135,378	\$3,432,689.69	

(END OF REPORT)

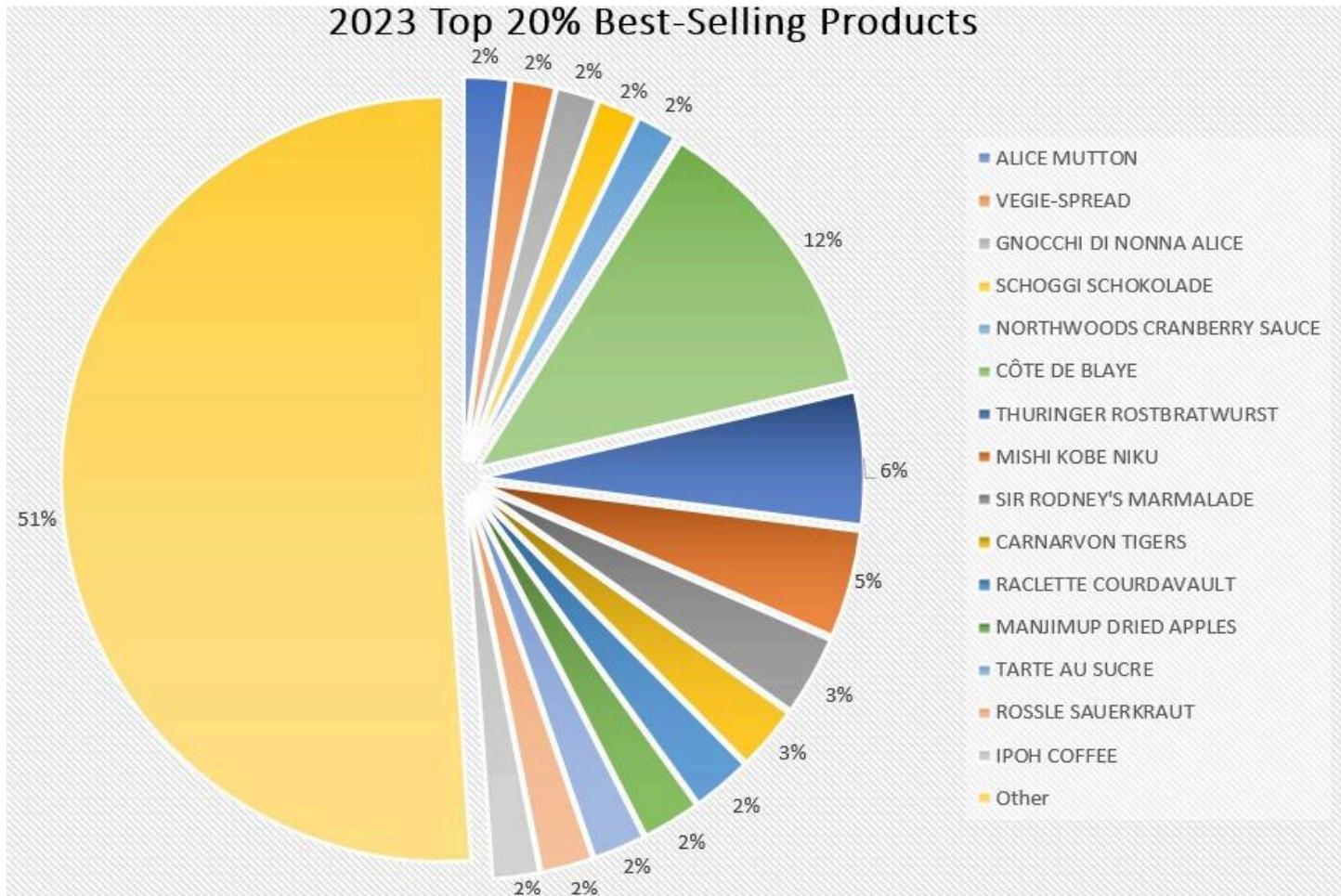
===== End of Page 1 =====

iv) Visualisation for Query 2



Observation:

"CÔTE DE BLAYE" is the single most significant contributor among the top 10, with 12% of total sales, followed by "THURINGER ROSTBRATWURST" at 6%. The Top 10 products contributed 40 % of the total sales while the remaining contributed by other products. This indicates that few of the products contribute the majority of the sales; so first, this can be insightful for Inventory Management where products like "CÔTE DE BLAYE" and "THURINGER ROSTBRATWURST" should be stocked in higher quantities to meet demand and avoid stockouts. Secondly, it can be utilised for sales strategies where the marketing team can focus marketing efforts on high-performing products to maximise return on investment. Promotions for "CÔTE DE BLAYE" might be particularly effective given its strong sales performance. After that, insights from the sales data can guide the development of new products. If certain types of products are consistently top sellers, developing similar products could be beneficial, such as "CÔTE DE BLAYE" and "THURINGER ROSTBRATWURST".

**Observation:**

The top 20% of best-selling products (15) contributed nearly half of total sales, indicating that a small number of products drive the majority of sales. This suggests that the marketing team should focus on developing or acquiring more products similar to these best-sellers, as they align with customer preferences and buying behaviour. Additionally, the business can allocate more marketing resources to promote these top-performing products, optimize inventory levels to ensure their availability, and analyze the characteristics and features that make these products successful to inform future product development. Understanding the factors contributing to the success of the top 20% can help improve the performance of the remaining 80%, ultimately enhancing overall sales and profitability.

3.1.3 Dynamic Report: Monthly sales performance and growth analysis

i) Purpose:

The query is essential because it offers a comprehensive monthly sales analysis over a year, including comparisons to previous months and growth percentages. Here is why this data is crucial:

1. **Trend Analysis:** By tracking sales month by month and comparing them to previous months, businesses can identify trends, seasonal variations, and patterns in consumer behaviour. This analysis helps understand which months typically see higher sales, which can be critical for planning marketing campaigns, promotions, and other sales strategies.
2. **Performance Evaluation:** The growth percentage column highlights how sales evolve monthly. This is key for evaluating the effectiveness of marketing strategies, sales promotions, and any market conditions or operational tactics changes.
3. **Budgeting and Forecasting:** With detailed monthly sales data, companies can more accurately forecast future sales and budget their expenses accordingly. Knowing the average and total sales for the year also aids in setting realistic goals for the next year and adjusting the monthly sales targets based on historical data.
4. **Resource Allocation:** Understanding when sales peaks and troughs occur allows a business to allocate resources, such as staffing and inventory, better to meet demand without overcommitting resources during slower periods.
5. **Strategic Planning:** Sales data provides foundational insights that inform broader business strategies. If certain months show declining sales, a company should consider strategic initiatives to boost customer engagement during those periods.

ii) SQL Code:

```

SET PAGESIZE 28
SET LINESIZE 1000
DEFINE page_count = 1
COLUMN "Sales" FORMAT A17
COLUMN "Previous_Month" FORMAT A17
COLUMN "Growth (%)" FORMAT 999.00

ACCEPT p_year PROMPT 'Please enter the year for the analysis(e.g. 2015-2024)::'
'

REPFOOTER' ( END OF REPORT ) '
TTITLE
'===== SKIP
1-

```

```

' &p_year Monthly Sales Performance and Growth Analysis      ' SQL.PNO " /
&page_count" SKIP 1-
'=====
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'=====

BTITLE
'===== SKIP
1-
'
          End of Page' FORMAT 9 SQL.PNO SKIP 1-
'===== SKIP
2-

```

WITH MonthlySales AS (
 SELECT
 d.cal_month_number,
 SUM(f.line_total) AS MonthlySales
 FROM
 order_Fact f
 JOIN
 date_dim d ON f.Date_key = d.Date_key
 WHERE
 d.cal_year_month LIKE '&p_year-%'
 GROUP BY
 d.cal_month_number
),
PreviousMonth AS (
 SELECT
 cal_month_number,
 LAG(MonthlySales) OVER (ORDER BY cal_month_number) AS PreviousMonthSales
 FROM
 MonthlySales
),
SalesAnalysis AS (
 SELECT
 a.cal_month_number,
 a.MonthlySales,
 b.PreviousMonthSales,
 CASE
 WHEN b.PreviousMonthSales IS NULL THEN NULL
 ELSE ROUND((a.MonthlySales - b.PreviousMonthSales) /
COALESCE(b.PreviousMonthSales, 1) * 100, 2)
 END AS GrowthRate,
 CASE
 WHEN b.PreviousMonthSales IS NULL THEN 'First Month'
 WHEN a.MonthlySales > b.PreviousMonthSales THEN 'Increase'
 WHEN a.MonthlySales < b.PreviousMonthSales THEN 'Decrease'
 END AS Status
 FROM
 MonthlySales a
 JOIN
 PreviousMonth b
 ON
 a.cal_month_number = b.cal_month_number
)

```
        ELSE 'Stable'
    END AS Status
FROM
    MonthlySales a
LEFT JOIN
    PreviousMonth b ON a.cal_month_number = b.cal_month_number
),
TotalAndAverage AS (
    SELECT
        'Total Sales Amount' AS Metric,
        SUM(MonthlySales) AS Amount,
        14 AS SortOrder -- Adjusted to 14 for clarity as it comes after the
months
    FROM
        MonthlySales
    UNION ALL
    SELECT
        'Average Sales Amount' AS Metric,
        SUM(MonthlySales) / 12 AS Amount,
        13 AS SortOrder -- Adjusted to 13 for clarity as it comes before total
    FROM
        MonthlySales
),
FinalResults AS (
    SELECT
        CASE cal_month_number
            WHEN 1 THEN 'January'
            WHEN 2 THEN 'February'
            WHEN 3 THEN 'March'
            WHEN 4 THEN 'April'
            WHEN 5 THEN 'May'
            WHEN 6 THEN 'June'
            WHEN 7 THEN 'July'
            WHEN 8 THEN 'August'
            WHEN 9 THEN 'September'
            WHEN 10 THEN 'October'
            WHEN 11 THEN 'November'
            WHEN 12 THEN 'December'
        END AS Month,
        TO_CHAR(a.MonthlySales, 'FM$999,999,990.00') AS "Sales",
        TO_CHAR(a.PreviousMonthSales, 'FM$999,999,990.00') AS "Previous_Month",
        TO_CHAR(a.GrowthRate, '999.00') AS "Growth (%)",
        a.Status,
        a.cal_month_number
    FROM
        SalesAnalysis a
    UNION ALL
    SELECT
        Metric AS Month,
```

```

        TO_CHAR(Amount, 'FM$999,999,990.00') AS "Sales",
        NULL AS "Previous_Month",
        NULL AS "Growth (%)",
        NULL AS Status,
        SortOrder AS cal_month_number
    FROM
        TotalAndAverage
)
SELECT Month, "Sales", "Previous_Month", "Growth (%)" FROM FinalResults
ORDER BY cal_month_number;

```

iii) Output:

SQL> start C:\DataWarehouseAssignment\owkq3.txt;
Please enter the year for the analysis(e.g. 2015-2024):: 2023

=====

2023 Monthly Sales Performance and Growth Analysis 1 / 1

=====

Report Date: 16-MAY-24

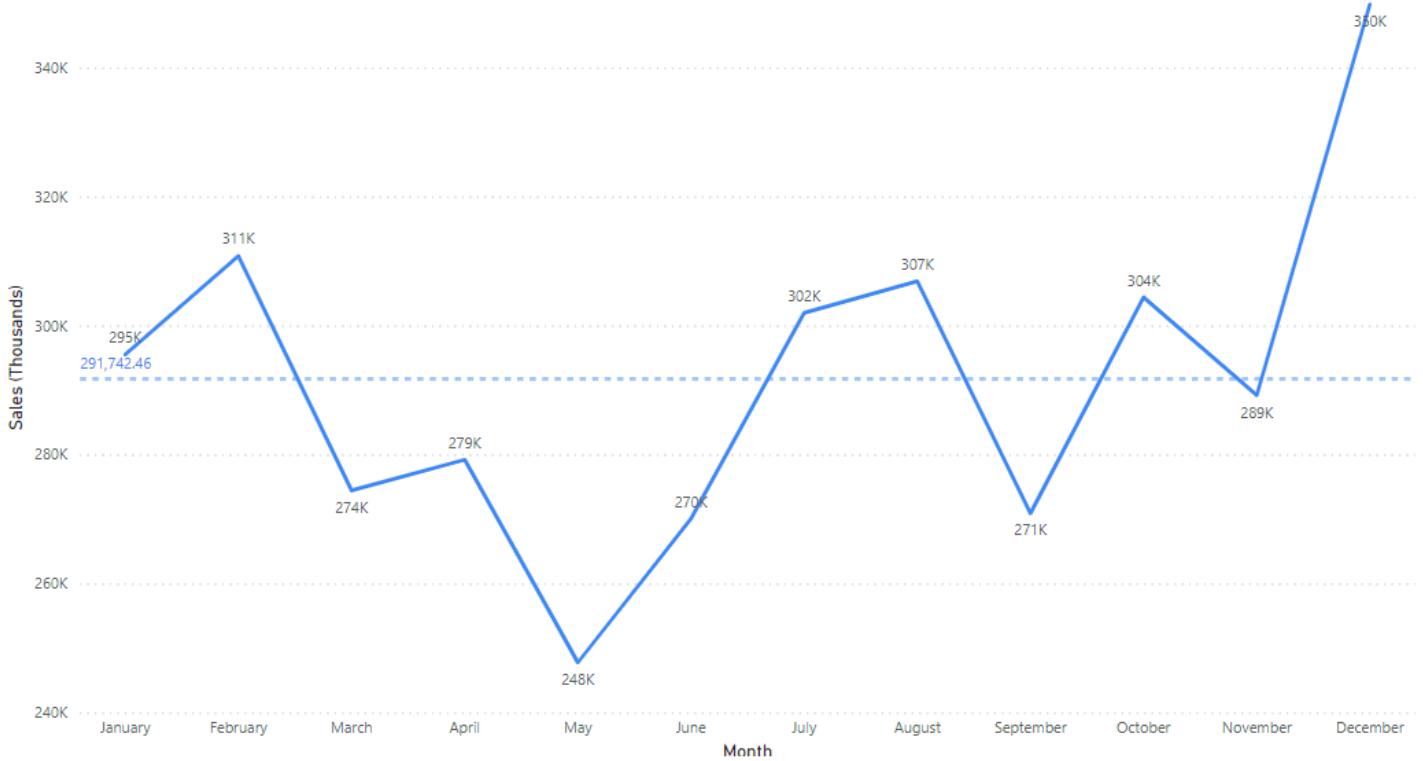
MONTH	Sales	Previous_Month	Growth
January	\$308,964.38		
February	\$271,432.76	\$308,964.38	-12.15
March	\$294,662.23	\$271,432.76	8.56
April	\$307,405.07	\$294,662.23	4.32
May	\$300,425.62	\$307,405.07	-2.27
June	\$299,272.98	\$300,425.62	-.38
July	\$282,819.37	\$299,272.98	-5.50
August	\$253,412.87	\$282,819.37	-10.40
September	\$296,752.48	\$253,412.87	17.10
October	\$274,320.34	\$296,752.48	-7.56
November	\$264,047.11	\$274,320.34	-3.74
December	\$279,174.48	\$264,047.11	5.73
Average Sales Amount	\$286,057.47		
Total Sales Amount	\$3,432,689.69		
(END OF REPORT)			

=====

End of Page 1

iv) Visualisation for Query 3:

Monthly Sales Performance in year 2023



Observation: The graph above clearly shows a significant drop in March, May, and September. May is the lowest point of the year, and December achieved the highest sales and has grown the most throughout the year. Businesses can utilize this information to optimize marketing strategies by creating targeted campaigns for low-sales months like March, May, and September, adjusting inventory and staffing levels to match seasonal demand, and allocating resources effectively to boost sales during these low periods while capitalizing on high-demand months like December to maximize overall performance.

