RWANDA POLYTECHNIC
NGOMA COLLEGE
P.O. Box 35 KIBUNGO - RWANDA
Tel: +250 785 – 883 - 746
Email: info@iprcngoma.rp.ac.rw
www.iprcngoma.rp.ac.rw

Module Detail		Trainee's Detail	
SECTOR:	ICT	Reg No:	1.23RP00987 2. 24RP15724
SUB-SECTOR:	Information Technology	Class:	Level 8 Information Technology
		Trainer's Detail	
CERTIFICATE:	Bachelors of Technology	Name:	Eng. NYIRIMANA J.M Vianney
MODULE (Code &Title):	ITLDM801 – DATA MINING AND DATA WAREHOUSE	Additional info	
Competence:	Apply Data Mining and Warehousing	Duration:	
		Due date:	1st April, 2025
Training Centre:	RP Ngoma College	Signature:	
Scored marks:		Decision:	Competent
			Not Yet Competent

Store Books ETL and Data Warehouse Project Report (PRACTICAL CAT)

Project Objectives

This project aims to transform the existing Store Books Sales database into a robust data warehouse, enabling advanced analytics and reporting capabilities. The project involves the use of Extract, Transform, Load (ETL) processes to migrate and transform data, and the implementation of SQL Server Analysis Services (SSAS) to create a cube for in-depth data analysis. This project involves the use of Microsoft SQL Server as a database type and SQL Server management Studio(SSMS) as a technology stack, Jupyter as python editor and power BI as a reporting tool.

Key terms

Database: it is a collection of related data that use OLTP instead of OLAP

Data warehouse: it is a storage of historical data from different sources

ETL: Extract Transform Load

SSMS: (SQL Server Management Studio) is Microsoft's free integrated environment used for Writing SQL queries, managing database.

SSAS: (SQL Server Analysis Services) is an online analytical processing (OLAP) and data mining tool in the Microsoft SQL Server ecosystem. Used to building cubes/tabular models for analytics

Primary Key (PK): A column (or combination of columns) that uniquely identifies each row in a table.

Foreign Key (FK): A column that creates a relationship between two tables by referencing the primary key of another table.

Business Key (BK/Natural Key): A real-world identifier from source systems that uniquely identifies an entity in business terms.

Surrogate Key (SK): An artificial/system-generated key (usually numeric) used as the primary key in dimension tables.

