RWANDA POLYTECHNIC

HUYE COLLEGE

MODULE NAME: DATA MINING AND WAREHOUSING.

DEPARTMENT: ICT

OPTION: IT

LEVEL: 8 YEAR 4

CLASS: IT B-tech

LIBRARY USAGE DATA WAREHOUSE SYSTEM

GROUP 5: MEMBERS

NAMES	REG NO	MARKS
MUKESHIMANA Eric	25RP21640	
GUMYUSENGE Florent	25RP20156	
TUYISHIMIRE Rosine	25RP20978	

Date: 04th Feb 2026

Contents

List of figures

1 Introduction

The University Library Analytics project aims to solve the problem of fragmented and inconsistent library data by building a centralized Data Warehouse. Currently, data from book lending, digital resource usage, and study room bookings exist in separate systems, making reporting slow, manual, and error prone.

This project will integrate data from all sources, clean and standardize it through ETL processes, and organize it using a dimensional data model. The system will support OLAP analysis and interactive dashboards, allowing the library to track usage trends, identify peak times, compare digital and physical resource use, and analyze student and department behavior.

2 Problem Statement

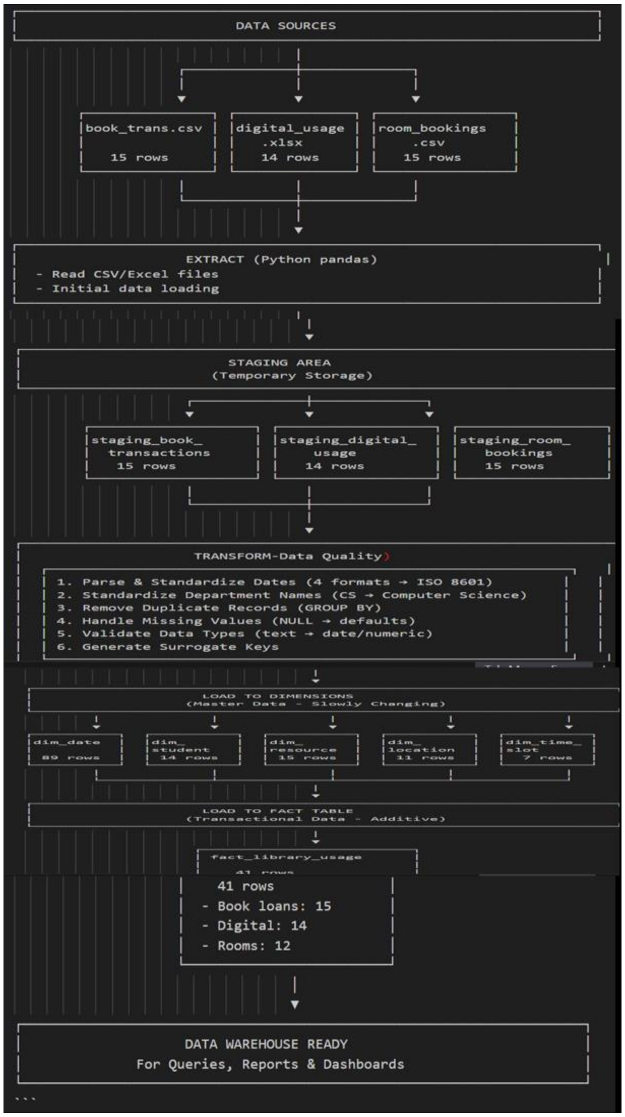
Currently, the Library Director manually combines data from these systems each month, which takes over many hours and results in inconsistent, error-prone reports. The library lacks a unified system for data analysis, historical trends, real-time dashboards, and reliable decision-making.

3 Project Objective

- The goal is to build a centralized Data Warehouse ("University Library Analytics") that: Consolidates data from all three sources
- Cleans and standardizes inconsistent data using ETL processes
- Organizes data using a dimensional model (Star or Snowflake schema)
- Supports OLAP operations (drill-down, roll-up, slicing, dicing)
- Provides interactive dashboards for decision-makers
- Ensures data security, quality, backup, and scalability

4 Team Member 1 Task

Design the Data Warehouse Architecture: This responsibility involves planning and building the overall structure of the data warehouse to ensure data is organized, integrated, and easy to.