3.2 Thong Cheng How

3.2.1 Top Ordering Companies Analysis By City In Year 2022 and 2023

i) Purpose:

This will allow the management to analyze customer spendings in the year of 2022 and 2023 by the most orders ordered in both years followed by the total amounts in both 2022 and 2023 and rank based on the number of orders made by each company in that country. To make this query more interesting, I group the customers by the country, to help the management analyze which company orders the most, also ranking the highest ranking order in each year of 2022 and 2023 will beneficially help the management to analyze the highest ranking for each country. Apart from grouping the country, a total amount column is to help the management to determine which country performs better in year 2022 or year 2023. Apart from that, I added the user input to help the management specify which country they want to visualize to help the management ease their time to find which country they want to compare.

ii) SQL Code:

```
--changes made for query 1
--remove n_orders and n_order_details OLTP tables
--added ranking order for each country
--analyze each company by 2022 and 2023
--made changes to the title of the report
--added formatting to the query
--remove the currentrowindicator
--remove the left join
--remove the view
--added a prompt for user input
```

```
cl scr
SET VERIFY OFF;
set pagesize 150
set linesize 160
clear columns
clear breaks
clear computes
```

```
PROMPT Enter the country name or 'ALL' to analyze all countries:
ACCEPT country_name CHAR PROMPT 'Country Name: '
```

```
DEFINE page_count = 1
```

```
REPFOOTER SKIP 1'
( END OF REPORT )'
```

TTITLE

```
'=====
===== SKIP 1-
' Top Ordering Companies Analysis By City
In Year 2022 and 2023 for ('&country_name') country           ' SQL.PNO " /
&page_count" SKIP 1-
'=====
===== -
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'=====
===== -

```

BTITLE

```
'=====
===== SKIP 1-
'
End of
Page' FORMAT 9 SQL.PNO SKIP 1-
'=====
===== -
' SKIP 2-

```

```
--ACCEPT country_name PROMPT 'Enter the country name or "ALL" to view all
country for analysis: '
```

```
BREAK ON country SKIP 2 ON report
```

```
column company_name format A25 heading 'Company Name'
column country format a15 heading 'Country'
column city heading 'City'
column total_number_of_orders_2022 format 99,999 heading 'Orders (2022)'
column total_number_of_orders_2023 format 99,999 heading 'Orders (2023)'
column total_amount_2022 format $9,999,999.99 heading 'Total Amount (2022)'
column total_amount_2023 format $9,999,999.99 heading 'Total Amount (2023)'
column rank_category format a19 heading 'Ranking (Orders)'
```

```
COMPUTE SUM LABEL 'SUM' OF total_amount_2022 on report
COMPUTE SUM OF total_amount_2023 on report
COMPUTE SUM OF total_number_of_orders_2022 on report
COMPUTE SUM OF total_number_of_orders_2023 on report
```

```
WITH ranked_companies AS (
  SELECT
    CD.comp_name AS company_name,
    CD.country AS country,
    CD.city AS city,
    ROUND(SUM(CASE WHEN D.cal_year = 2022 THEN order_fact.line_total ELSE 0
END), 2) AS total_amount_2022,
```

```

        ROUND(SUM(CASE WHEN D.cal_year = 2023 THEN order_fact.line_total ELSE 0
END), 2) AS total_amount_2023,
        COUNT(CASE WHEN D.cal_year = 2022 THEN order_fact.orderid END) AS
total_number_of_orders_2022,
        COUNT(CASE WHEN D.cal_year = 2023 THEN order_fact.orderid END) AS
total_number_of_orders_2023
    FROM
        customer_dim CD
    JOIN order_fact ON CD.customer_key = order_fact.customer_key
    JOIN Date_dim D ON order_fact.date_key = D.date_key
    WHERE D.cal_year IN (2022, 2023) AND (UPPER('&country_name') = 'ALL' OR
UPPER(CD.country) = UPPER('&country_name'))
    GROUP BY CD.comp_name, CD.country, CD.city
)
SELECT
    company_name,
    country,
    city,
    total_number_of_orders_2022,
    total_amount_2022 AS Total_Amount_2022,
    total_number_of_orders_2023,
    total_amount_2023 AS Total_Amount_2023,
    CASE
        WHEN total_number_of_orders_2022 = 1 OR total_number_of_orders_2023 = 1
THEN '-'
        WHEN total_number_of_orders_2022 = MAX(total_number_of_orders_2022)
OVER (PARTITION BY country) THEN
            CASE
                WHEN total_number_of_orders_2023 =
MAX(total_number_of_orders_2023) OVER (PARTITION BY country) THEN 'Both
Highest'
                ELSE 'Highest in 2022'
            END
        WHEN total_number_of_orders_2023 = MAX(total_number_of_orders_2023)
OVER (PARTITION BY country) THEN 'Highest in 2023'
        ELSE '-'
    END AS rank_category
FROM ranked_companies
ORDER BY country, rank_category;

```

PROMPT Script Execution Completed

```

clear columns
clear breaks
clear computes
ttitle off
btitle off

```

iii) Output for Query 1:

Enter the country name or 'ALL' to analyze all countries:
Country Name: all

Top Ordering Companies Analysis By City In Year 2022 and 2023 for (all) country

1 / 1

Report Date: 14-MAY-24

Company Name	Country	City	Orders (2022)	Total Amount (2022)	Orders (2023)	Total Amount (2023)	Ranking (Orders)
Cactus Comidas para Rancho grande	Argentina	Buenos Aires	256	\$35,465.00	266	\$37,678.65	-
Oceano Atlantico Ltd		Buenos Aires	307	\$46,019.67	207	\$30,439.74	Highest in 2022
		Buenos Aires	304	\$45,923.83	274	\$37,507.16	Highest in 2023
Ernst Handel Piccolo und mehr	Austria	Graz	246	\$37,509.45	239	\$33,825.76	-
		Salzburg	274	\$42,775.52	273	\$35,331.93	Both Highest
Supremes delices Maison Dewey	Belgium	Charleroi	292	\$37,904.36	230	\$28,285.48	Highest in 2022
		Bruxelles	226	\$31,179.37	247	\$35,414.59	Highest in 2023
Gourmet Lanchonetes Familia Arquibaldo Comercio Mineiro Hanari Carnes Wellington Importado Ricardo Adocicados Que Delicia Tradicao Hipermecad Queen Cozinha	Brazil	Campinas	294	\$44,924.64	262	\$31,971.73	-
		Sao Paulo	249	\$31,116.29	259	\$33,318.49	-
		Sao Paulo	279	\$40,297.74	261	\$37,293.32	-
		Rio de Janeiro	256	\$34,834.15	271	\$36,185.12	-
		Resende	235	\$36,725.82	260	\$32,178.70	-
		Rio de Janeiro	293	\$37,894.69	261	\$40,815.80	-
		Rio de Janeiro	275	\$39,549.17	231	\$26,422.72	-
		Sao Paulo	306	\$48,445.23	221	\$30,248.28	Highest in 2022
		Sao Paulo	269	\$37,278.67	302	\$39,352.10	Highest in 2023
Mere Paillarde Bottom-Dollar Market Laughing Bacchus Win	Canada	Montreal	231	\$32,846.67	235	\$35,097.53	-
		Tsawassen	287	\$44,082.04	220	\$30,690.95	Highest in 2022
		Vancouver	251	\$35,328.53	258	\$33,472.15	Highest in 2023
Simons bistro	Denmark	Kobenhavn	281	\$37,934.82	261	\$36,590.89	Highest in 2022
Wartian Herkku Wilman Kala	Finland	Oulu	254	\$37,802.84	309	\$41,794.14	-
		Helsinki	264	\$34,105.86	310	\$43,983.00	Both Highest
Bon app Vins et alcools Chev Victuailles en stock Specialites du monde Paris specialites La corne d'abondance France restauration Folies gourmandes Du monde entier Blondel pere et fils La maison d'Asie	France	Marseille	296	\$39,769.17	264	\$36,905.29	-
		Reims	237	\$32,956.65	256	\$40,154.09	-
		Lyon	303	\$42,052.81	267	\$37,647.11	-
		Paris	243	\$38,212.48	250	\$32,546.21	-
		Paris	256	\$32,928.14	218	\$33,511.88	-
		Versailles	251	\$31,655.82	251	\$34,625.76	-
		Nantes	233	\$34,849.90	241	\$34,612.71	-
		Lille	306	\$38,928.00	219	\$26,574.61	-
		Nantes	293	\$40,179.69	265	\$34,394.01	-
		Strasbourg	317	\$43,646.70	252	\$37,642.86	Highest in 2022
		Toulouse	315	\$45,836.11	309	\$47,765.22	Highest in 2023
Alfreds Futterkiste Blauer See Delikates Morgenstern Gesundko Lehmanns Marktstand Ottilie's Käseladen Königlich Essen Die Wandernde Kuh Frankenversand Drachenblut Delikate Toms Spezialitäten QUICK-Stop	Germany	Berlin	284	\$36,285.68	294	\$43,835.47	-
		Mannheim	280	\$36,435.82	282	\$35,243.33	-
		Leipzig	217	\$29,457.72	253	\$35,446.44	-
		Frankfurt a.M.	274	\$42,171.16	236	\$28,773.28	-
		Köln	263	\$36,028.97	240	\$33,177.19	-
		Brandenburg	304	\$43,139.65	268	\$32,433.28	-
		Stuttgart	234	\$43,525.09	285	\$34,173.80	-
		München	211	\$27,719.92	269	\$32,502.40	-
		Aachen	258	\$35,824.82	264	\$33,333.84	-
		Münster	320	\$43,189.19	289	\$40,120.11	Highest in 2022
		Cunewalde	315	\$45,410.30	305	\$37,508.75	Highest in 2023
Hungry Owl All-Night	Ireland	Cork	299	\$36,771.02	268	\$39,515.02	Both Highest
Reggiani Caseifici Magazzini Alimentari Franchi S.p.A.	Italy	Reggio Emilia	256	\$38,820.61	283	\$39,762.08	-
		Bergamo	274	\$34,898.89	228	\$26,533.09	Highest in 2022
		Torino	266	\$36,349.16	297	\$41,585.30	Highest in 2023
Centro comercial Moc	Mexico	Mexico D.F.	283	\$45,693.64	283	\$41,522.30	-

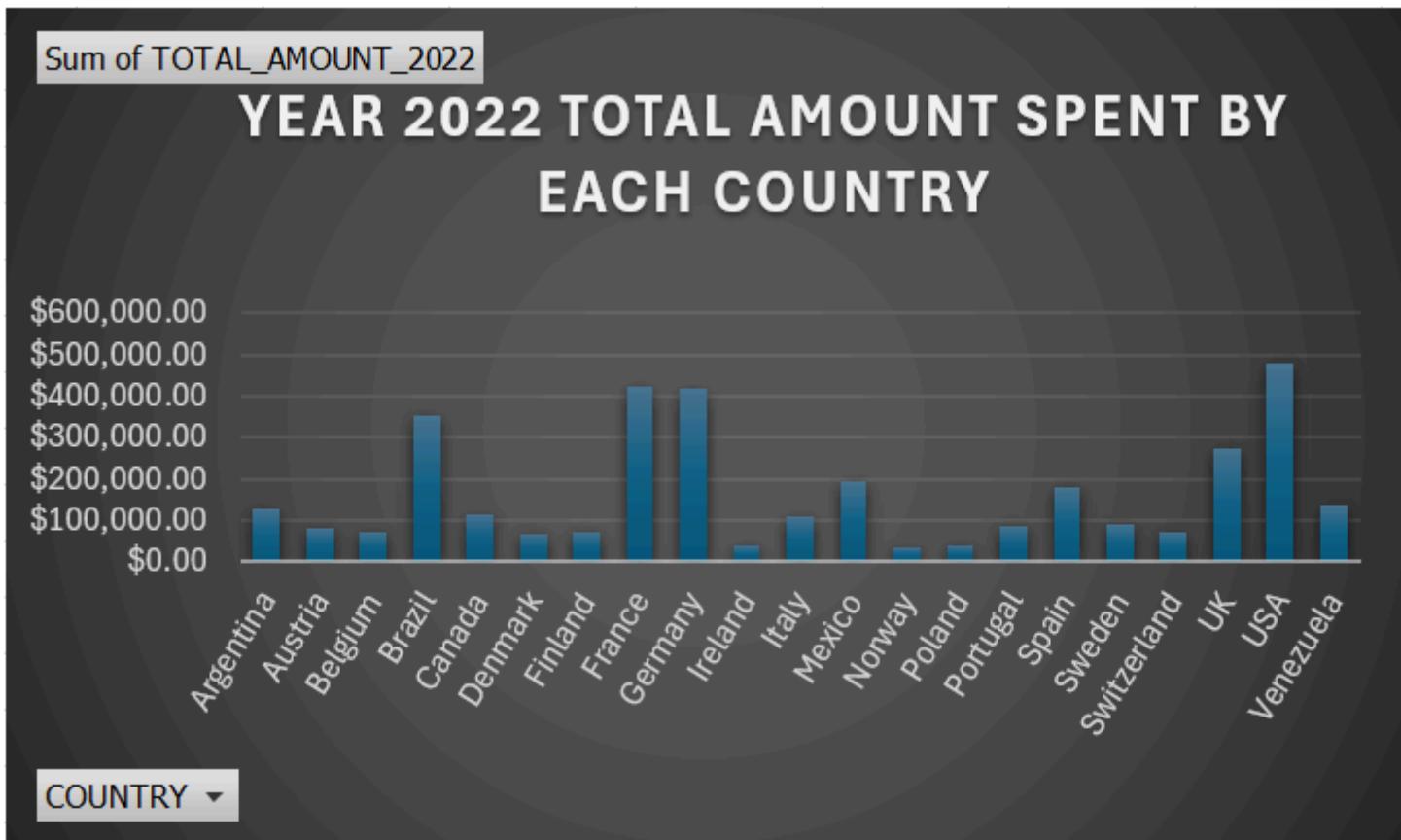
Mexico			Mexico D.F.	262	\$34,708.30	230	\$28,712.85 -
Tortuga Restaurante			Mexico D.F.	248	\$34,283.05	269	\$37,147.72 -
Pericles Comidas clásicas			Mexico D.F.	280	\$44,086.08	251	\$34,323.23 -
Antonio Moreno Taquería			Mexico D.F.	291	\$35,874.21	288	\$43,722.76 Both Highest
Ana Trujillo Emparedados							
Sante Gourmet	Norway	Stavern		262	\$34,201.16	284	\$43,757.22 Both Highest
Wolski Zajazd	Poland	Warszawa		238	\$37,583.24	320	\$44,876.06 Both Highest
Princesa Isabel Vinh	Portugal	Lisboa		263	\$34,574.07	247	\$32,141.65 -
Furia Bacalhau e Fru		Lisboa		310	\$52,137.05	261	\$41,370.91 Both Highest
Bolido Comidas preparadas	Spain	Madrid		257	\$34,984.52	279	\$39,227.91 -
FISSA Fabrica Inter.		Madrid		242	\$36,027.87	229	\$38,346.16 -
Galeria del gastronómico		Barcelona		244	\$27,470.87	243	\$34,540.30 -
Romero y tomillo		Madrid		297	\$38,863.42	233	\$32,386.71 Highest in 2022
Godos Cocina Tipica		Sevilla		281	\$40,093.50	307	\$44,265.38 Highest in 2023
Folk och HB	Sweden	Bräcke		286	\$40,639.03	217	\$28,301.95 Highest in 2022
Berglunds snabbköp		Luleå		281	\$47,526.64	272	\$39,061.44 Highest in 2023
Chop-suey Chinese	Switzerland	Bern		267	\$37,888.55	255	\$35,897.88 Both Highest
Richter Supermarkt		Geneve		247	\$32,550.81	255	\$34,933.02 Highest in 2023
B's Beverages	UK	London		276	\$44,445.01	229	\$29,217.29 -
Seven Seas Imports		London		270	\$32,371.53	291	\$36,861.34 -
Island Trading		Cowes		263	\$39,896.43	246	\$34,973.97 -
Eastern Connection		London		267	\$36,483.54	247	\$30,553.87 -
Consolidated Holding		London		222	\$31,324.87	268	\$37,292.48 -
North/South		London		300	\$42,784.76	242	\$29,534.96 Highest in 2022
Around the Horn		London		276	\$46,109.73	313	\$40,643.99 Highest in 2023

Old World Delicatess	USA	Anchorage	287	\$38,550.37	290	\$40,293.99 -
Trail's Head Gourmet		Kirkland	299	\$41,294.12	290	\$43,191.70 -
Hungry Coyote Import		Elgin	264	\$33,756.77	208	\$29,165.84 -
Lazy K Kountry Store		Walla Walla	266	\$40,473.15	215	\$33,934.60 -
Let's Stop N Shop		San Francisco	252	\$38,062.85	252	\$38,701.99 -
Lonesome Pine Restau		Portland	238	\$29,134.60	292	\$42,891.41 -
Rattlesnake Canyon G		Albuquerque	288	\$44,565.70	256	\$36,413.60 -
Save-a-lot Markets		Boise	232	\$26,401.80	277	\$38,462.28 -
Split Rail Beer & Al		Lander	251	\$34,621.04	276	\$39,334.94 -
The Big Cheese		Portland	286	\$38,809.36	279	\$36,927.67 -
The Cracker Box		Butte	217	\$30,141.07	222	\$25,653.84 -
White Clover Markets		Seattle	311	\$37,573.14	309	\$50,920.96 Both Highest
Great Lakes Food Mar		Eugene	311	\$46,228.87	258	\$36,373.97 Highest in 2022
GROSELLA-Restaurante	Venezuela	Caracas	269	\$35,942.62	269	\$39,779.29 -
LILA-Supermercado		Barquisimeto	242	\$34,850.08	228	\$36,192.38 -
HILARION-Abastos		San Cristobal	296	\$35,241.95	239	\$35,156.85 Highest in 2022
LINO-Delicatessen		I. de Margarita	275	\$32,907.44	281	\$42,401.82 Highest in 2023