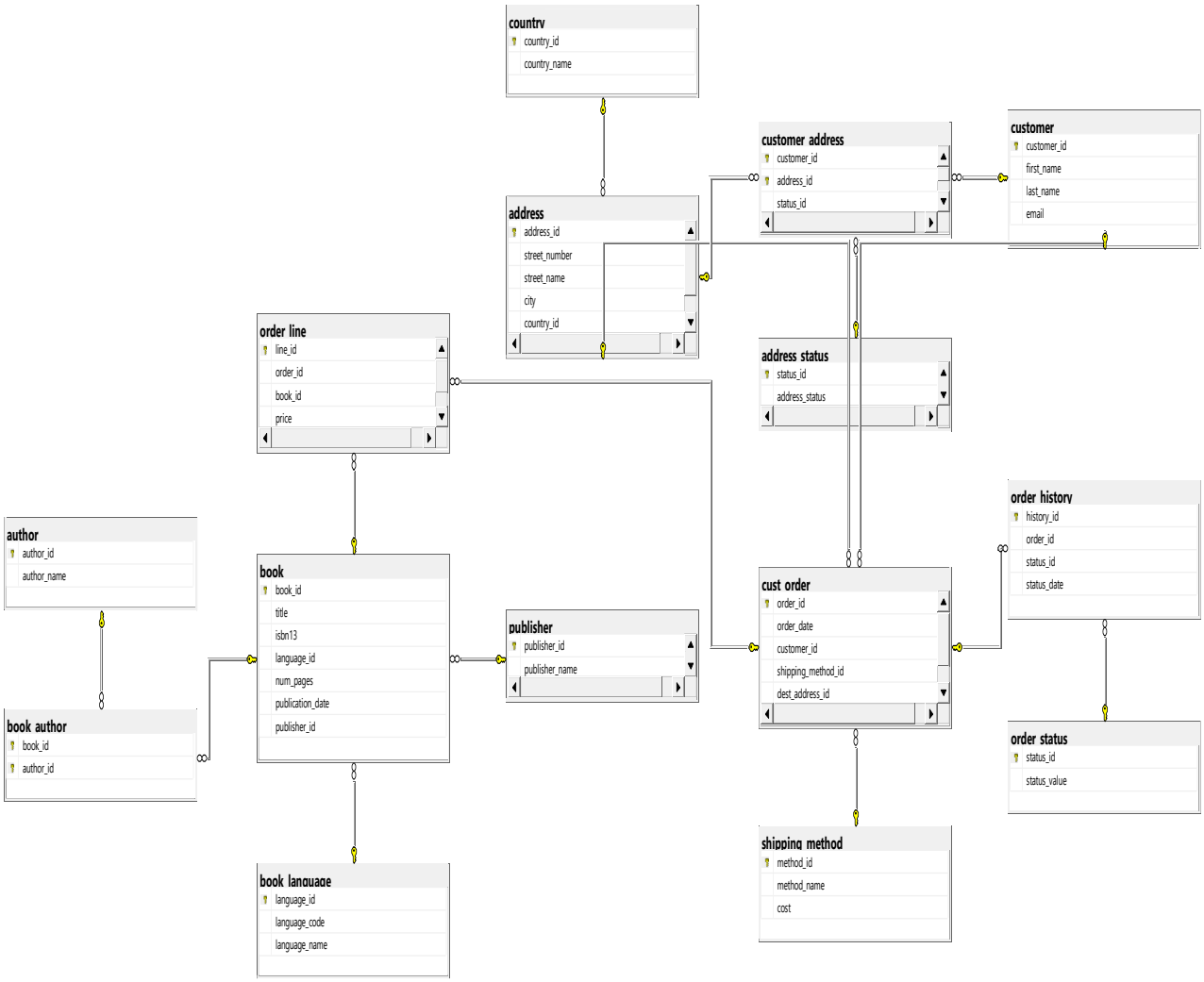
Source Database Structure

The source database contains several key tables with the following relationships:

- **author**: Contains author information (author_id, author_name)
- **publisher**: Contains publisher information (publisher_id, publisher_name)
- **book_language**: Contains language information (language_id, language_code, language_name)
- **book**: Contains book details with foreign keys to language and publisher

- **book_author**: Junction table establishing many-to-many relationship between books and authors

The following below is the overview of the Store Books Sales source database schema structure which shows us an illustration of tables, relationships, and key entities