DATA SOURCES

book_trans.csv
15 rows

digital_usage
.xlsx
14 rows

room_bookings
.csv
15 rows

EXTRACT (Python pandas)
- Read CSV/Excel files
- Initial data loading

STAGING AREA
(Temporary Storage)

staging_book_
transactions
15 rows

staging_digital_
usage
14 rows

staging_room_
bookings
15 rows

TRANSFORM-Data Quality)

1. Parse & Standardize Dates (4 formats → ISO 8601)
2. Standardize Department Names (CS → Computer Science)
3. Remove Duplicate Records (GROUP BY)
4. Handle Missing Values (NULL → defaults)
5. Validate Data Types (text → date/numeric)
6. Generate Surrogate Keys

LOAD TO DIMENSIONS
(Master Data - Slowly Changing)

dim_date
89 rows

dim_
student
14 rows

dim_
resource
15 rows

dim_
location
11 rows

dim_time_
slot
7 rows

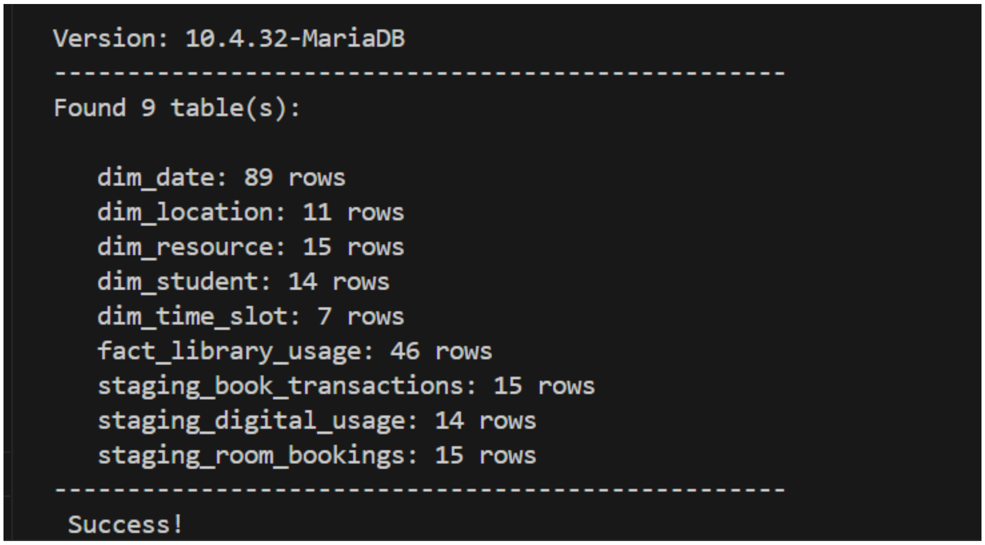
LOAD TO FACT TABLE
(Transactional Data - Additive)

fact_library_usage
41 rows
- Book loans: 15
- Digital: 14
- Rooms: 12

DATA WAREHOUSE READY
For Queries, Reports & Dashboards

```

*Figure 1:Data Warehouse Architecture*
```

Creates fact and dimension tables



```
Version: 10.4.32-MariaDB
--------------------------------------------------
Found 9 table(s):

   dim_date: 89 rows
   dim_location: 11 rows
   dim_resource: 15 rows
   dim_student: 14 rows
   dim_time_slot: 7 rows
   fact_library_usage: 46 rows
   staging_book_transactions: 15 rows
   staging_digital_usage: 14 rows
   staging_room_bookings: 15 rows
--------------------------------------------------
 Success!
```

Figure 2:fact and dimension tables

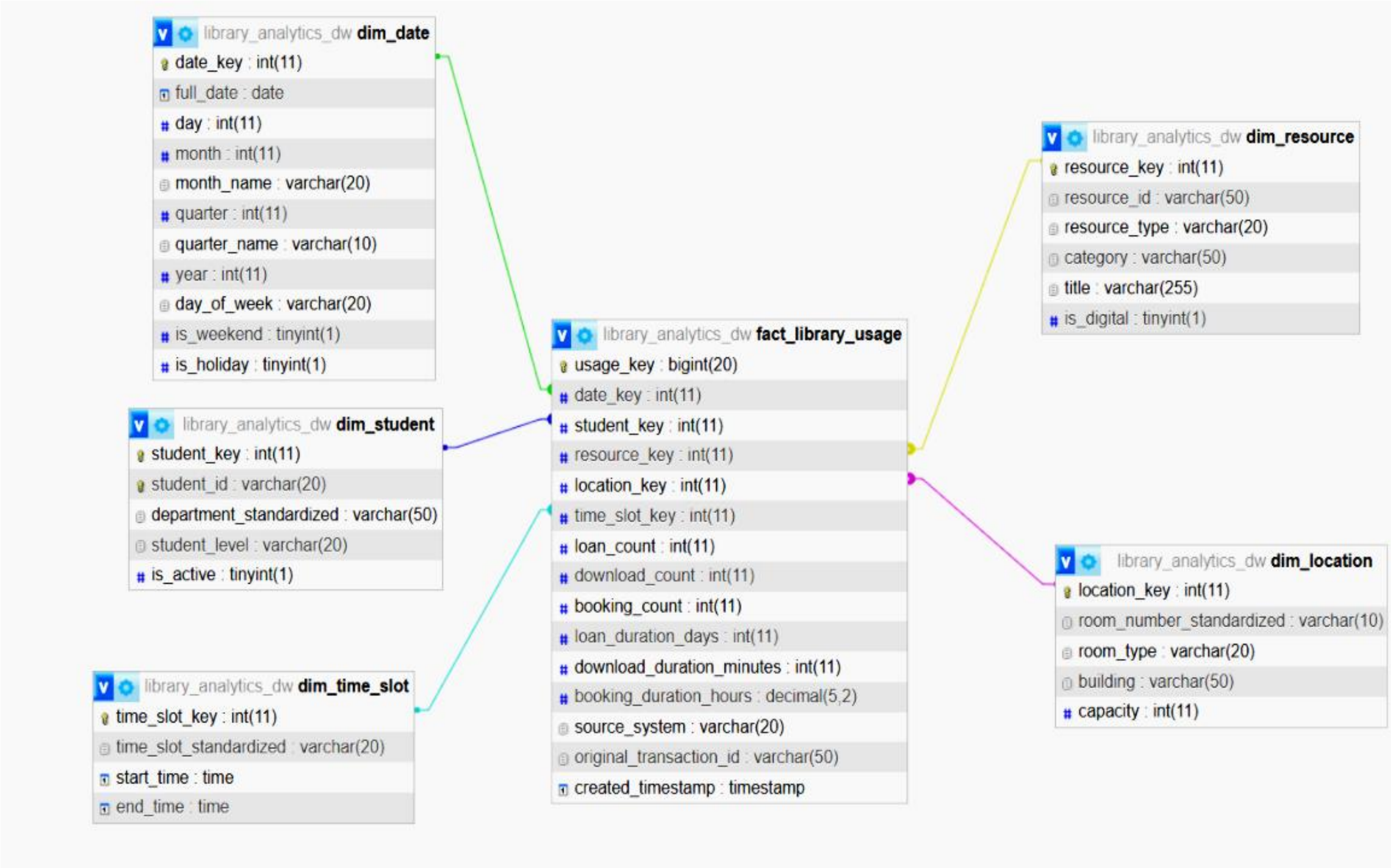
The **fact table** stores measurable library activities such as book loans, digital downloads, and room bookings, and links to all dimension tables for analysis.