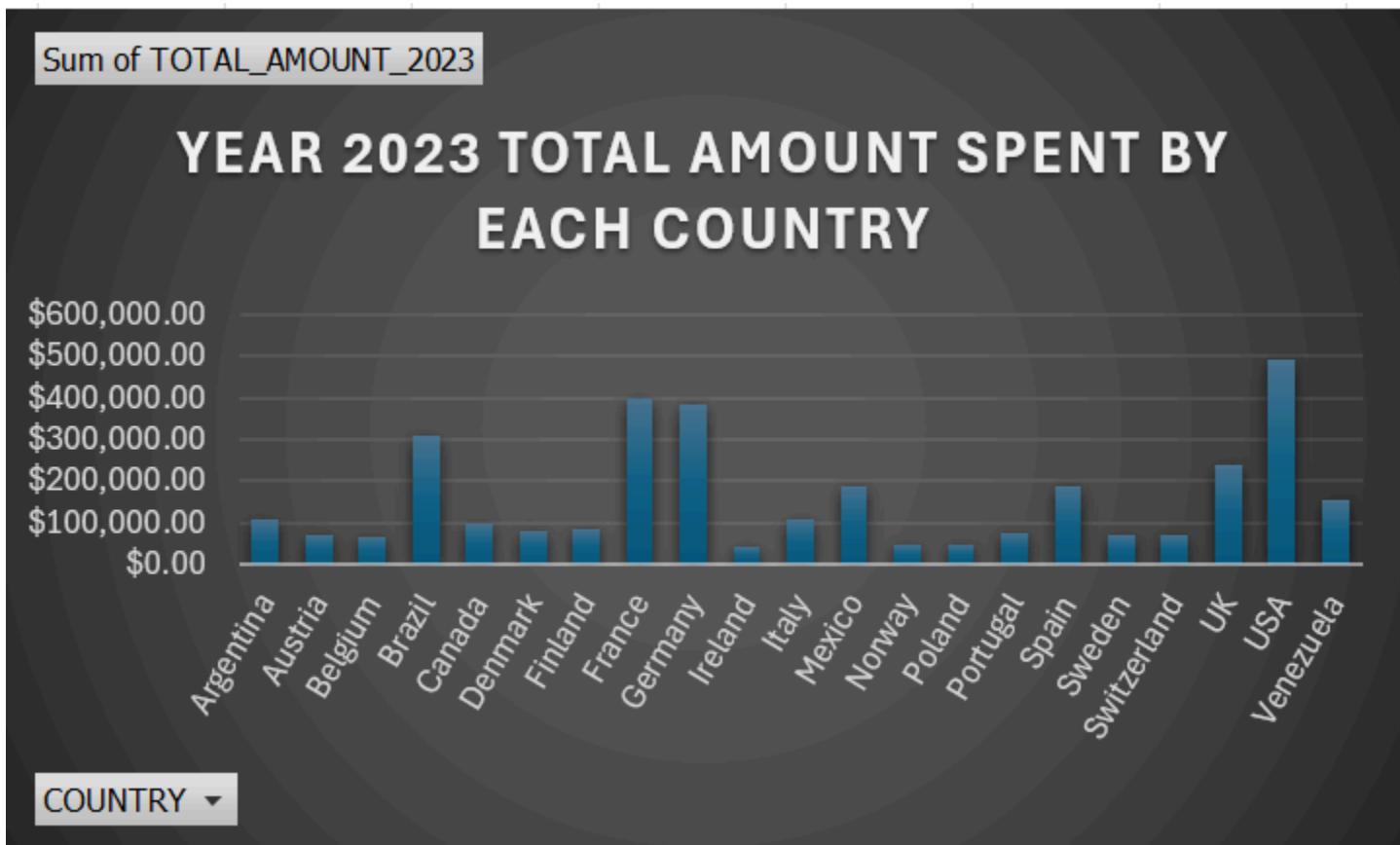
SUM			24,566	\$3,447,880.19	23,774	\$3,295,927.72
(END OF REPORT)						
=====						
End of Page 1						
=====						

91 rows selected.

Script Execution Completed

iv) Visualization for Query 1:

Observation: This chart helps the management to interpret the total amount by each distinct country. Based on the chart, the USA makes the most amount of sales in the year 2022. France and Germany have the second highest amount of sales in 2022, followed by Brazil and followed by the UK. Ireland, Norway and Poland who came in last in the total number of sales by each distinct country. This will help the management to decide which country to focus on in the coming years, because these locations can vary by location and by cost of transport.



Observation: This chart helps the management to determine which country spends the most in 2023. Knowing the country will help the management to build strong relationships with high spending customers by offering discounts for products or services in terms of supplier transportation. This could build a strong relationship between the customer and the management. As for less spending customers, the management could dive deeper into knowing the customers background, location. To determine why they are spending less compared to previous years. However in this case the top spenders are from the USA and the lowest spenders are from Ireland, Norway and Poland. The management can use this information to determine why this is happening.

3.2.2 Employee performance report comparison year 2022 vs year 2023

i) Purpose:

This query is intended to analyze the performance for each employee. The management can use this query to identify the sales of each employee and the number of customers each employee managed in that specific area of the management. The sales of 2022 and 2023 made by each employee can be used to compare their ability to help the company in terms of return of investment of each specific employee.

ii) SQL code:

```
--changes made for query 2
--remove the cross joins
--remove the currentrowindicator
--added prompt and user input
--added formatting to the query
```

```
cl scr
set linesize 140
set pagesize 30
SET UNDERLINE ON
SET UNDERLINE =
clear columns
clear breaks
clear computes

PROMPT Enter the employee name or 'ALL' to analyze all employees:
ACCEPT emp_name CHAR PROMPT 'Employee Name: '

DEFINE page_count = 1
REPFOOTER '
END OF REPORT )'
TTITLE
'=====
===== SKIP 1-
'          Employee Performance Report In Year 2022 and 2023 for
('&emp_name') Employee           ' SQL.PNO " / &page_count" SKIP 1-
'=====
===== -
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'=====
===== -
BTITLE
'=====
===== SKIP 1-
'          End of
Page' FORMAT 9 SQL.PNO SKIP 1-
'=====
===== SKIP 2-

--ACCEPT emp_name PROMPT 'Enter the employee name or "ALL" to view all
employees: '

column employee_name format A18 heading 'Employee Name'
column location format a15 heading 'Location'
column total_sales_2022 format $9,999,999.00 heading 'Total Sales 2022'
COLUMN sales_percentage_2022 FORMAT 999.99 HEADING 'Sales PCT 2022 (%)'
column total_sales_2023 format $9,999,999.00 heading 'Total Sales 2023'
COLUMN sales_percentage_2023 FORMAT 999.99 HEADING 'Sales PCT 2023 (%)'
column num_customers format 99,999 heading 'Total Customers'
```

```
column customer_percentage format 999.99 heading 'CUST PCT (%)'
```

```
BREAK ON REPORT SKIP 2
COMPUTE SUM LABEL 'SUM' OF total_sales_2022 on report
COMPUTE SUM OF total_sales_2023 on report
COMPUTE SUM OF num_customers on report
COMPUTE SUM OF sales_percentage_2022 on report
COMPUTE SUM OF sales_percentage_2023 on report
```

```
SELECT
```

```
    EMP.employee_name,
    EMP.location,
    EMP.num_customers,
    EMP.total_sales_2022,
    ROUND((EMP.total_sales_2022 / TOTAL_SALES.TOTAL_SALES_2022) * 100, 2) AS
sales_percentage_2022,
    EMP.total_sales_2023,
    ROUND((EMP.total_sales_2023 / TOTAL_SALES.TOTAL_SALES_2023) * 100, 2) AS
sales_percentage_2023
FROM (
    SELECT
        ED.employeeename AS employee_name,
        ED.country || ', ' || ED.city AS location,
        COUNT(DISTINCT order_fact.customer_key) AS num_customers,
        SUM(CASE WHEN EXTRACT(YEAR FROM DD.cal_date) = 2022 THEN
order_fact.line_total ELSE 0 END) AS total_sales_2022,
        SUM(CASE WHEN EXTRACT(YEAR FROM DD.cal_date) = 2023 THEN
order_fact.line_total ELSE 0 END) AS total_sales_2023,
        ROW_NUMBER() OVER (PARTITION BY ED.employee_key ORDER BY
SUM(order_fact.line_total) DESC) AS sales_rank
    FROM
        employee_dim ED
    JOIN
        order_fact ON ED.employee_key = order_fact.employee_key
    JOIN
        date_dim DD ON order_fact.date_key = DD.date_key
    WHERE
        UPPER('&emp_name') = 'ALL' OR UPPER(ED.employeeename) =
UPPER('&emp_name')
    GROUP BY
        ED.employee_key, ED.employeeename, ED.country, ED.city
) EMP
CROSS JOIN (
    SELECT
        SUM(CASE WHEN EXTRACT(YEAR FROM DD.cal_date) = 2022 THEN
order_fact.line_total ELSE 0 END) AS TOTAL_SALES_2022,
        SUM(CASE WHEN EXTRACT(YEAR FROM DD.cal_date) = 2023 THEN
order_fact.line_total ELSE 0 END) AS TOTAL_SALES_2023
```

```

    FROM
        order_fact
    JOIN
        date_dim DD ON order_fact.date_key = DD.date_key
) TOTAL_SALES
WHERE
    EMP.sales_rank = 1
ORDER BY
    employee_name;

```

PROMPT Script Execution Completed

```

clear columns
clear breaks
clear computes
ttitle off
btitle off

```

iii) Output for query 2:

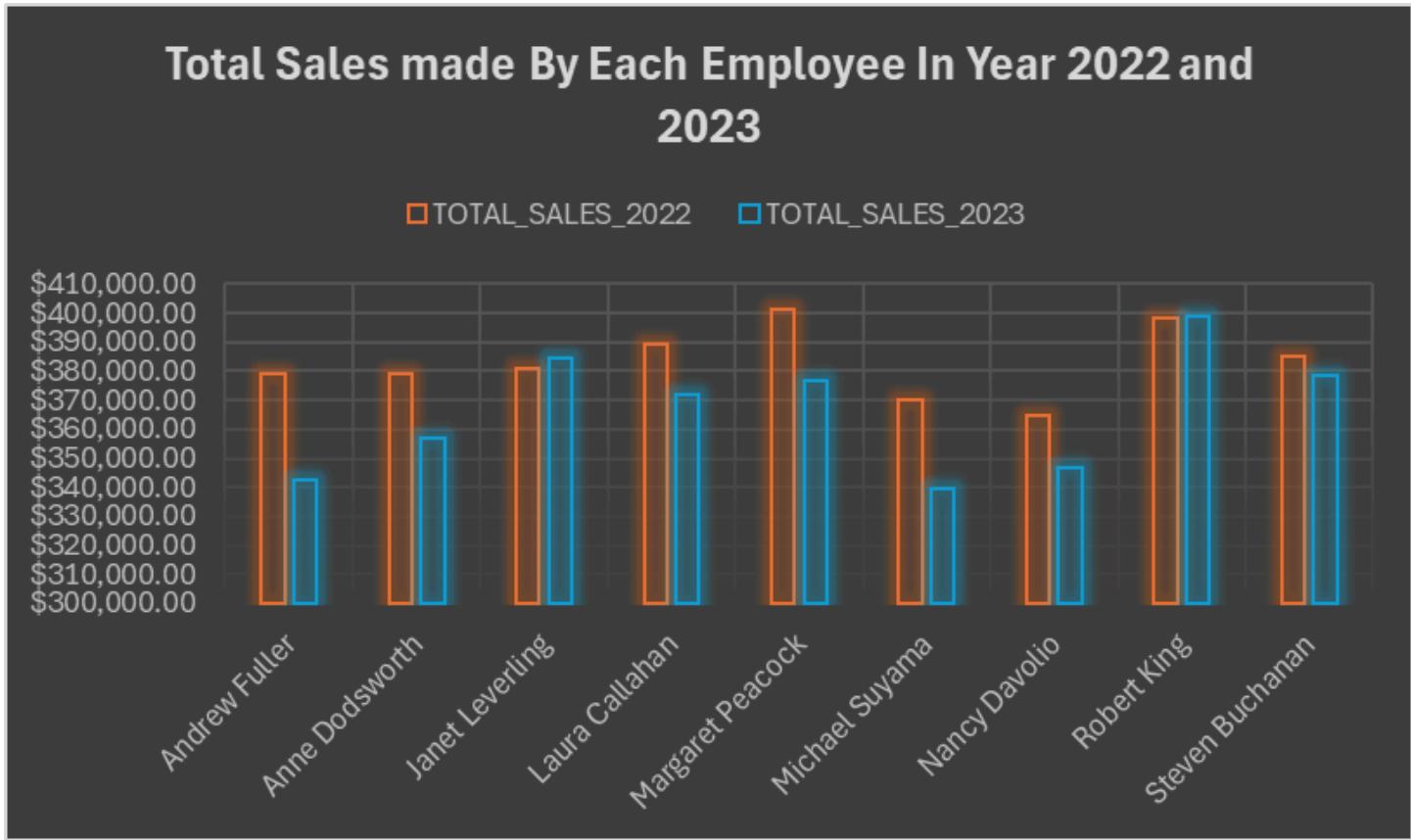
Employee Performance Report In Year 2022 and 2023 for (ALL) Employee							1 / 1
Employee Name	Location	Total Customers	Total Sales 2022	Sales PCT 2022 (%)	Total Sales 2023	Sales PCT 2023 (%)	
Andrew Fuller	USA, TACOMA	4,770	\$378,799.47	10.99	\$342,814.28	10.40	
Anne Dodsworth	UK, LONDON	4,804	\$378,999.32	10.99	\$356,740.65	10.82	
Janet Leverling	USA, KIRKLAND	4,827	\$381,058.96	11.05	\$384,682.60	11.67	
Laura Callahan	USA, SEATTLE	4,773	\$389,196.86	11.29	\$371,635.74	11.28	
Margaret Peacock	USA, REDMOND	4,746	\$401,538.47	11.65	\$376,846.68	11.43	
Michael Suyama	UK, LONDON	4,764	\$370,128.89	10.73	\$339,276.74	10.29	
Nancy Davolio	USA, SEATTLE	4,787	\$364,847.62	10.58	\$346,850.41	10.52	
Robert King	UK, LONDON	4,774	\$398,397.46	11.55	\$398,667.28	12.10	
Steven Buchanan	UK, LONDON	4,791	\$384,913.14	11.16	\$378,413.34	11.48	
SUM		43,036	\$3,447,880.19	99.99	\$3,295,927.72	99.99	

(END OF REPORT)

End of Page 1

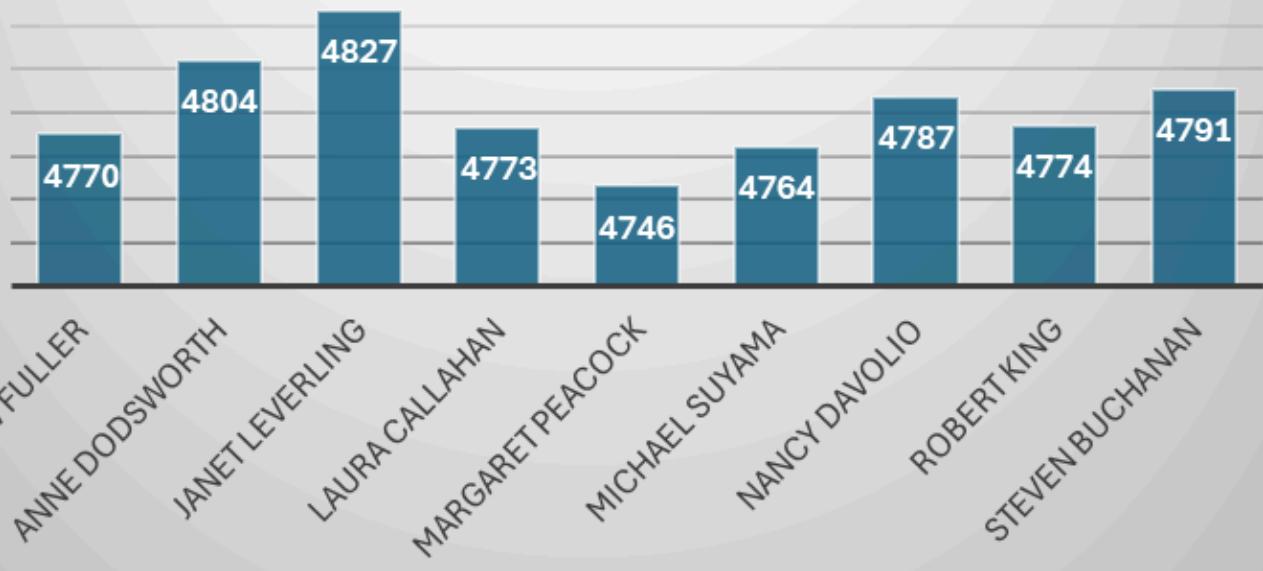
9 rows selected.

Script Execution Completed

iv) Visualization for Query 2:

Observation: This chart helps the management to compare each employee's total sales in the year 2022 and year 2023. The orange horizontal line indicates sales made in 2022 and the light blue line's indicate the sales made in 2023. From the chart above, Nancy Davolio, Michael Suyama, Margaret Peacock, Anne Dodsworth and Andrew Fuller reduce sales in 2023 compared to 2022. As only Janet Leverling improve in sales in 2023.

Total Number of Customer Handled By Each Employee



Observation: This chart shows the number of customers each employee had handled in the year of 2022 and year 2023. From the chart, Janet Leverling had handled the most customers. Margaret Peacock handled the least customers. This shows that Janet sales greatly influenced the number of sales on the number of customers Janet is handling, this could be because of lack of manpower in Janet area. The management can help Janet by employing more employees to that area or transfer some employees to help Janet out.