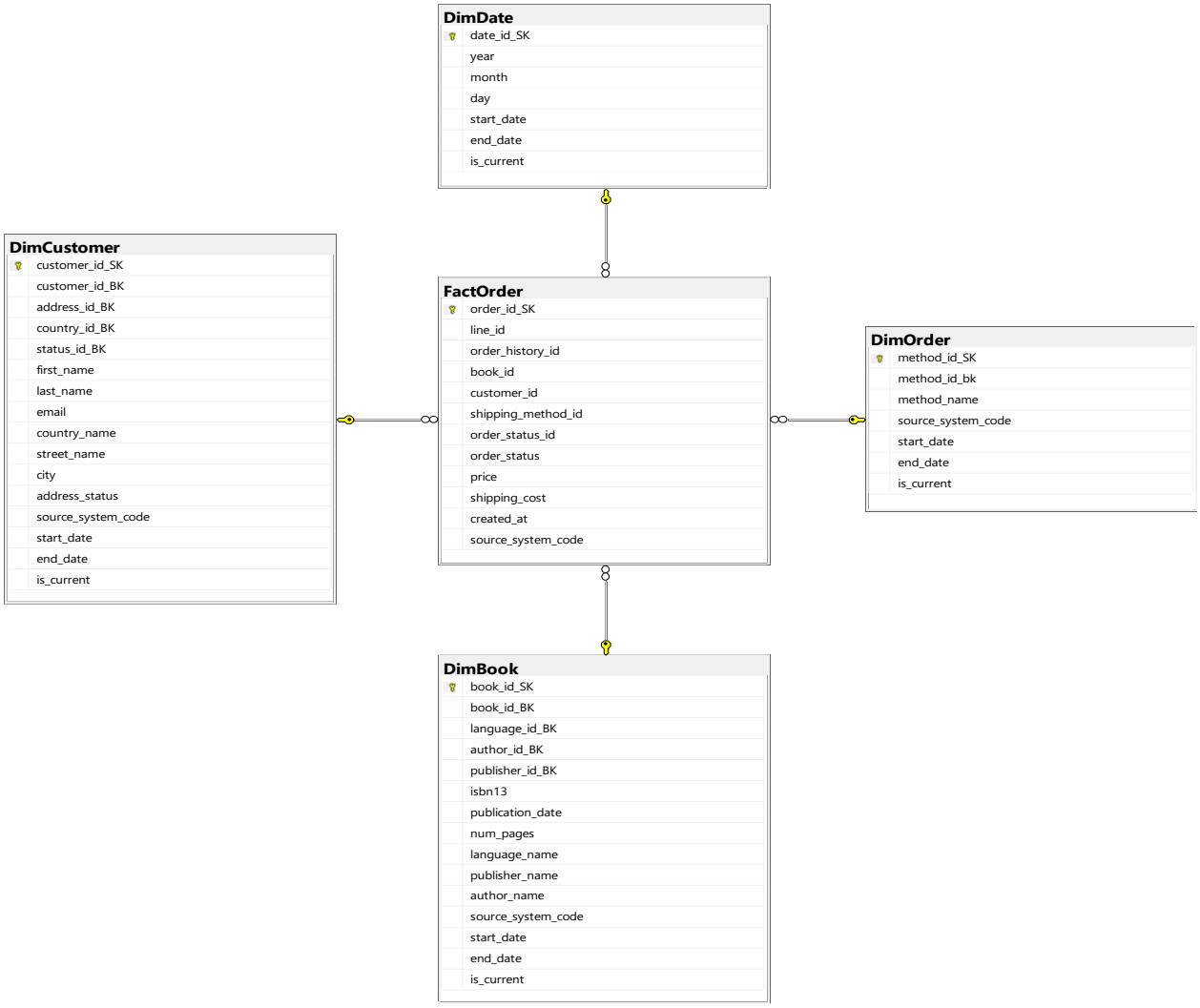


Data Warehouse Design

The data warehouse follows a **star schema** design with the following Dimension Tables:

1. **DimOrder**: Stores order method information
2. **DimCustomer**: Contains customer details with address information
3. **DimBook**: Comprehensive book information including author, publisher, and language
4. **DimDate**: Standard date dimension for temporal analysis

Fact Table: FactOrder: Central fact table containing order details with foreign keys to all dimensions



Implementation Approach

1. **ETL Process**: Using SQL Server Integration Services (SSIS) to extract, transform and load data

Installation of Libraries

! pip install sqlalchemy pyodbc

Import Necessary Libraries

import pyodbc

import pandas as pd

from datetime import datetime

Database Connection

conn = pyodbc.connect("DRIVER={ODBC Driver 17 for SQL Server};SERVER=23RP00987;DATABASE=Store_books;Trusted_Connection=yes;")

cursor = conn.cursor()

SAMPLE DATA EXTRATION

AUTHER TABLE

cursor.execute("SELECT * FROM author")

rows = cursor.fetchall()

data = [list(row) for row in rows]

author_column_names = [column[0] for column in cursor.description]

Define your own column names

author_column_names = ['author_id', 'author_name']

Create a Pandas DataFrame with your custom column names

authorData = pd.DataFrame(data, columns=author_column_names)

authorData

OUTPUT

	author_id	author_name
0	1	A. Bartlett Giamatti
1	2	A. Elizabeth Delany
2	3	A. Merritt
3	4	A. Roger Merrill
4	5	A. Walton Litz