3.2.3 Best Performing Product Analysis Shipped by Suppliers In Year 2022 and comparison in year 2023

i) Purpose:

This query will help the management to determine which supplier supplied what products the most in that specific country and city. As well as information such as order change rate of that product comparison in year 2023 compared to the year 2022. This query also gives information such as total number of orders of that product in the year of 2023.

ii) SQL code:

```
--changes made for query 3
--remove product percentage count
--the country is now delivered to customer instead of supplier location
--added a prompt to allow user to enter
--remove the productpercentagecount and replaced it to order change
--added two new columns to compare in 2022 and 2023
--remove the view
--added formatting to the query
--change the title for the query
```

```
cl scr
```

```
set linesize 220
set pagesize 50
SET UNDERLINE ON
SET UNDERLINE =
clear columns
clear breaks
clear computes
```

```
PROMPT Enter the country name or 'ALL' to analyze all countries:
ACCEPT country_name CHAR PROMPT 'Country Name: '
```

```
DEFINE page_count = 1
REPFOOTER '
END OF REPORT )'
TTITLE
'=====
====='
===== SKIP 1-
'                               Best Performing Product Analysis Shipped by
Supplier In Year 2022 and 2023 for ('&country_name') country
SQL.PNO " / &page_count" SKIP 1-
'=====
====='
===== -
```

```
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
=====
=====
====='
-
BTITLE
=====
=====
====='
SKIP 1-
'
End of
Page' FORMAT 9 SQL.PNO SKIP 1-
=====
=====
====='
SKIP 2-
```

```
column suppliername format A40 heading 'Supplier Name'
column country format a16 heading 'Customer Country'
column city format a11 heading 'City'
COLUMN bestperformingproduct FORMAT a35 HEADING 'Best Product'
column totalproducts_2022 format 999,999 heading 'Orders (2022)'
column totalproducts_2023 format 999,999 heading 'Orders (2023)'
column totalproducts format 999,999 heading 'TTL Orders'
column percentagechange heading 'Order Change (%)'
column order_year format 9999 heading 'Year'
BREAK ON REPORT SKIP 2
```

```
--ACCEPT country_name PROMPT 'Enter the country name or "ALL" to view all
country for analysis: '
```

```
WITH SupplierTotalProducts AS (
    SELECT
        s.SUPPLIERNAME,
        p.PRODUCTNAME,
        c.COUNTRY,
        c.CITY,
        COUNT(*) AS TotalProducts,
        SUM(CASE WHEN dd.CAL_YEAR = 2022 THEN 1 ELSE 0 END) AS
TotalProducts_2022,
        SUM(CASE WHEN dd.CAL_YEAR = 2023 THEN 1 ELSE 0 END) AS
TotalProducts_2023,
        ROW_NUMBER() OVER (PARTITION BY s.SUPPLIERNAME ORDER BY COUNT(*) DESC)
AS ProductRank
    FROM
        product_dim p
    JOIN
```

```
        order_fact o ON p.PRODUCT_KEY = o.PRODUCT_KEY
    JOIN
        supplier_dim s ON o.SUPPLIER_KEY = s.SUPPLIER_KEY
    JOIN
        date_dim dd ON o.DATE_KEY = dd.DATE_KEY
    JOIN
        customer_dim c ON o.CUSTOMER_KEY = c.CUSTOMER_KEY
WHERE
    dd.CAL_YEAR IN (2022, 2023) AND (UPPER('&country_name') = 'ALL' OR
    UPPER(c.country) = UPPER('&country_name'))
    GROUP BY
        s.SUPPLIERNAME, p.PRODUCTNAME, c.COUNTRY, c.CITY
),
RankedProducts AS (
    SELECT
        STP.SUPPLIERNAME,
        STP.PRODUCTNAME,
        STP.COUNTRY,
        STP.CITY,
        STP.TotalProducts,
        STP.TotalProducts_2022,
        STP.TotalProducts_2023,
        STP.ProductRank,
        ROW_NUMBER() OVER (PARTITION BY STP.SUPPLIERNAME ORDER BY
        STP.ProductRank) AS SupplierRank
    FROM
        SupplierTotalProducts STP
)
SELECT
    RP.SUPPLIERNAME,
    RP.PRODUCTNAME AS BestPerformingProduct,
    RP.COUNTRY,
    RP.CITY,
    RP.TotalProducts,
    RP.TotalProducts_2022,
    RP.TotalProducts_2023,
    ROUND((RP.TotalProducts_2023 - RP.TotalProducts_2022) * 100.0 /
    NULLIF(RP.TotalProducts_2022, 0), 2) AS PercentageChange
FROM
    RankedProducts RP
WHERE
    RP.SupplierRank = 1
ORDER BY
    RP.SUPPLIERNAME;
```

PROMPT Script Execution Completed

```
clear columns
clear breaks
clear computes
ttitle off
btitle off
```

iii) Output for query 3:

Best Performing Product Analysis Shipped by Supplier In Year 2022 and 2023 for (all) country							1 / 1
Report Date: 14-MAY-24	Supplier Name	Best Product	Customer Country	TTL Orders (2022)	Orders (2023)	Order Change (%)	
	AUX JOYEUX ECCLESIASTIQUES	CHARTREUSE VERTE	UK	London	59	30	-3.33
	BIGFOOT BREWERIES	LAUGHING LUMBERJACK LAGER	UK	London	44	20	20
	COOPERATIVA DE QUESOS LAS CABRAS	QUESO MANCHEGO LA PASTORA	UK	London	50	21	38.1
	ESCARGOTS NOUVEAUX	ESCARGOTS DE BOURGOGNE	Mexico	Mexico D.F.	47	24	-4.17
	EXOTIC LIQUIDS	CHAI	UK	London	46	22	9.09
	FORETS D'ERABLES	SIROP D'ERABLE	UK	London	43	24	-20.83
	FORMAGGI FORTINI S.R.L.	MASCARPONE FABIOLI	UK	London	50	29	-27.59
	G'DAY, MATE	FILÔ MIX	UK	London	47	28	-32.14
	GAÏ PATURAGE	RACLETTE COURDAVVAULT	UK	London	52	31	-32.26
	GRANDMA KELLY'S HOMESTEAD	NORTHWOODS CRANBERRY SAUCE	UK	London	53	25	12
	HELI SUBWAREN GMBH & CO. KG	GUMBAR GUMMIBARCHEN	UK	London	36	18	160
	KARKKI OY	VALKOINEN SUKLAAN	UK	London	46	24	-8.33
	LEKA TRADING	SINGAPOREAN HOKKIEN FRIED MEE	Mexico	Mexico D.F.	47	28	-32.14
	LYNGBYSLID	ROGEDE SILD	UK	London	52	26	0
	MA MAISON	TOURIERE	Mexico	Mexico D.F.	38	14	71.43
	MAYUMI'S	GENEN SHOYU	Mexico	Mexico D.F.	46	23	0
	NEW ENGLAND SEAFOOD CANNERY	BOSTON CRAB MEAT	UK	London	48	28	-28.57
	NEW ORLEANS CAJUN DELIGHTS	CHEF ANTON'S GUMBO MIX	UK	London	49	19	57.89
	NORD-OST-FISCH HANDELSGESELLSCHAFT MBH	NORD-OST MATZESHERING	Mexico	Mexico D.F.	42	19	21.05
	NORSKE MEIERIER	GUDBRANDSDALSTØT	UK	London	44	20	20
	PASTA BUTTINI S.R.L.	GNOCCHI DI NONNA ALICE	UK	London	44	24	-16.67
	PAVLOVA, LTD.	VEGIE-SPREAD	UK	London	44	25	-24
	PB KNACKEBROD AB	TUNNBROD	UK	London	53	23	30.43
	PLUTZER LEBENSMITTELGROBMARKTE AG	ORIGINAL FRANKFURTER GRUNE SOBE	UK	London	53	29	-17.24
	REFRESCOS AMERICANAS LTDA	GUARANA FANTASTICA	UK	London	46	23	0
	SPECIALTY BISCUITS, LTD.	TEATIME CHOCOLATE BISCUITS	UK	London	51	28	-17.86
	SVENSK SJOFODA AB	ROD KAVIAR	UK	London	56	36	-44.44
	TOKYO TRADERS	LONGLIFE TOFU	UK	London	52	27	-7.41
	ZAANSE SNOEPFABRIEK	ZAANSE KOEKEN	UK	London	42	20	10

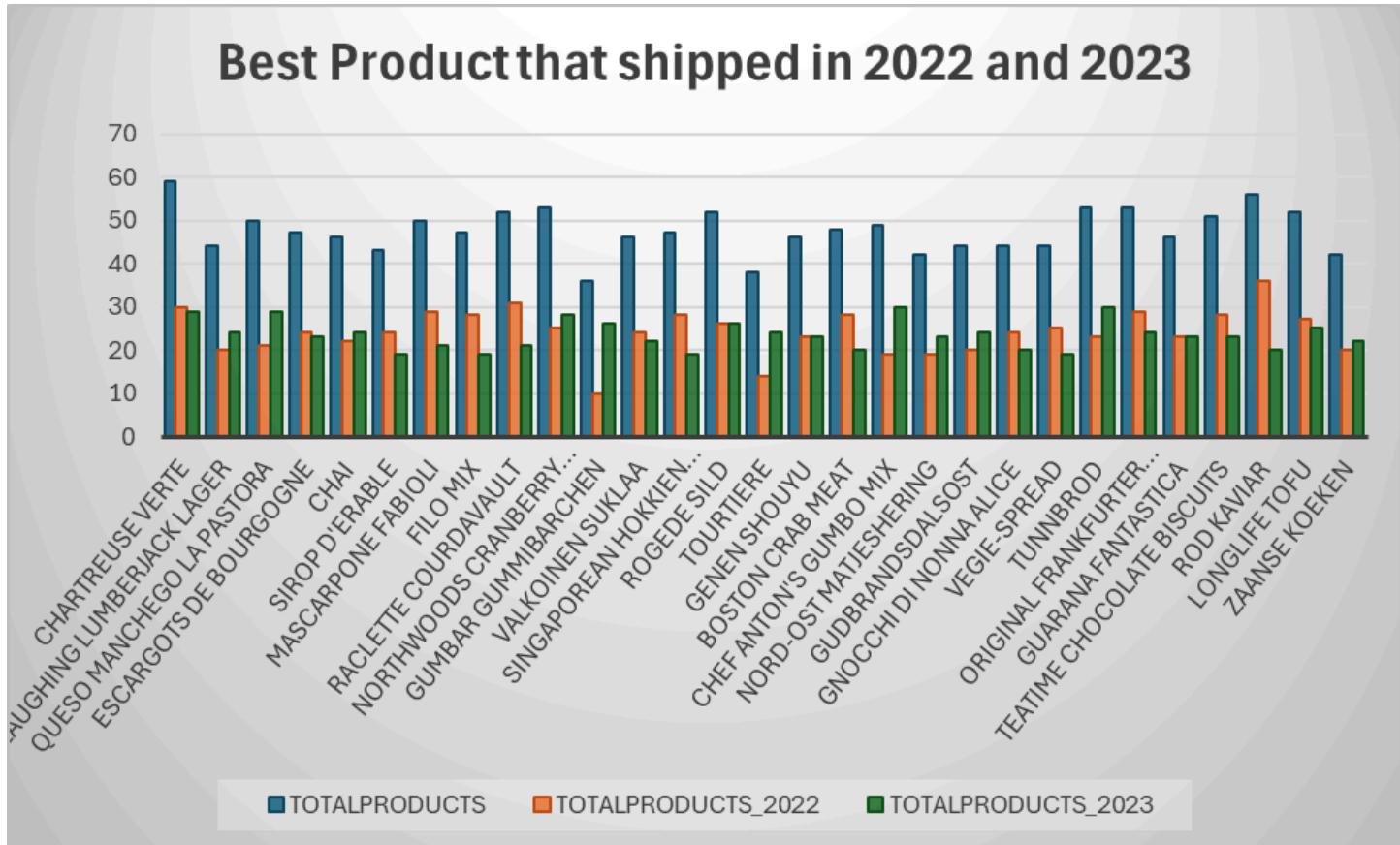
(END OF REPORT)

=====
End of Page 1
=====

29 rows selected.

Script Execution Completed

iv) Visualization for Query 3:



Observation: From this chart, the highest product count in 2022 is Raclette has the most number of orders in the year of 2022 and the highest product count in 2023 is Chef Anton's Matjeshering. Gumbar has the lowest number of products ordered in 2022 and Boston Crab Meat has the lowest number of orders in 2023. This will help the management to determine which products are less likely to order by comparing the number of orders in year 2022 and 2023 by analyzing the total products shipped to that country.

3.3 Ryan Kho Yuen Thian

3.3.1 For every month of two given years, calculate total orders, sales amount, average sales/order & differences for all or a given country

i) Purpose:

This query plays a crucial role in the company's strategic decision-making process by providing comprehensive insights into sales performance across various countries or a specific one, depending on user input. By meticulously examining monthly sales data over two designated years, the analytics team can identify trends, patterns and areas of strength or weakness in their sales operations. For example, the team could identify the months in which their sales peak or dip. This information enables them to fine-tune marketing strategies, optimise pricing and make informed decisions to enhance revenue and profitability. Furthermore, comparing sales metrics, such as yearly sales and average sales per order, between years allows for an assessment of past strategies' effectiveness.

ii) SQL Code:

```

SET VERIFY OFF;
SET PAGESIZE 44;
SET LINESIZE 480;
clear columns;
clear breaks;
clear computes;

PROMPT Enter the year(s) for analysis (comma-separated, e.g., 2022, 2023):
ACCEPT year_input CHAR PROMPT 'Year(s): '

PROMPT Enter the country name or 'ALL' to analyze all countries:
ACCEPT country_name CHAR PROMPT 'Country Name: '

DEFINE page_count = 1
REPFOOTER '
END OF REPORT )'
TTITLE
'=====
====='
SKIP 1-
'
                               Monthly Analysis of Years
&year_input for '&country_name' Country
/ &page_count" SKIP 1-
'
====='
====='

-
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'
====='
====='

-
```

```

BTITLE
=====
=====

SKIP 1-
'
          End of
Page' FORMAT 9 SQL.PNO SKIP 1-
=====
=====

SKIP 2-


COLUMN year FORMAT A10 HEADING "Year";
COLUMN cal_month_name FORMAT A10 HEADING "Month";
COLUMN total_orders FORMAT 999,999 HEADING "Total Orders";
COLUMN total_sales_amt FORMAT $999,999.99 HEADING "Total Sales Amount";
COLUMN avg_sales_amt_per_order FORMAT $999,999.99 HEADING "Avg Sales/Order";
COLUMN orders_change FORMAT 999.99 HEADING "Orders Change (%)" FORMAT A17;
COLUMN sales_change FORMAT $999,999.99 HEADING "Sales Change";
COLUMN avg_sales_change FORMAT $999,990.99 HEADING "Avg Sales Change";
COLUMN sales_change_pct FORMAT 990.99 HEADING "Sales Change (%)" FORMAT A16;
COLUMN highest_lowest_sales FORMAT A20 HEADING "Highest/Lowest Sales";
BREAK ON year SKIP 1 ON report;
COMPUTE SUM OF total_sales_amt ON year;
COLUMN total_sales_amt FORMAT $999,999,999.99 HEADING "Total Sales Amount";
COMPUTE SUM LABEL 'Total Orders: ' OF total_orders ON year;

WITH monthly_sales AS (
  SELECT
    TO_CHAR(d.cal_date, 'YYYY-MM') AS month_year,
    TO_CHAR(d.cal_year) AS year,
    d.cal_month_name AS cal_month_name,
    COUNT(DISTINCT orf.orderid) AS total_orders, -- Adjusted to count
distinct order IDs
    SUM(orf.line_total) AS total_sales_amt,
    SUM(orf.line_total) / COUNT(DISTINCT orf.orderid) AS
avg_sales_amt_per_order, -- Adjusted to calculate average sales per order
    MAX(SUM(orf.line_total)) OVER (PARTITION BY TO_CHAR(d.cal_year)) AS
max_sales,
    MIN(SUM(orf.line_total)) OVER (PARTITION BY TO_CHAR(d.cal_year)) AS
min_sales
  FROM
    order_fact orf
  JOIN
    date_dim d ON orf.date_key = d.date_key
  JOIN
    customer_dim c ON orf.customer_key = c.customer_key
  WHERE
    TO_CHAR(d.cal_year) IN (&year_input)
)

```

```

        AND (UPPER('&country_name') = 'ALL' OR UPPER(c.country) =
UPPER('&country_name'))
    GROUP BY
        TO_CHAR(d.cal_date, 'YYYY-MM'),
        TO_CHAR(d.cal_year),
        d.cal_month_name
)
SELECT
    year,
    cal_month_name,
    total_orders,
    total_sales_amt,
    avg_sales_amt_per_order,
    avg_sales_amt_per_order - LAG(avg_sales_amt_per_order) OVER (PARTITION BY
year ORDER BY month_year) AS avg_sales_change,
    CASE
        WHEN month_year LIKE '%-01' THEN TO_CHAR(ROUND(((total_orders -
LAG(total_orders) OVER (PARTITION BY year ORDER BY month_year)) /
NULLIF(LAG(total_orders) OVER (PARTITION BY year ORDER BY month_year), 0)) *
100, 2), '999.99')
        ELSE TO_CHAR(ROUND(((total_orders - LAG(total_orders) OVER (PARTITION
BY year ORDER BY month_year)) / NULLIF(LAG(total_orders) OVER (PARTITION BY
year ORDER BY month_year), 0)) * 100, 2), '990.99') || '%'
    END AS orders_change,
    total_sales_amt - LAG(total_sales_amt) OVER (PARTITION BY year ORDER BY
month_year) AS sales_change,
    CASE
        WHEN month_year LIKE '%-01' THEN TO_CHAR(ROUND(((total_sales_amt -
LAG(total_sales_amt) OVER (PARTITION BY year ORDER BY month_year)) /
NULLIF(LAG(total_sales_amt) OVER (PARTITION BY year ORDER BY month_year), 0)) *
100, 2), '999.99')
        ELSE TO_CHAR(ROUND(((total_sales_amt - LAG(total_sales_amt) OVER
(PARTITION BY year ORDER BY month_year)) / NULLIF(LAG(total_sales_amt) OVER
(PARTITION BY year ORDER BY month_year), 0)) * 100, 2), '990.99') || '%'
    END AS sales_change_pct,
    CASE
        WHEN total_sales_amt = max_sales THEN ' Highest'
        WHEN total_sales_amt = min_sales THEN ' Lowest'
        ELSE ' -'
    END AS highest_lowest_sales
FROM
    monthly_sales
ORDER BY
    year,
    month_year;

clear columns
clear breaks
clear computes

```

title off

btitle off

iii) Output 1 (For years 2022 & 2023 and ALL countries):

Enter the year(s) for analysis (comma-separated, e.g., 2022, 2023):

Year(s): 2022, 2023

Enter the country name or 'ALL' to analyze all countries:

Country Name: ALL

Monthly Analysis of Years 2022, 2023 for ALL Country											1 / 1
Report Date: 16-MAY-24											
Year	Month	Total Orders	Total Sales Amount	Avg Sales/Order	Change Sales	Orders Change (%)	Sales Change (%)	Sales Change (%)	Highest/Lowest Sales		
2022	JANUARY	909	\$281,331.66	\$309.50					-		
	FEBRUARY	771	\$257,124.27	\$333.49	\$24.00	-15.18%	-\$24,207.39	-8.60%	-		
	MARCH	666	\$220,161.92	\$330.57	-\$2.92	-13.62%	-\$36,962.35	-14.38%	Lowest		
	APRIL	901	\$278,929.59	\$309.58	-\$21.00	35.29%	\$58,767.67	26.69%	-		
	MAY	844	\$256,027.65	\$303.35	-\$6.23	-6.33%	-\$22,901.94	-8.21%	-		
	JUNE	841	\$282,221.39	\$335.58	\$32.23	-0.36%	\$26,193.74	10.23%	Highest		
	JULY	674	\$230,137.22	\$341.45	\$5.87	-19.86%	-\$52,084.17	-18.46%	-		
	AUGUST	892	\$282,041.52	\$316.19	-\$25.26	32.34%	\$51,904.30	22.55%	-		
	SEPTEMBER	892	\$281,012.09	\$315.04	-\$1.15	0.00%	-\$1,029.43	-0.36%	-		
	OCTOBER	812	\$273,719.62	\$337.09	\$22.06	-8.97%	-\$7,292.47	-2.60%	-		
	NOVEMBER	851	\$268,709.19	\$315.76	-\$21.34	4.80%	-\$5,010.43	-1.83%	-		
	DECEMBER	871	\$270,198.56	\$310.22	-\$5.54	2.35%	\$1,489.37	0.55%	-		

sum		9,924	\$3,181,614.68								
2023	JANUARY	801	\$266,870.76	\$333.17					-		
	FEBRUARY	822	\$245,593.70	\$298.78	-\$34.40	2.62%	-\$21,277.06	-7.97%	-		
	MARCH	870	\$284,452.21	\$326.96	\$28.18	5.84%	\$38,858.51	15.82%	-		
	APRIL	882	\$268,979.15	\$304.97	-\$21.99	1.38%	-\$15,473.06	-5.44%	-		
	MAY	947	\$304,999.29	\$322.07	\$17.10	7.37%	\$36,020.14	13.39%	Highest		
	JUNE	849	\$270,313.88	\$318.39	-\$3.68	-10.35%	-\$34,685.41	-11.37%	-		
	JULY	922	\$295,175.29	\$320.15	\$1.76	8.60%	\$24,861.41	9.20%	-		
	AUGUST	829	\$251,770.71	\$303.70	-\$16.44	-10.09%	-\$43,404.58	-14.70%	-		
	SEPTEMBER	781	\$244,189.09	\$312.56	\$8.86	-5.79%	-\$7,661.62	-3.04%	Lowest		
	OCTOBER	824	\$263,146.47	\$319.35	\$6.79	5.51%	\$19,037.38	7.80%	-		
	NOVEMBER	862	\$267,875.23	\$310.76	-\$8.59	4.61%	\$4,728.76	1.80%	-		
	DECEMBER	946	\$288,801.98	\$305.29	-\$5.47	9.74%	\$20,926.75	7.81%	-		

sum		10,335	\$3,252,087.76								

iv) Output 2 (For years 2022 & 2023 and USA only):

Enter the year(s) for analysis (comma-separated, e.g., 2022, 2023):

Year(s): 2022, 2023

Enter the country name or 'ALL' to analyze all countries:

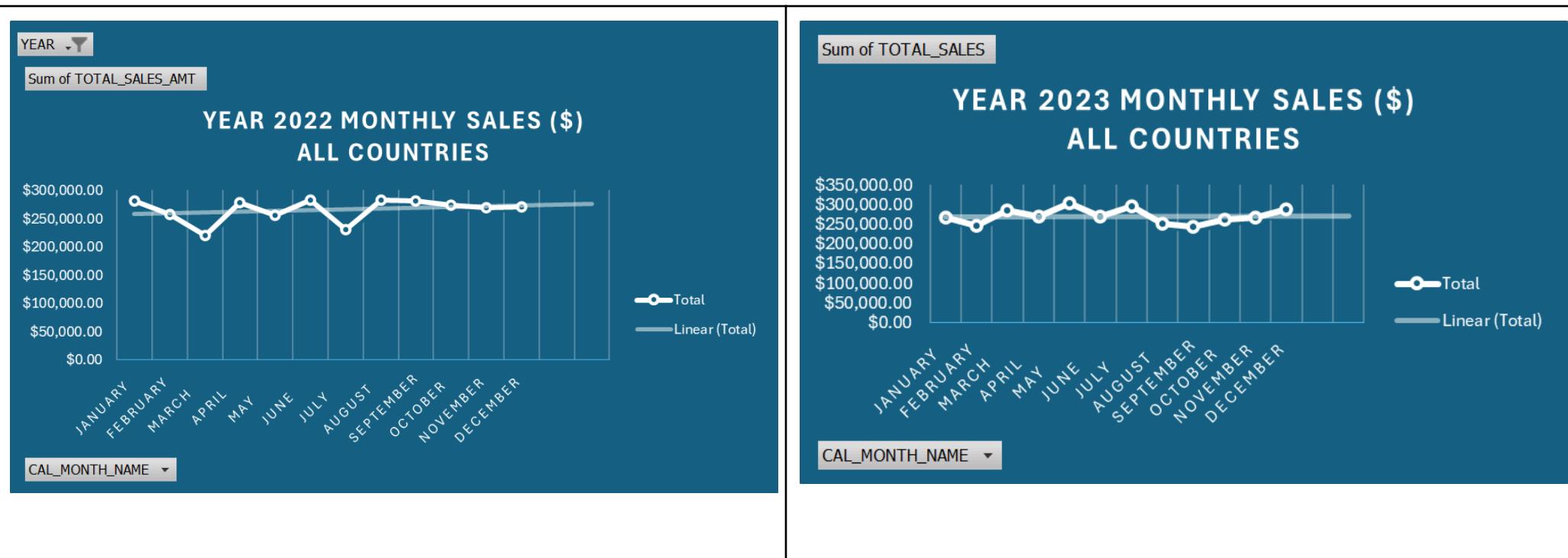
Country Name: USA

Monthly Analysis of Years 2022, 2023 for USA Country										1 / 1	
Report Date: 16-MAY-24											
Year	Month	Total Orders	Total Sales Amount	Avg Sales/Order	Avg Sales Change	Orders Change (%)	Sales Change (%)	Sales Change (%)	Highest/Lowest Sales		
2022	JANUARY	134	\$38,792.39	\$289.50					-		
	FEBRUARY	121	\$41,622.44	\$343.99	\$54.49	-9.70%	\$2,830.05	7.30%	-		
	MARCH	92	\$31,278.76	\$339.99	-\$4.00	-23.97%	-\$10,343.68	-24.85%	-		
	APRIL	126	\$38,153.38	\$302.80	-\$37.18	36.96%	\$6,874.62	21.98%	-		
	MAY	110	\$32,353.91	\$294.13	-\$8.68	-12.70%	-\$5,799.47	-15.20%	-		
	JUNE	110	\$35,663.12	\$324.21	\$30.08	0.00%	\$3,309.21	10.23%	-		
	JULY	88	\$28,993.78	\$329.47	\$5.26	-20.00%	-\$6,669.34	-18.70%	Lowest		
	AUGUST	139	\$49,456.67	\$355.80	\$26.33	57.95%	\$20,462.89	70.58%	Highest		
	SEPTEMBER	130	\$49,451.37	\$380.40	\$24.59	-6.47%	-\$5.30	-0.01%	-		
	OCTOBER	104	\$35,430.99	\$340.68	-\$39.71	-20.00%	-\$14,020.38	-28.35%	-		
	NOVEMBER	133	\$41,160.78	\$309.48	-\$31.20	27.88%	\$5,729.79	16.17%	-		
	DECEMBER	110	\$39,995.50	\$363.60	\$54.12	-17.29%	-\$1,165.28	-2.83%	-		

sum		1,397	\$462,353.09								
2023	JANUARY	101	\$31,164.88	\$308.56					-		
	FEBRUARY	107	\$39,518.83	\$369.33	\$60.77	5.94%	\$8,353.95	26.81%	-		
	MARCH	131	\$45,503.58	\$347.36	-\$21.98	22.43%	\$5,984.75	15.14%	-		
	APRIL	145	\$44,970.25	\$310.14	-\$37.22	10.69%	-\$533.33	-1.17%	-		
	MAY	135	\$48,054.71	\$355.96	\$45.82	-6.90%	\$5,084.46	6.86%	Highest		
	JUNE	122	\$37,323.23	\$305.93	-\$50.03	-9.63%	-\$10,731.48	-22.33%	-		
	JULY	123	\$35,384.74	\$287.68	-\$18.25	0.82%	-\$1,938.49	-5.19%	-		
	AUGUST	101	\$30,585.90	\$302.83	\$15.15	-17.89%	-\$4,798.84	-13.56%	Lowest		
	SEPTEMBER	111	\$38,816.39	\$349.70	\$46.87	9.90%	\$8,230.49	26.91%	-		
	OCTOBER	129	\$39,282.86	\$304.52	-\$45.18	16.22%	\$466.47	1.20%	-		
	NOVEMBER	116	\$34,885.86	\$300.74	-\$3.78	-10.08%	-\$4,397.00	-11.19%	-		
	DECEMBER	114	\$34,523.35	\$302.84	\$2.10	-1.72%	-\$362.51	-1.04%	-		

sum		1,435	\$460,014.58								

v) **Visualisation for Query 1: Time Series Analysis of Monthly Sales for Years 2022 and 2023 for All Countries**



Observation: From these time series graphs, we are able to analyze the trends in the company's sales over time. Based on the **2022** graph, we can clearly see that there is an upward trend in sales over time. Sales were the highest in June but the lowest in March. Its monthly sales remain quite constant in the second half of the year. As for the **2023** graph, its upward trend is not as obvious as the **2022** graph. Sales were the highest in May but the lowest in September. From these 2 graphs, sales tend to vary more in the first half of the year than in the second half.

3.3.2 Sales Distribution by Country and Region for 2 Given Years for Comparison (includes Ranking)

i) Purpose:

This query provides a comparative analysis of sales performance across different countries and regions for two specified years. By examining sales data from these periods, the company can identify trends, changes and areas of improvement in their sales operations. This information enables them to make strategic decisions aimed at optimizing sales strategies, refining marketing efforts and allocating resources effectively to enhance competitiveness in the market. Additionally, the company can evaluate the effectiveness of their initiatives and adjust their approach accordingly to stay ahead of competitors. The query also ranks countries and regions based on their sales figures for each year. This provides valuable insights into the relative performance of different regions over time, highlighting areas where the company may have gained or lost market share. By understanding which regions perform well consistently or show improvement over time, the company can prioritize resources and strategic initiatives to capitalise on growth opportunities and address challenges in less performing areas.

ii) SQL Code:

```

SET VERIFY OFF;
SET PAGESIZE 42;
SET LINESIZE 170;
clear columns;
clear breaks;
clear computes;

DEFINE page_count = 2

ACCEPT year1 PROMPT 'Enter the first year: '
ACCEPT year2 PROMPT 'Enter the second year: '

REPFOOTER SKIP 3 '                                     ( END OF
REPORT )'
TTITLE
'=====
===== SKIP 1-
'          Analysis of Sales by Country and Region for Years
&year1 and &year2      ' SQL.PNO " / &page_count" SKIP 1-
'=====
===== -
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'=====
===== -
BTITLE
'=====
===== ' SKIP 1-
'          End of Page' FORMAT 9
SQL.PNO SKIP 1-
```

```

=====
=====' SKIP 2-

COLUMN country FORMAT A20 HEADING "Country";
COLUMN region FORMAT A20 HEADING "Region";
COLUMN sales_&year1 FORMAT $999,999.99 HEADING "Sales &year1";
COLUMN sales_&year2 FORMAT $999,999.99 HEADING "Sales &year2";
COLUMN diff_actual FORMAT $999,999.99 HEADING "Actual Diff";
COLUMN diff_percentage FORMAT 999.99 HEADING "Sales Diff (%)" FORMAT A15;
COLUMN rank_&year1 FORMAT 999 HEADING "Rank &year1";
COLUMN rank_&year2 FORMAT 999 HEADING "Rank &year2";
BREAK ON country SKIP 1 ON report;

SELECT
    country,
    NVL(region, 'Unknown') AS region,
    TO_CHAR("&year1", '$999,999.99') AS sales_&year1,
    TO_CHAR("&year2", '$999,999.99') AS sales_&year2,
    TO_CHAR("&year2" - "&year1", '$999,999.99') AS diff_actual,
    CASE
        WHEN ((&year2 - &year1) / "&year1") * 100 < 1 AND ((&year2 -
&year1) / "&year1") * 100 > -1 THEN TO_CHAR(((&year2 - &year1) /
&year1) * 100, '990D00') || '%'
        ELSE TO_CHAR(((&year2 - &year1) / "&year1") * 100, '999.99') || '%'
    END AS diff_percentage,
    RANK() OVER (ORDER BY "&year1" DESC) AS rank_&year1,
    RANK() OVER (ORDER BY "&year2" DESC) AS rank_&year2
FROM (
    SELECT
        d.cal_year,
        trim(c.country) as country,
        trim(c.region) as region,
        SUM(orf.line_total) AS total_sales_amt
    FROM
        order_fact orf
    JOIN
        date_dim d ON orf.date_key = d.date_key
    JOIN
        customer_dim c ON orf.customer_key = c.customer_key
    WHERE
        d.cal_year IN (&year1, &year2)
    GROUP BY
        d.cal_year,
        trim(c.country),
        trim(c.region)
)
PIVOT (
    SUM(total_sales_amt)
    FOR cal_year IN ('&year1' AS "&year1", '&year2' AS "&year2")

```

```
)
ORDER BY
    country,
    region;
```

```
SET VERIFY ON;
clear columns;
clear breaks;
clear computes;
```

iii) Output 1 (For Years 2022 & 2023):

```
Enter the first year: 2022
Enter the second year: 2023
```

Analysis of Sales by Country and Region for Years 2022 and 2023							1 / 2	
Country	Region	Sales 2022	Sales 2023	Actual Diff	Sales Diff (%)	Rank 2022	Rank 2023	
Argentina	Unknown	\$100,424.24	\$121,977.87	\$21,553.63	21.46%	10	8	
Austria	Unknown	\$72,818.50	\$65,850.02	-\$6,968.48	-9.57%	14	19	
Belgium	Unknown	\$66,811.61	\$76,604.46	\$9,792.85	14.66%	18	13	
Brazil	RJ	\$94,707.30	\$114,188.25	\$19,480.95	20.57%	11	10	
	SP	\$217,034.37	\$219,597.18	\$2,562.81	1.18%	4	4	
Canada	BC	\$72,845.02	\$71,083.16	-\$1,761.86	-2.42%	13	15	
	Qu[bec]	\$39,652.52	\$34,623.41	-\$5,029.11	-12.68%	24	25	
Denmark	Unknown	\$77,609.94	\$73,182.23	-\$4,427.71	-5.71%	12	14	
Finland	Unknown	\$59,772.68	\$67,178.94	\$7,406.26	12.39%	19	18	
France	Unknown	\$362,498.23	\$389,494.62	\$26,996.39	7.45%	2	1	
Germany	Unknown	\$390,790.12	\$372,461.14	-\$18,328.98	-4.69%	1	2	
Ireland	Co. Cork	\$31,800.58	\$40,474.99	\$8,674.41	27.28%	33	22	
Italy	Unknown	\$114,135.29	\$89,633.46	-\$24,501.83	-21.47%	8	11	
Mexico	Unknown	\$174,463.31	\$177,438.27	\$2,974.96	1.71%	5	5	
Norway	Unknown	\$35,731.29	\$40,174.19	\$4,442.90	12.43%	28	24	
Poland	Unknown	\$43,211.92	\$34,561.31	-\$8,650.61	-20.02%	21	26	