SAMPLE DATA TRANSFORMATION

HANDLING NULL VALUES

author table

```
[75]: authorData.isnull().sum()

[75]: author_id      0
      author_name    0
      dtype: int64

[76]: authorData.fillna("Unknown", inplace=True)
```

DimBook TABLE

RENAMING TABLE COLUMNS

current_date = datetime.now()

New_BookData = New_BookData.rename(columns={'book_id':'book_id_BK','language_id':'language_id_BK'})

```
Newbook_authorData =
Newbook_authorData.rename(columns={'author_id':'author_id_BK'})
New_BookData2 =
New_BookData2.rename(columns={'publisher_id':'publisher_id_BK','isbn13':'isbn13','public
ation_date':'publication_date','num_pages':'num_pages'})
DimBookData = pd.concat([New_BookData,
                 Newbook_authorData,
                 New_BookData2,
                 Newbook_languageData,
                 NewpublisherData,
                 NewauthorData
                 ],
                axis=1, join='inner')
DimBookData['source_system_code'] = 'gravity_books'
DimBookData['start_date'] = current_date
DimBookData['end_date'] = pd.to_datetime('2025-12-31')
DimBookData['is_current'] = 1
DimBookData
```

book_id_BK	language_id_BK	author_id_BK	publisher_id_BK	isbn13	publication_date	num_pages	language_name	publisher_name	author_name	source_sy
1	2	2778	1010	8987059752	1996-09-01	276	English	10/18	A. Bartlett Giamatti	gra
2	1	5049	1967	20049130001	2004-10-04	352	United States English	1st Book Library	A. Elizabeth Delany	gra
3	1	4902	1967	23755004321	2003-03-11	128	French	1st World Library	A. Merritt	gra

DATA TRANSFORMATION OF FactOrder

Foreign Keys from Dimention Tables

```
DimDateData_date_id_SK = DimDateData[['date_id_SK']]
DimBookData_book_id = DimBookData[['book_id_SK']]
DimCustomerData_customer_id_SK = DimCustomerData[['customer_id_SK']]
DimOrderData_method_id_SK = DimOrderData[['method_id_SK']]
```

Renaming columns

```
current_date = datetime.now()
Neworder_lineData_line_id =
Neworder_lineData_line_id.rename(columns={'order_line':'line_id'})
```

```
NewDimDateData_date_id_SK = DimDateData_date_id_SK.rename(columns={'date_id_SK':'order_history_id'})
NewDimBookData_book_id = DimBookData_book_id.rename(columns={'book_id_SK':'book_id'})
NewDimCustomerData_customer_id = DimCustomerData_customer_id_SK.rename(columns={'customer_id_SK':'customer_id'})
NewDimOrderData_method_id_SK = DimOrderData_method_id_SK.rename(columns={'method_id_SK':'shipping_method_id'})
Neworder_statusData_status_id= Neworder_statusData_status_id.rename(columns={'status_id':'order_status_id'})
Neworder_statusData_status_value = Neworder_statusData_status_value.rename(columns={'status_value':'order_status'})
Neworder_lineData_price = Neworder_lineData_price.rename(columns={'price':'price'})
Newshipping_methodData_cost = Newshipping_methodData_cost.rename(columns={'cost':'shipping_cost'})
Newcust_orderData_created_at = cust_orderData_order_date.rename(columns={'start_date':'created_at'})
FactOrderData = pd.concat([
                    Neworder_lineData_line_id,
                    NewDimDateData_date_id_SK,
                    NewDimBookData_book_id,
                    NewDimCustomerData_customer_id,
                    NewDimOrderData_method_id_SK,
                    Neworder_statusData_status_id,
                    Neworder_statusData_status_value,
                    Neworder_lineData_price,
                    Newshipping_methodData_cost,
                    Newcust_orderData_created_at
                ],
                axis=1, join='inner')
FactOrderData['source_system_code'] = 'gravity_books Database'

FactOrderData
```

OUTPUT

0]:

	line_id	order_history_id	book_id	customer_id	shipping_method_id	order_status_id	order_status	price	shipping_cost	created_at	source_system_code
0	1	1	1	1	1	1	Order Received	3.40	5.90	2025-02-07 12:48:45.310	gravity_books Database
1	2	2	2	2	2	2	Pending Delivery	12.17	8.90	2025-01-19 03:24:33.310	gravity_books Database
2	3	3	3	3	3	3	Delivery In Progress	4.97	11.90	2024-09-25 22:48:36.310	gravity_books Database
3	4	4	4	4	4	4	Delivered	0.57	24.50	2024-05-19 01:51:54.310	gravity_books Database

SAMPLE DATA LOADING

CONNECTION TO DATA WAREHOUSE

```
conn = pyodbc.connect("DRIVER={ODBC Driver 17 for SQL Server};SERVER=23RP00987;DATABASE=Store_Book_DW;Trusted_Connection=yes;")
cursor = conn.cursor()
```

LOADING DATA TO FactOrder

```
insert_query = """INSERT INTO FactOrder (
        line_id,
        order_history_id,
        book_id,
        customer_id,
        shipping_method_id,
        order_status_id,
        order_status,
        price,
        shipping_cost,
        created_at,
        source_system_code)
                VALUES
                (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"""

for index, row in FactOrderData.iterrows():
    cursor.execute(insert_query, (
        row['line_id'],
        row['order_history_id'],
        row['book_id'],
```

```
        row['customer_id'],
        row['shipping_method_id'],
        row['order_status_id'],
        row['order_status'],
        row['price'],
        row['shipping_cost'],
        row['created_at'],
        row['source_system_code'],
    ))
conn.commit()
print("Data inserted successfully into dbo.FactOrder table.")
```

Retrieving/Fetching FactOrder data

```
cursor.execute("SELECT * FROM FactOrder")
rows = cursor.fetchall()
data = [list(row) for row in rows]
DimFactOrder_column_names = [column[0] for column in cursor.description]

# Define your own column names
DimOrder_column_names = ['order_id_SK','line_id', 'order_history_id', 'book_id', 'customer_id', 'date_Id_SK', 'order_status_id','order_status','price','shipping_cost','source_system_code','created_at']
# Create a Pandas DataFrame with your custom column names
FactOrderData = pd.DataFrame(data, columns = DimFactOrder_column_names )
FactOrderData
```

	order_id_SK	line_id	order_history_id	book_id	customer_id	shipping_method_id	order_status_id	order_status	price	shipping_cost	created_at	source_system_code
0	1	1	1	1	1	1	1	Order Received	3.40	5.90	2025-02-07 12:48:45.310	gravity_books Database
1	2	2	2	2	2	2	2	Pending Delivery	12.17	8.90	2025-01-19 03:24:33.310	gravity_books Database
2	3	3	3	3	3	3	3	Delivery In Progress	4.97	11.90	2024-09-25 22:48:36.310	gravity_books Database
3	4	4	4	4	4	4	4	Delivered	0.57	24.50	2024-05-19 01:51:54.310	gravity_books Database
4	5	1	1	1	1	1	1	Order	3.40	5.90	2025-02-07	gravity_books

2. **Dimensional Modeling**: Star schema optimized for analytical queries

DimOrder

```
CREATE TABLE DimOrder (
    method_id_SK INT PRIMARY KEY IDENTITY(1,1),
    method_id_bk VARCHAR(50) UNIQUE NOT NULL,
    method_name VARCHAR(100) NOT NULL,
    source_system_code VARCHAR(50) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE DEFAULT NULL,
    is_current BIT DEFAULT 1
);
```

DimBook

```
CREATE TABLE DimBook (
    book_id_SK INT IDENTITY(1,1) PRIMARY KEY,
    book_id_BK VARCHAR(50) UNIQUE NOT NULL,
    language_id_BK VARCHAR(50) NOT NULL,
    author_id_BK VARCHAR(50) NOT NULL,
    publisher_id_BK VARCHAR(50) NOT NULL,
    isbn13 VARCHAR(13) NOT NULL,
    publication_date DATE NOT NULL,
    num_pages INT NOT NULL,
    language_name VARCHAR(100) NOT NULL,
    publisher_name VARCHAR(100) NOT NULL,
    author_name VARCHAR(100) NOT NULL,
    source_system_code VARCHAR(50) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NULL,
    is_current BIT DEFAULT 1
);
```