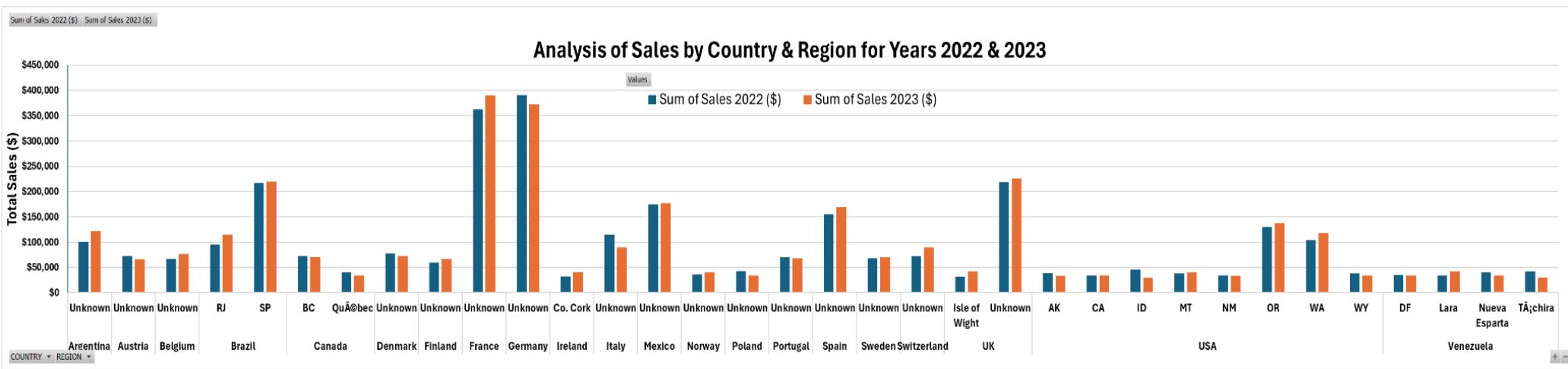
End of Page 1

Analysis of Sales by Country and Region for Years 2022 and 2023							2 / 2	
Report Date: 14-MAY-24								
Country	Region	Sales 2022	Sales 2023	Actual Diff	Sales Diff (%)	Rank 2022	Rank 2023	
Portugal	Unknown	\$69,829.55	\$67,622.53	-\$2,207.02	-3.16%	16	17	
Spain	Unknown	\$154,454.42	\$169,256.29	\$14,801.87	9.58%	6	6	
Sweden	Unknown	\$68,377.30	\$70,241.81	\$1,864.51	2.73%	17	16	
Switzerland	Unknown	\$71,415.03	\$89,400.85	\$17,985.82	25.18%	15	12	
UK	Isle of Wight	\$30,947.79	\$41,451.97	\$10,504.18	33.94%	34	21	
	Unknown	\$218,882.23	\$225,037.50	\$6,155.27	2.81%	3	3	
USA	AK	\$39,285.20	\$32,972.60	-\$6,312.60	-16.07%	25	32	
	CA	\$33,960.18	\$34,186.22	\$226.04	0.67%	32	28	
	ID	\$45,550.96	\$29,607.24	-\$15,943.72	-35.00%	20	34	
	MT	\$37,909.48	\$40,379.61	\$2,470.13	6.52%	26	23	
	NM	\$34,344.90	\$33,359.47	-\$985.43	-2.87%	30	31	
	OR	\$129,740.81	\$137,177.86	\$7,437.05	5.73%	7	7	
	WA	\$103,833.66	\$118,513.25	\$14,679.59	14.14%	9	9	
	WY	\$37,727.90	\$33,818.33	-\$3,909.57	-10.36%	27	30	
Venezuela	DF	\$35,072.90	\$34,373.09	-\$699.81	-2.00%	29	27	
	Lara	\$34,118.01	\$41,754.34	\$7,636.33	22.38%	31	20	
	Nueva Esparta	\$39,685.25	\$34,004.43	-\$5,680.82	-14.31%	23	29	
	Táchira	\$42,172.19	\$30,406.87	-\$11,765.32	-27.90%	22	33	

(END OF REPORT)

End of Page 2

iv) Visualisation for Query 2: Analysis of Sales by Country & Region for Years 2022 & 2023



Observation: This bar chart allows us to identify the best performing regions from each country and compare the performance, in terms of sales generated, of each region in each country between the years 2022 and 2023. Several regions performed approximately the same between the 2 years, for example SP in Brazil, Mexican Region and Swedish Region. However, there are other regions where their sales performance either improved or deteriorated. For example. France's sales improved from the Year 2022 to 2023 while Germany's sales dropped from the Year 2022 to 2023. In 2022, the German region contributed the most sales while in 2023, France contributed the most sales.

3.3.3 Top 5 Products in each Country for 2 Given Years for comparison (includes their quarterly & yearly sales)

i) Purpose:

The purpose of this query is to generate a report that highlights the top 5 products for each country for two specified years. It calculates the sales amount for each quarter (Q1, Q2, Q3, Q4) as well as the total sales amount for these products. By analyzing sales data across different quarters and countries, the company can identify which products perform the best in each country and how their sales trend over time. This information can be valuable for strategic decision-making, such as inventory management, marketing efforts, and product development, to maximize sales and profitability.

ii) SQL Code:

```

SET VERIFY OFF;
SET PAGESIZE 68;
SET LINESIZE 400;
clear columns;
clear breaks;
clear computes;

DEFINE page_count = 5

ACCEPT year1 PROMPT 'Enter the first year: '
ACCEPT year2 PROMPT 'Enter the second year: '

REPFOOTER SKIP 5 '
( END OF REPORT )'
TTITLE
'=====
===== SKIP
1-
'
          Top 5 Products for each
Country for Years &year1 and &year2           ' SQL.PNO " /
&page_count" SKIP 1-
'=====
===== -
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'=====
===== -
BTITLE
'=====
===== ' SKIP
1-
'
          End of
Page' FORMAT 9 SQL.PNO SKIP 1-
'=====
===== ' SKIP
2-

```

```

COLUMN country FORMAT A20 HEADING 'Country';
COLUMN cal_year FORMAT A5 HEADING "Year";
COLUMN productname FORMAT A40 HEADING 'Product Name';
COLUMN q1_sales_amt FORMAT $999,990.99 HEADING 'Q1 Sales Amount';
COLUMN q2_sales_amt FORMAT $999,990.99 HEADING 'Q2 Sales Amount';
COLUMN q3_sales_amt FORMAT $999,990.99 HEADING 'Q3 Sales Amount';
COLUMN q4_sales_amt FORMAT $999,990.99 HEADING 'Q4 Sales Amount';
COLUMN total_sales_amt FORMAT $999,999.99 HEADING 'Total Sales Amount';
COLUMN product_rank FORMAT 999 HEADING 'Rank';
BREAK ON country SKIP 1 ON cal_year SKIP 1;

WITH quarterly_sales AS (
  SELECT
    c.country,
    p.productname,
    d.cal_year,
    SUM(CASE WHEN d.cal_quarter = 'Q1' THEN orf.line_total ELSE 0.00 END)
  AS q1_sales_amt,
    SUM(CASE WHEN d.cal_quarter = 'Q2' THEN orf.line_total ELSE 0.00 END)
  AS q2_sales_amt,
    SUM(CASE WHEN d.cal_quarter = 'Q3' THEN orf.line_total ELSE 0.00 END)
  AS q3_sales_amt,
    SUM(CASE WHEN d.cal_quarter = 'Q4' THEN orf.line_total ELSE 0.00 END)
  AS q4_sales_amt,
    SUM(orf.line_total) AS total_sales_amt,
    RANK() OVER (PARTITION BY c.country, d.cal_year ORDER BY
    SUM(orf.line_total) DESC) AS product_rank
  FROM
    order_fact orf
  JOIN
    date_dim d ON orf.date_key = d.date_key
  JOIN
    customer_dim c ON orf.customer_key = c.customer_key
  JOIN
    product_dim p ON orf.product_key = p.product_key
  JOIN
    (
      SELECT productid, productname
      FROM product_dim
      WHERE UPPER(currentRowIndicator) = 'Y'
    ) latest_products ON p.productid = latest_products.productid
  WHERE
    d.cal_year IN (&year1, &year2)
  GROUP BY
    c.country,
    p.productname,
    d.cal_year
)

```

```
)  
SELECT  
    country,  
    TO_CHAR(cal_year, '9999') AS cal_year,  
    productname,  
    q1_sales_amt,  
    q2_sales_amt,  
    q3_sales_amt,  
    q4_sales_amt,  
    total_sales_amt  
FROM  
    quarterly_sales  
WHERE  
    product_rank <= 5  
ORDER BY  
    country,  
    cal_year,  
    product_rank;  
  
clear columns;  
clear breaks;  
clear computes;
```

iii) Output 1 (For Years 2022 and 2023):

***Note: Since the output is long (5 pages), I will only show the 1st & 5th pages**

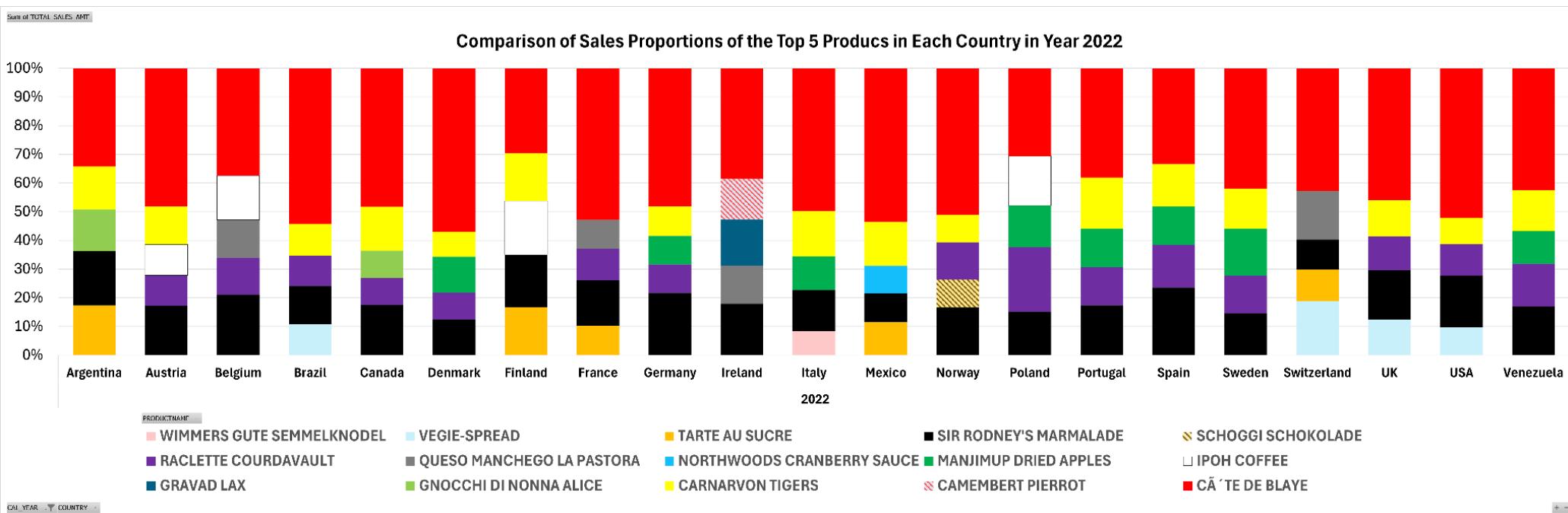
Enter the first year: 2022

Enter the second year: 2023

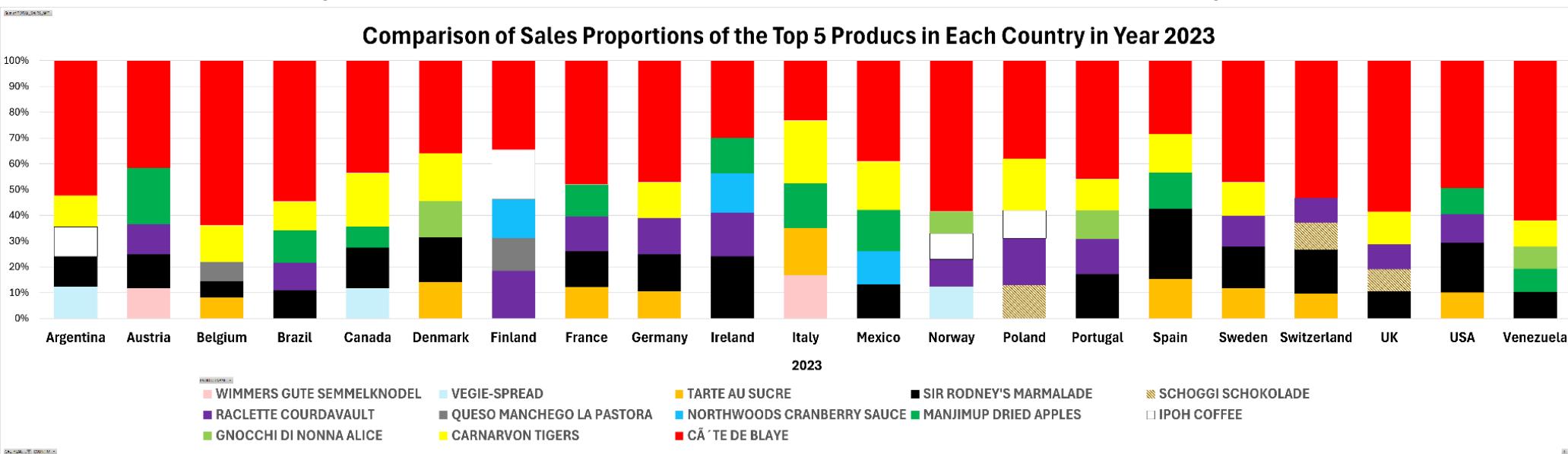
Top 5 Products for each Country for Years 2022 and 2023								1 / 5
Country	Year	Product Name	Q1 Sales Amount	Q2 Sales Amount	Q3 Sales Amount	Q4 Sales Amount	Total Sales Amount	
Argentina	2022	CHTE DE BLINE	\$927.52	\$4,421.54	\$511.19	\$3,699.54	\$9,559.79	
		SIR RODNEY'S MARMALADE	\$896.67	\$1,937.52	\$1,278.99	\$1,166.40	\$5,279.58	
		TARTE AU SUCRE	\$433.84	\$2,435.92	\$1,232.00	\$736.53	\$4,838.29	
		CARNARVON TIGERS	\$558.75	\$771.88	\$1,736.88	\$1,121.88	\$4,189.39	
		GNOCCHI DI NONNA ALICE	\$1,153.68	\$1,126.70	\$921.88	\$835.24	\$4,037.50	
	2023	CHTE DE BLINE	\$8,666.52	\$4,368.83	\$3,644.21	\$3,061.87	\$19,741.43	
		VEGIE-SPREAD	\$1,088.73	\$2,054.95	\$607.58	\$877.57	\$4,628.83	
		CARNARVON TIGERS	\$1,236.25	\$367.51	\$865.63	\$2,098.75	\$4,568.14	
		SIR RODNEY'S MARMALADE	\$1,965.06	\$0.00	\$1,161.54	\$1,306.53	\$4,433.13	
		IPOH COFFEE	\$540.96	\$957.72	\$1,738.80	\$1,094.34	\$4,331.82	
Austria	2022	CHTE DE BLINE	\$2,869.52	\$5,733.76	\$782.60	\$2,318.80	\$11,704.68	
		SIR RODNEY'S MARMALADE	\$648.00	\$1,101.60	\$1,724.49	\$688.50	\$4,162.59	
		CARNARVON TIGERS	\$706.25	\$573.13	\$536.25	\$1,389.38	\$3,205.01	
		IPOH COFFEE	\$785.68	\$1,228.20	\$372.60	\$220.80	\$2,607.28	
		RACLETTE COURDAVULT	\$544.50	\$1,449.25	\$282.70	\$319.00	\$2,595.45	
	2023	CHTE DE BLINE	\$2,450.55	\$453.22	\$0.00	\$4,966.98	\$7,870.75	
		MANJIMUP DRIED APPLES	\$1,488.24	\$1,684.87	\$608.44	\$357.22	\$4,138.77	
		SIR RODNEY'S MARMALADE	\$596.16	\$695.79	\$652.86	\$532.98	\$2,477.79	
		WIMMERS GUTE SEMMELKNODEL	\$698.92	\$601.50	\$476.15	\$445.55	\$2,222.12	
		RACLETTE COURDAVULT	\$198.00	\$396.00	\$750.20	\$870.65	\$2,214.85	
Belgium	2022	CHTE DE BLINE	\$0.00	\$4,943.26	\$801.04	\$521.73	\$6,266.03	
		SIR RODNEY'S MARMALADE	\$696.60	\$561.33	\$1,250.64	\$1,027.08	\$3,535.65	
		IPOH COFFEE	\$291.64	\$985.78	\$733.24	\$573.16	\$2,583.82	
		QUESO MANCHEGO LA PASTORA	\$176.70	\$573.80	\$488.68	\$965.96	\$2,205.14	
		RACLETTE COURDAVULT	\$1,056.55	\$464.75	\$581.35	\$46.20	\$2,148.85	
	2023	CHTE DE BLINE	\$8,424.10	\$5,886.60	\$2,677.17	\$1,873.49	\$18,861.36	
		CARNARVON TIGERS	\$729.38	\$519.38	\$1,796.88	\$1,109.38	\$4,155.02	
		TARTE AU SUCRE	\$1,362.64	\$394.40	\$366.79	\$281.01	\$2,404.84	
		QUESO MANCHEGO LA PASTORA	\$855.00	\$277.02	\$244.72	\$860.70	\$2,237.44	
		SIR RODNEY'S MARMALADE	\$813.24	\$476.28	\$0.00	\$541.08	\$1,830.60	
Brazil	2022	CHTE DE BLINE	\$10,466.22	\$16,858.73	\$9,067.06	\$11,733.66	\$48,125.67	
		SIR RODNEY'S MARMALADE	\$1,490.40	\$3,303.99	\$3,844.26	\$3,172.77	\$11,811.42	
		CARNARVON TIGERS	\$2,738.76	\$1,738.13	\$2,635.01	\$2,694.38	\$9,806.28	
		VEGIE-SPREAD	\$1,851.70	\$2,619.53	\$3,234.55	\$1,883.76	\$9,589.54	
		RACLETTE COURDAVULT	\$4,433.00	\$2,285.80	\$2,186.25	\$522.50	\$9,427.55	
	2023	CHTE DE BLINE	\$12,173.71	\$8,128.98	\$12,948.40	\$19,973.32	\$53,224.41	
		MANJIMUP DRIED APPLES	\$2,126.36	\$702.78	\$3,668.13	\$5,907.38	\$12,404.65	
		CARNARVON TIGERS	\$5,459.39	\$765.63	\$2,988.76	\$1,693.13	\$10,906.91	
		SIR RODNEY'S MARMALADE	\$1,284.66	\$2,692.44	\$2,848.77	\$3,000.15	\$10,726.02	
		RACLETTE COURDAVULT	\$1,529.00	\$2,692.25	\$3,371.50	\$2,685.65	\$10,278.40	
Canada	2022	CHTE DE BLINE	\$5,349.05	\$4,677.13	\$5,823.35	\$729.90	\$16,579.43	
		SIR RODNEY'S MARMALADE	\$805.14	\$2,531.25	\$1,356.75	\$1,302.48	\$5,995.62	
		CARNARVON TIGERS	\$2,476.25	\$541.88	\$1,048.13	\$1,175.00	\$5,241.26	
		GNOCCHI DI NONNA ALICE	\$494.76	\$971.28	\$1,091.74	\$738.34	\$3,296.12	