DimCustomer

```
CREATE TABLE DimCustomer (
    customer_id_SK INT IDENTITY(1,1) PRIMARY KEY,
    customer_id_BK VARCHAR(50) UNIQUE NOT NULL,
    address_id_BK VARCHAR(50) NOT NULL,
    country_id_BK VARCHAR(50) NOT NULL,
    status_id_BK VARCHAR(50) NOT NULL,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    email VARCHAR(255) NOT NULL,
    country_name VARCHAR(100) NOT NULL,
    street_name VARCHAR(200) NOT NULL,
    city VARCHAR(100) NOT NULL,
    address_status VARCHAR(50) NOT NULL,
    source_system_code VARCHAR(50) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NULL,
    is_current BIT DEFAULT 1
);
```

DimDate

```
CREATE TABLE DimDate (
    date_id_SK INT IDENTITY(1,1) PRIMARY KEY,
    year INT NOT NULL,
    month INT NOT NULL,
    day INT NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NULL,
    is_current BIT DEFAULT 1
);
```

FactOrder

```
CREATE TABLE FactOrder (
order_id_SK INT IDENTITY(1,1) PRIMARY KEY,
line_id VARCHAR(50) NOT NULL,
order_history_id INT NOT NULL,
book_id INT NOT NULL,
```

customer_id INT NOT NULL,

shipping_method_id INT NOT NULL,

order_status_id VARCHAR(50) NOT NULL,

order_status VARCHAR(100) NOT NULL,

price DECIMAL(10, 2) NOT NULL,

shipping_cost DECIMAL(10, 2) NOT NULL,

created_at DATETIME NOT NULL,

source_system_code VARCHAR(50) NOT NULL,

FOREIGN KEY (shipping_method_id) REFERENCES DimOrder (method_id_SK),

FOREIGN KEY (book_id) REFERENCES DimBook (book_id_SK),

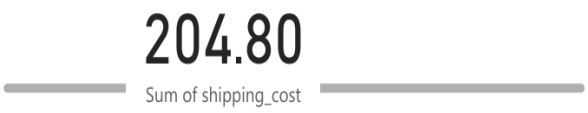
FOREIGN KEY (customer_id) REFERENCES DimCustomer (customer_id_SK),

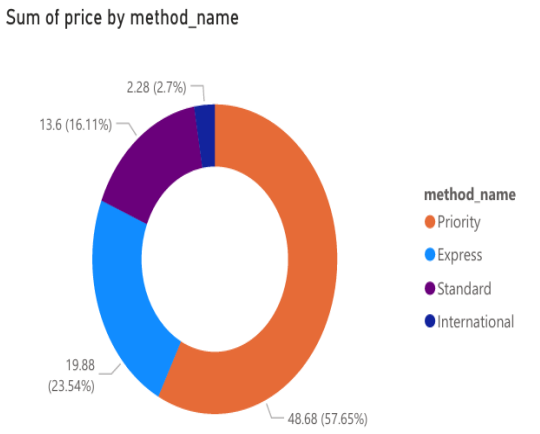
FOREIGN KEY (order_history_id) REFERENCES DimDate (date_id_SK)