Top 5 Products for each Country for Years 2022 and 2023							5 / 5
Report Date: 14-MAY-24							
Country	Year	Product Name	Q1 Sales Amount	Q2 Sales Amount	Q3 Sales Amount	Q4 Sales Amount	Total Sales Amount
Switzerland	2022	TARTE AU SUCRE	\$341.65	\$690.69	\$915.99	\$195.23	\$2,143.56
		SIR RODNEY'S MARMALADE	\$1,056.24	\$214.65	\$613.98	\$142.56	\$2,027.43
		CHTE DE BLYE	\$2,463.73	\$6,321.37	\$5,773.29	\$1,678.50	\$16,236.89
		SIR RODNEY'S MARMALADE	\$672.30	\$2,666.52	\$714.42	\$1,123.47	\$5,176.71
		SCHOGGI SCHOKOLADE	\$1,074.23	\$978.53	\$106.68	\$1,032.96	\$3,192.40
	2023	TARTE AU SUCRE	\$405.24	\$1,095.94	\$0.00	\$1,431.66	\$2,932.84
		RACLETTE COURDAVOUT	\$385.55	\$1,054.35	\$685.30	\$775.50	\$2,900.70
		CHTE DE BLYE	\$7,968.25	\$3,393.89	\$8,263.38	\$10,479.40	\$30,104.92
		SIR RODNEY'S MARMALADE	\$2,166.75	\$5,090.85	\$2,340.09	\$1,614.33	\$11,212.02
		CARNARVON TIGERS	\$2,255.00	\$1,030.63	\$2,156.27	\$2,832.50	\$8,274.40
UK	2022	VEGIE-SPREAD	\$2,768.33	\$2,726.20	\$1,304.28	\$1,360.90	\$8,159.71
		RACLETTE COURDAVOUT	\$2,178.55	\$1,825.45	\$888.25	\$2,934.25	\$7,826.50
		CHTE DE BLYE	\$16,371.27	\$13,283.04	\$10,874.66	\$8,845.70	\$49,374.67
		CARNARVON TIGERS	\$1,335.01	\$2,495.02	\$4,121.26	\$2,611.90	\$10,563.19
		SIR RODNEY'S MARMALADE	\$2,738.61	\$2,201.58	\$2,890.08	\$982.53	\$8,812.80
	2023	RACLETTE COURDAVOUT	\$2,648.80	\$2,482.15	\$1,305.70	\$1,770.45	\$8,207.10
		SCHOGGI SCHOKOLADE	\$2,064.62	\$1,710.35	\$1,665.57	\$1,772.24	\$7,212.78
		CHTE DE BLYE	\$16,247.43	\$11,773.19	\$25,630.67	\$20,929.83	\$74,581.12
		SIR RODNEY'S MARMALADE	\$6,394.14	\$6,383.61	\$7,609.95	\$5,378.40	\$25,766.10
		RACLETTE COURDAVOUT	\$4,115.10	\$3,166.90	\$4,978.60	\$3,451.80	\$15,712.40
USA	2022	VEGIE-SPREAD	\$3,494.88	\$3,734.57	\$3,654.25	\$3,090.55	\$13,974.25
		CARNARVON TIGERS	\$3,440.00	\$2,195.63	\$3,958.15	\$3,600.64	\$13,194.42
		CHTE DE BLYE	\$21,269.75	\$15,630.86	\$15,491.18	\$11,717.86	\$64,109.65
		SIR RODNEY'S MARMALADE	\$6,970.86	\$8,132.40	\$4,862.43	\$5,031.72	\$24,997.41
		RACLETTE COURDAVOUT	\$3,746.60	\$2,768.70	\$3,828.55	\$4,160.20	\$14,584.05
	2023	TARTE AU SUCRE	\$1,199.47	\$3,689.62	\$4,372.42	\$3,839.01	\$13,100.52
		MANJIMUP DRIED APPLES	\$1,393.90	\$5,488.68	\$2,478.81	\$3,694.63	\$13,056.02
		CHTE DE BLYE	\$7,233.08	\$6,028.88	\$5,952.48	\$1,256.90	\$20,471.34
		SIR RODNEY'S MARMALADE	\$2,165.94	\$301.32	\$2,821.23	\$2,941.11	\$8,229.60
		RACLETTE COURDAVOUT	\$0.00	\$2,456.30	\$2,620.20	\$2,060.85	\$7,137.35
Venezuela	2022	CARNARVON TIGERS	\$2,189.38	\$1,388.13	\$1,922.51	\$1,371.25	\$6,871.27
		MANJIMUP DRIED APPLES	\$1,774.97	\$1,606.96	\$1,058.94	\$1,084.91	\$5,525.78
		CHTE DE BLYE	\$3,654.75	\$4,273.97	\$9,480.73	\$9,251.49	\$26,660.94
		SIR RODNEY'S MARMALADE	\$1,702.62	\$1,135.62	\$794.61	\$805.95	\$4,438.80
		CARNARVON TIGERS	\$624.38	\$1,806.26	\$550.00	\$1,415.00	\$4,395.64
	2023	MANJIMUP DRIED APPLES	\$0.00	\$1,639.82	\$1,392.84	\$861.78	\$3,894.44
		GNOCCHI DI NONNA ALICE	\$580.64	\$543.40	\$2,187.66	\$310.84	\$3,622.54
		(END OF REPORT)					
		End of Page 5					

iv) Visualisation for Query 3 (part 1): Comparisons of Sales Proportions of the Top 5 Products in Each Country (Year 2022)



Observation: The stacked bar chart for Year 2022 not only displays the top 5 products in each country but also illustrates their sales proportions within each country. This visualization enables us to identify the products that contributed the most to overall sales. Each product is represented by a distinct color. The CATE DE BLAYE product emerges as one of the top 5 products across all countries, suggesting its widespread popularity. Furthermore, in several countries, CATE DE BLAYE ranks as the top-selling product. Conversely, products like Schoggi Schokolade are less popular, featuring among the top 5 products in Norway alone. Armed with this analysis, the company can make informed decisions about resource allocation, inventory management and marketing strategies.

v) Visualisation for Query 3 (part 2): Comparisons of Sales Proportions of the Top 5 Products in Each Country (Year 2023)


Observation: The stacked bar chart for Year 2023 not only displays the top 5 products in each country but also illustrates their sales proportions within each country. This visualization enables us to identify the products that contributed the most to overall sales. Each product is represented by a distinct color. The CATE DE BLAYE product emerges as one of the top 5 products across all countries, suggesting its widespread popularity. Furthermore, in several countries, CATE DE BLAYE ranks as the top-selling product. Surprisingly, the Schoggi Schokolade gained popularity over time as it appeared as one of the top 5 products in more countries, like the UK and Switzerland. Besides that, we could observe which products remained as one of the top 5 products in each country and those that didn't. For example, in 2022, the Gravad Lax product was 1 of the top five products in Ireland. However, in 2023, it no longer was. Armed with this analysis, the company can make informed decisions about resource allocation, inventory management and marketing strategies.

3.4 Yong Zee Lin

3.4.1 Yearly Shipper Performance Report

i) Purpose:

The shipper performance analysis evaluates how different shipping companies perform based on metrics like total orders, revenue, average order value, and order frequency. It helps identify top performers and areas for improvement and also guides decisions on shipping logistics to enhance efficiency and customer satisfaction.

Total Orders: The number of orders each shipper has handled.

Total Revenue: The total amount of revenue generated by each shipper.

Average Order Value: The average value of orders handled by each shipper.

ii) SQL Code:

```

SET LINESIZE 1000
DEFINE page_count = 1
SET PAGESIZE 21
clear columns

ACCEPT report_year PROMPT 'Enter report year: '

DEFINE page_count = 1
REPFOOTER '                               ( END OF REPORT ) '
TTITLE
'=====
===== SKIP 1-
'                               Shippers Performance Analysis in &report_year
SQL.PNO " / &page_count" SKIP 1-
'=====
===== -
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'=====
===== -
BTITLE
'=====
===== SKIP 1-
'                               End of Page' FORMAT 9 SQL.PNO SKIP 1-
'=====
===== -
COLUMN Shipper_Name FORMAT A25 HEADING "Shipper Name"
COLUMN Total_Orders FORMAT 999,999 HEADING "Total Orders"
COLUMN Total_Revenue FORMAT $99,999,999.99 HEADING "Total Revenue ($)"
COLUMN Average_Order_Value FORMAT $999,999.99 HEADING "Average_Order_Value ($)"

```

```
CREATE OR REPLACE VIEW Shipper_Performance_Analysis AS
SELECT
    Shipper_Name,
    Total_Orders,
    Total_Revenue,
    Average_Order_Value
FROM (
    SELECT
        f.shipperName AS Shipper_Name,
        Count(f.quantity) AS Total_Orders,
        SUM(f.line_total) AS Total_Revenue,
        SUM(f.line_total) / Count(f.quantity) AS Average_Order_Value
    FROM
        order_fact f
    JOIN
        Date_Dim d ON f.date_key = d.date_key
    WHERE
        d.cal_year = &report_year
    GROUP BY
        f.shipperName
    UNION ALL
    SELECT
        'Total' AS Shipper_Name,
        Count(f.quantity) AS Total_Orders,
        SUM(f.line_total) AS Total_Revenue,
        SUM(f.line_total) / Count(f.quantity) AS Average_Order_Value
    FROM
        order_fact f
    JOIN
        Date_Dim d ON f.date_key = d.date_key
    WHERE
        d.cal_year = &report_year
);
select * from Shipper_Performance_Analysis;
```

```
TTITLE OFF
```

```
BTITLE OFF
```

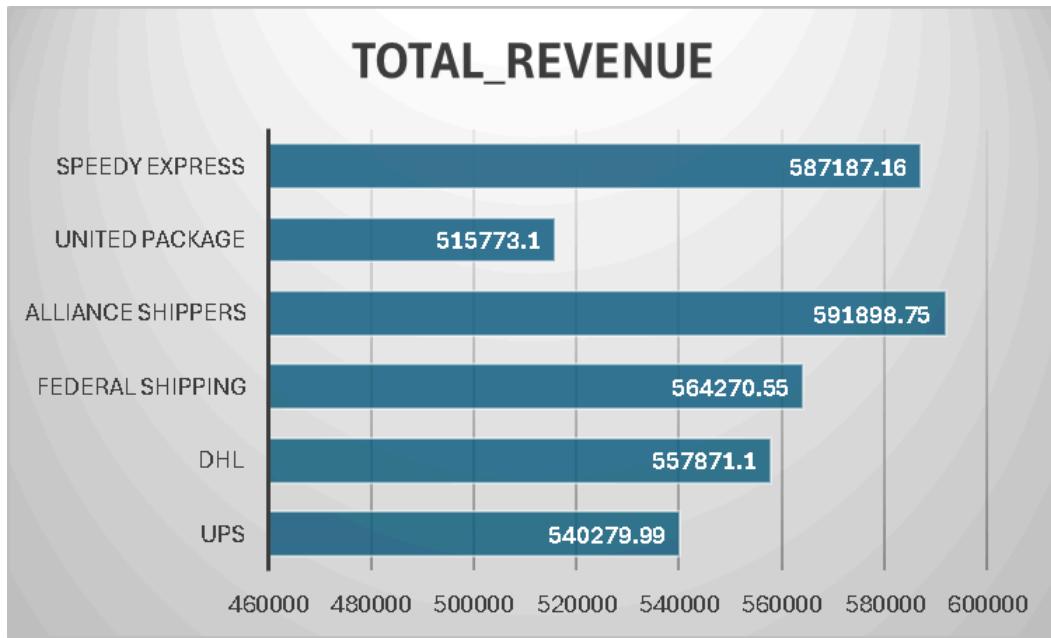
```
clear columns
```

iii) Output:

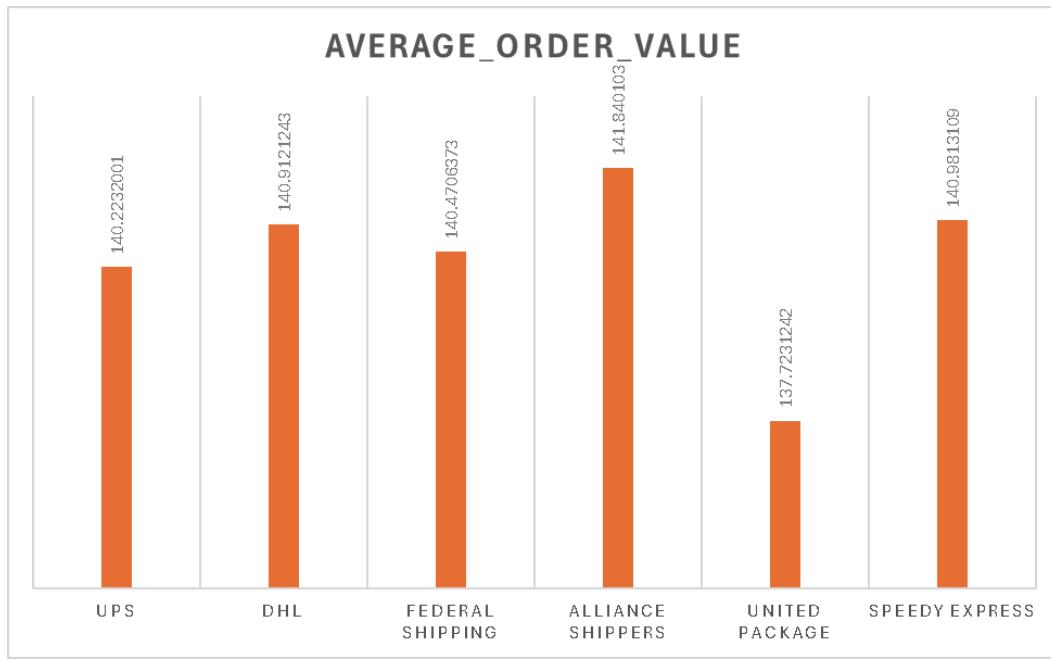
```
Enter report year: 2023
old 18:      d.cal_year = &report_year
new 18:      d.cal_year = 2023
old 32:      d.cal_year = &report_year
new 32:      d.cal_year = 2023
```

View created.

=====			
Shippers Performance Analysis in 2023			1 / 1
Report Date: 16-MAY-24			
Shipper Name	Total Orders	Total Revenue (\$)	Average_Order_Value (\$)
UPS	3,853	\$540,279.99	\$140.22
DHL	3,959	\$557,871.10	\$140.91
Federal Shipping	4,017	\$564,270.55	\$140.47
Alliance Shippers	4,173	\$591,898.75	\$141.84
United Package	3,745	\$515,773.10	\$137.72
Speedy Express	4,165	\$587,187.16	\$140.98
Total	23,912	\$3,357,280.65	\$140.40
(END OF REPORT)			
=====			
End of Page 1			
=====			

iv) Visualization for Query 1:

Observation: Based on the chart above, we can see the total revenue generated by various shipping companies. Among them, Alliance Shippers had generated a total revenue of \$591,898.75. Following closely is DHL with \$557,871.10. Federal Shipping accruing a total revenue of \$564,270.55. Speedy Express follows with \$587,187.16 in total revenues. UPS stands at \$540,279.99 in total revenue. Next, United Package trails have the total revenue with \$515,773.10. This analysis highlights Alliance Shippers' dominance in revenue generation and is closely followed by DHL and Federal Shipping. Lastly, United Package exhibiting comparatively lower revenue among the listed companies.



Observation: Based on the chart of the average order value between the shippers, we can see that Alliance Shippers has the highest average order value at 141.84. After that, Speedy Express followed closely behind Alliance Shippers at 140.98. United Package has the lowest average order value at 137.72. Next, UPS, DHL, and Federal Shipping fall within a narrow range between 140.22 and 140.91. This analysis suggests that while some companies prioritize higher-value orders, others maintain a consistent average order value. Understanding

these differences can inform strategic decisions regarding pricing, customer targeting, and service offerings within the shipping industry.

3.4.2 Top 10 Customer Lifetime Value in 2023

i) Purpose

This query is finding the top 10 customers with the highest Customer Lifetime Value (CLV). It is calculated based on the total orders made, total amount spent, and the duration of the customer's relationship with the company. By ranking customers according to their CLV, businesses can prioritize their efforts towards retaining and nurturing these valuable relationships. This analysis aids in strategic decision-making, such as resource allocation, marketing strategies, and customer retention initiatives. This is ultimately aiming to maximize long-term profitability and sustainable growth for the company.

ii) SQL Code:

```

SET LINESIZE 1000
DEFINE page_count = 1
SET PAGESIZE 24
clear columns

DEFINE page_count = 1
REPFOOTER '                               ( END OF REPORT ) '
TTITLE
=====
=====' SKIP 1-
'
          Top 10 Customer Lifetime Value in 2023
' SQL.PNO " / &page_count" SKIP 1-
=====
=====' -
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
=====
=====' -
BTITLE
=====
=====' SKIP 1-
'
          End of Page' FORMAT 9 SQL.PNO SKIP
1-
=====
=====' SKIP 2-

COLUMN Rank FORMAT 99;
COLUMN Customer_ID FORMAT 999999 HEADING "CustomerID";
COLUMN Company_Name HEADING "Company Name" FORMAT A20;
COLUMN City FORMAT A15 HEADING "City";
COLUMN Country FORMAT A15 HEADING "Country";
COLUMN Total_Orders FORMAT 999,999 HEADING "Total Orders";
COLUMN Total_Spent FORMAT $999,999.99 HEADING "Total Spent ($)";

```

```
COLUMN CLV FORMAT $999,999.99 HEADING "CLV ($)" ;

CREATE OR REPLACE VIEW Top10CLV AS
SELECT
    RANK() OVER (ORDER BY CLV DESC) AS Rank,
    Customer_ID,
    Company_Name,
    City,
    Country,
    Total_Orders,
    Total_Spent,
    CLV
FROM (
    SELECT
        cd.cust_no AS Customer_ID,
        cd.comp_name AS Company_Name,
        cd.city AS City,
        cd.country AS Country,
        COUNT(DISTINCT F.orderid) AS Total_Orders,
        SUM(F.line_total) AS Total_Spent,
        SUM(F.line_total) / 9 AS CLV,
        ROW_NUMBER() OVER (ORDER BY SUM(F.line_total) DESC) AS rank
    FROM
        Order_Fact F
    JOIN
        customer_dim cd ON F.customer_key = cd.customer_key
    JOIN
        Date_dim d ON F.date_key = d.date_key
    WHERE
        d.cal_year = 2023
    GROUP BY
        cd.cust_no, cd.comp_name, cd.city, cd.country
)
WHERE
    rank <= 10
ORDER BY
    CLV DESC;

select * from Top10CLV;

TTITLE OFF
BTITLE OFF
clear columns
```

iii) Output:

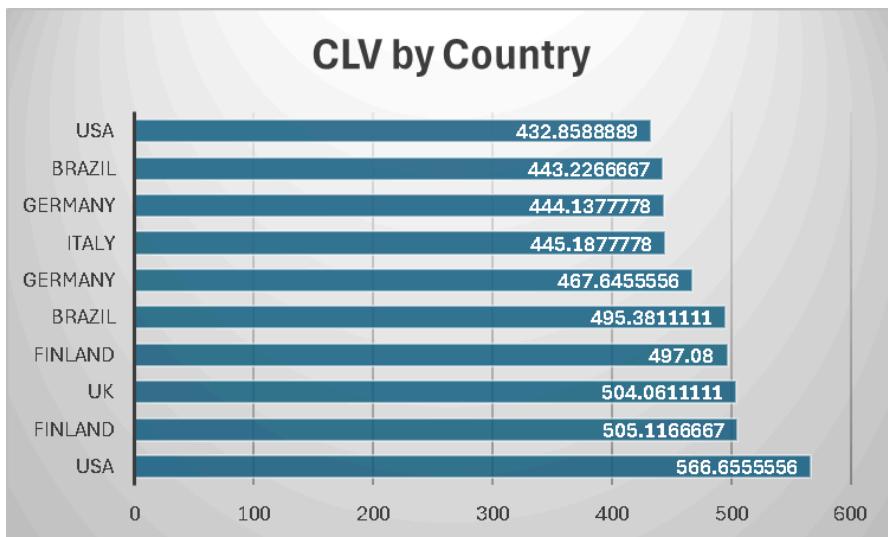
Top 10 Customer Lifetime Value in 2023							1 / 1
RANK	CustomerID	Company Name	City	Country	Total Orders	Total Spent (\$)	CLV (\$)
1	100218	Hungry Coyote Import	Elgin	USA	2	\$5,099.90	\$566.66
2	105547	Wartian Herkku	Oulu	Finland	5	\$4,546.05	\$505.12
3	102749	Eastern Connection	London	UK	3	\$4,536.55	\$504.06
4	101637	Wilman Kala	Helsinki	Finland	4	\$4,473.72	\$497.08
5	105491	Gourmet Lanchonetes	Campinas	Brazil	4	\$4,458.43	\$495.38
6	103787	Ottilies Kaseladen	Koln	Germany	4	\$4,208.81	\$467.65
7	100959	Magazzini Alimentari	Bergamo	Italy	5	\$4,006.69	\$445.19
8	105681	Koniglich Essen	Brandenburg	Germany	4	\$3,997.24	\$444.14
9	102363	Wellington Importado	Resende	Brazil	3	\$3,989.04	\$443.23
10	102637	White Clover Markets	Seattle	USA	4	\$3,895.73	\$432.86

(END OF REPORT)

End of Page 1

iv) Visualization for Query 2:

Observation: I create a chart of rankings from 1 to 10 and corresponding customer lifetime values. First, the customer ranked 1rd has a CLV of 566.66, while the customer ranked 10th has a CLV of 432.86. This indicates that the top-ranked customers are likely to be more valuable to the business over their lifetime compared to lower-ranked ones. Understanding the relationship between rank and CLV can help businesses prioritize their resources and focus on retaining and nurturing their most valuable customers.



Observation: This chart provides a breakdown of Customer Lifetime Values (CLVs) by company name. It reveals the proportion each company contributes to the total CLV. The USA country has the highest CLV which is 432.86, followed by Brazil with 443.23. This insight helps businesses prioritize resources and tailor strategies to maximize returns from high-CLV customers. It also highlights opportunities for targeted marketing and improved customer service to enhance loyalty and drive growth. Understanding CLV by company name enables businesses to segment their customer base effectively and tailor strategies to meet the unique needs of different customer groups. This can ultimately lead to higher satisfaction and profitability.

3.4.3 Product Category Sales in 2023

i) Purpose

This report calculates various metrics such as Total Orders, Total Revenue, Percentage of Orders, and Percentage of Revenue for each product category. Additionally, it identifies the most ordered and least ordered products within each category.

ii) SQL Code:

```

SET LINESIZE 1200
DEFINE page_count = 1
SET PAGESIZE 24
clear columns

DEFINE page_count = 1
REPFOOTER '
( END OF REPORT )'
TTITLE
'=====
=====
=====
====='
SKIP 1-
'
Product Category
' SQL.PNO " /
Sales in 2023
&page_count" SKIP 1-

```

```
'=====
'=====-
SKIP 1 LEFT 'Report Date: ' _Date SKIP 1 -
'=====
'=====-
BTITLE
'=====
'=====-
SKIP 1-
'
of Page' FORMAT 9 SQL.PNO SKIP 1-
'=====
'=====-
SKIP 2-
COLUMN Product_Category FORMAT A16 HEADING "Product Category";
COLUMN Total_Orders FORMAT 999,999 HEADING "Total Order";
COLUMN Total_Sales FORMAT $999,999,999.99 HEADING "Total Sales($)";
COLUMN Percentage_Of_Orders FORMAT 999.99 HEADING "Order Percentage(%)";
COLUMN Percentage_Of_Sales FORMAT 999.99 HEADING "Sales Percentage(%)";
COLUMN Most_Ordered_Product FORMAT A31 HEADING "Most Ordered Product";
COLUMN Least_Ordered_Product FORMAT A32 HEADING "Least Ordered Product";

CREATE OR REPLACE VIEW Product_Category_Sales AS
SELECT
    a.Product_Category,
    a.Total_Orders,
    a.Total_Sales,
    a.Percentage_Of_Orders,
    a.Percentage_Of_Sales,
    most_ordered.Most_Ordered_Product,
    least_ordered.Least_Ordered_Product
FROM (
    SELECT
        p.categoryName AS Product_Category,
        Count(f.quantity) AS Total_Orders,
        SUM(f.line_total) AS Total_Sales,
        ROUND((Count(f.quantity) / SUM(Count(f.quantity)) OVER () * 100, 2) AS Percentage_Of_Orders,
        ROUND((SUM(f.line_total) / SUM(SUM(f.line_total)) OVER () * 100, 2) AS Percentage_Of_Sales
    FROM
        order_fact f
    JOIN
        Product_dim p ON f.product_key = p.product_key
    JOIN
```

```

        Date_Dim d ON f.date_key = d.date_key
    WHERE
        d.cal_year = 2023
    GROUP BY
        p.categoryName
) a
JOIN (
    SELECT
        p.categoryName AS Product_Category,
        p.productName AS Most_Ordered_Product,
        RANK() OVER (PARTITION BY p.categoryName ORDER BY Count(f.quantity)
DESC) AS Most_Ordered_Rank
    FROM
        order_fact f
    JOIN
        Product_dim p ON f.product_key = p.product_key
    JOIN
        Date_Dim d ON f.date_key = d.date_key
    WHERE
        d.cal_year = 2023
    GROUP BY
        p.categoryName, p.productName
) most_ordered ON a.Product_Category = most_ordered.Product_Category AND
most_ordered.Most_Ordered_Rank = 1
JOIN (
    SELECT
        p.categoryName AS Product_Category,
        p.productName AS Least_Ordered_Product,
        RANK() OVER (PARTITION BY p.categoryName ORDER BY Count(f.quantity)) AS
Least_Ordered_Rank
    FROM
        order_fact f
    JOIN
        Product_dim p ON f.product_key = p.product_key
    JOIN
        Date_Dim d ON f.date_key = d.date_key
    WHERE
        d.cal_year = 2023
    GROUP BY
        p.categoryName, p.productName
) least_ordered ON a.Product_Category = least_ordered.Product_Category AND
least_ordered.Least_Ordered_Rank = 1;

select * from Product_Category_Sales;

TTITLE OFF
BTITLE OFF
clear columns

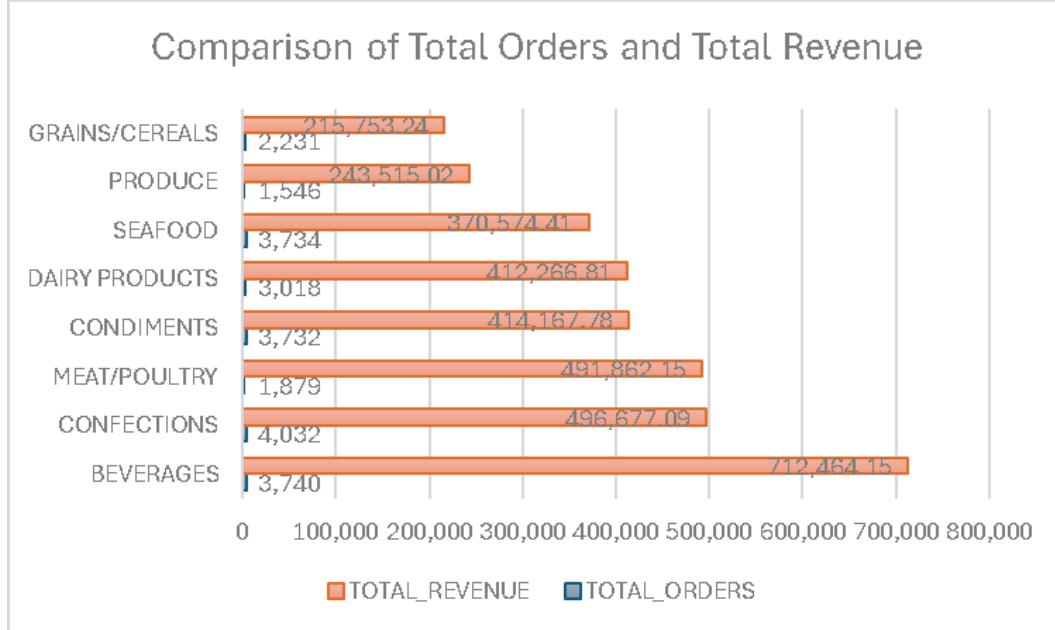
```

iii) Output

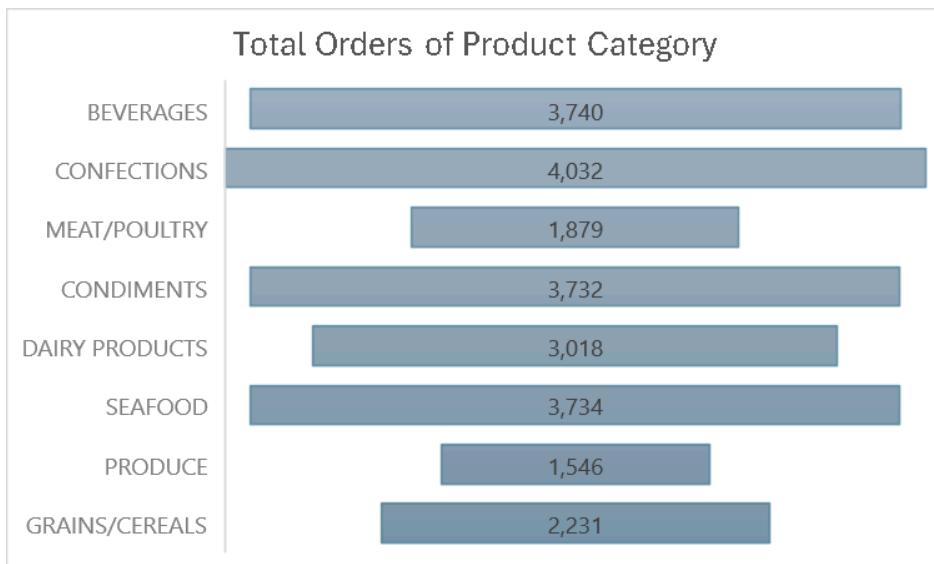
Product Category Sales in 2023						1 / 1
Report Date: 16-MAY-24						
Product Category	Total Order	Total Sales(\$)	Order Percentage(%)	Sales Percentage(%)	Most Ordered Product	Least Ordered Product
BEVERAGES	3,740	\$712,464.15	15.64	21.22	CÔTE DE BLAYE	CHARTREUSE VERTE
CONDIMENTS	3,732	\$414,167.78	15.61	12.34	GULA MALACCA	LOUISIANA FIERY HOT PEPPER SAUCE
CONFETIONS	4,032	\$496,677.09	16.86	14.79	NUNUCA NUB-NOUGAT-CRÈME	VALKOINEN SUKLA
DAIRY PRODUCTS	3,018	\$412,266.81	12.62	12.28	GORGONZOLA TELINO	GUDBRANDSDALSOFT
GRAINS/CEREALS	2,231	\$215,753.24	9.33	6.43	TUNNBROD	RAVIOLI ANGELO
MEAT/POULTRY	1,879	\$491,862.15	7.86	14.65	PERTH PASTIES	TOURTIERE
PRODUCE	1,546	\$243,515.02	6.47	7.25	UNCLE BOB'S ORGANIC DRIED PEARS	MANJIMUP DRIED APPLES
SEAFOOD	3,734	\$370,574.41	15.62	11.04	NORD-OST MATJESHERING	JACK'S NEW ENGLAND CLAM CHOWDER
(END OF REPORT)						

End of Page 1

iv) Visualization for Query 3:



Observation: Comparing orders and revenue reveals key insights into product category sales. Beverages comprise 3,740 of orders but it generates a higher revenue share at 212,464.15. In contrast, confetions account for 4,032 of orders but it contributes 496,677.09 to revenue. This indicates potentially lower transaction values. Meat/poultry have 1,879 orders and yield significant revenue at 491,862.15. It hints at a higher value per unit. The alignment between orders and revenue varies across categories, reflecting diverse pricing strategies and customer preferences.



Observation: Based on this chart, we can see that confections lead with the highest total orders. It indicates a strong preference for sweet treats. Staple items like Meat/Poultry and Dairy Products follow closely behind. Meanwhile, Produce registers the lowest orders. This is because it is influenced by factors like seasonality and perishability. Beverages, Condiments, and Seafood occupy a middle ground in terms of popularity. Analyzing these trends can inform strategies for inventory management and marketing to better cater to consumer preferences.