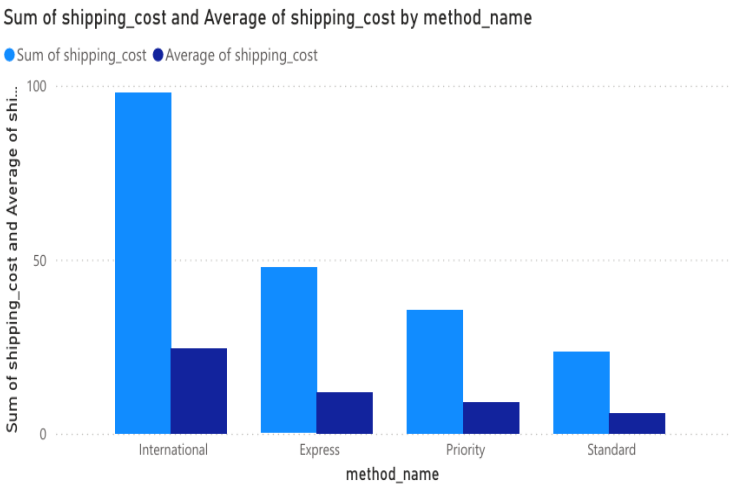
);

Store Books Dashboard by using Power BI desktop Reporting Tool

first_name	last_name	email	isbn13	Sum of price	Average of price
Ruthanne	Vatini	rvatini1@fema.gov	20049130001	48.68	12.17
Reidar	Turbitt	rturbitt2@geocities.jp	23755004321	19.88	4.97
Ursola	Purdy	upurdy0@cdbaby.com	8987059752	13.60	3.40
Rich	Kirsz	rkirsz3@jalbum.net	34406054602	2.28	0.57
Total				**84.44**	**5.28**







FINAL QUERIES AND THEIR OUTPUTS

1.SELECT

```
   title,authors, isbn13, publisher_name, COUNT(*) AS sales
FROM (
   SELECT
      b.title,
      STUFF((
         SELECT ', ' + a2.author_name
         FROM author a2
         INNER JOIN book_author ba2 ON a2.author_id = ba2.author_id
         WHERE ba2.book_id = b.book_id
         FOR XML PATH('')
      ), 1, 2, '') AS authors,
      b.isbn13,
      p.publisher_name,
      ol.line_id
   FROM order_line ol
   INNER JOIN book b ON ol.book_id = b.book_id
   INNER JOIN publisher p ON b.publisher_id = p.publisher_id
   GROUP BY b.title, b.isbn13, p.publisher_name, ol.line_id, b.book_id
) sub
GROUP BY title, authors, isbn13, publisher_name
ORDER BY COUNT(*) DESC
OFFSET 0 ROWS FETCH NEXT 20 ROWS ONLY;
```

Results Messages

	title	authors	isbn13	publisher_name	sales
1	The Only Dance There Is	Ram Dass, Richard Alpert	9780385084130	Anchor	5
2	Jojo's Bizarre Adventure Tome 9: Ruée vers la fal...	Hirohiko Araki	9782290319369	J'ai Lu	5
3	The Unfortunate Miss Fortunes	Anne Stuart, Eileen Dreyer, Jennifer Crusie	9780312940980	St. Martin's Paperbacks	5
4	In the Land of Time: And Other Fantasy Tales	Lord Dunsany, S.T. Joshi	9780142437766	Penguin Classics	5
5	Through the Arc of the Rain Forest	Karen Tei Yamashita	9780918273826	Coffee House Press	5
6	The Medium is the Massage	Jerome Agel, Marshall McLuhan, Quentin Fiore	9781584230700	Gingko Press	5
7	Tamsin	Peter S. Beagle	9780451457639	Firebird	5
8	Chimera	John Barth	9780618131709	Mariner Books	5
9	The Raven Prince (Princes Trilogy #1)	Elizabeth Hoyt	9780446618472	Warner Forever	4
10	To Green Angel Tower Part 2 (Memory Sorrow a...	Tad Williams	9780886776060	DAW Fantasy	4
11	The Great Gatsby	Alexander Scourby, Kathleen Parkinson, Matthew ...	9780060098919	Caedmon	4

Conclusion

This project transformed the Store Books Sales database into a powerful analytics solution by implementing a star schema data warehouse with a central FactOrder table and key dimension tables. Through efficient ETL processes, data was cleansed and optimized for analysis, providing valuable insights via interactive Power BI dashboards. the future improvements could include inventory metrics and automated data refreshes. Overall, the platform enhances bookstore operations, customer experiences, and revenue growth by delivering actionable business intelligence.