The **dimension tables** (Date, Student, Resource, and Location) provide descriptive context like time, department, resource type, and room details.

Together, they support efficient reporting, trend analysis, and OLAP operations such as drill-down, roll-up, slicing, and dicing.

Produces ERD diagrams and SQL scripts

Produces **ERD diagrams** to visually show relationships between fact and dimension tables in the data warehouse.



library_analytics_dw **dim_date**
date_key : int(11)
full_date : date
day : int(11)
month : int(11)
month_name : varchar(20)
quarter : int(11)
quarter_name : varchar(10)
year : int(11)
day_of_week : varchar(20)
is_weekend : tinyint(1)
is_holiday : tinyint(1)

library_analytics_dw **dim_student**
student_key : int(11)
student_id : varchar(20)
department_standardized : varchar(50)
student_level : varchar(20)
is_active : tinyint(1)

library_analytics_dw **dim_time_slot**
time_slot_key : int(11)
time_slot_standardized : varchar(20)
start_time : time
end_time : time

library_analytics_dw **fact_library_usage**
usage_key : bigint(20)
date_key : int(11)
student_key : int(11)
resource_key : int(11)
location_key : int(11)
time_slot_key : int(11)
loan_count : int(11)
download_count : int(11)
booking_count : int(11)
loan_duration_days : int(11)
download_duration_minutes : int(11)
booking_duration_hours : decimal(5,2)
source_system : varchar(20)
original_transaction_id : varchar(50)
created_timestamp : timestamp

library_analytics_dw **dim_resource**
resource_key : int(11)
resource_id : varchar(50)
resource_type : varchar(20)
category : varchar(50)
title : varchar(255)
is_digital : tinyint(1)

library_analytics_dw **dim_location**
location_key : int(11)
room_number_standardized : varchar(10)
room_type : varchar(20)
building : varchar(50)
capacity : int(11)

Figure 3: ERD diagrams

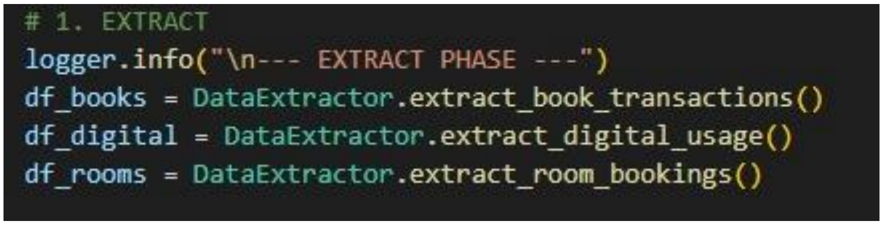
5 Team Member 2 – ETL Specialist

Team Member 2 was responsible for the ETL (Extract, Transform, Load) process and data quality management in the University Library Analytics Data Warehouse project.

The primary mission was to extract data from heterogeneous sources, clean and standardize it, and load it into staging and warehouse tables to ensure reliable reporting

5.1 Extracts (ETL – Extract Phase)

“Extracts” refers to the process of **collecting data from all source systems and bringing it into a staging area** before transformation and loading into the data warehouse.

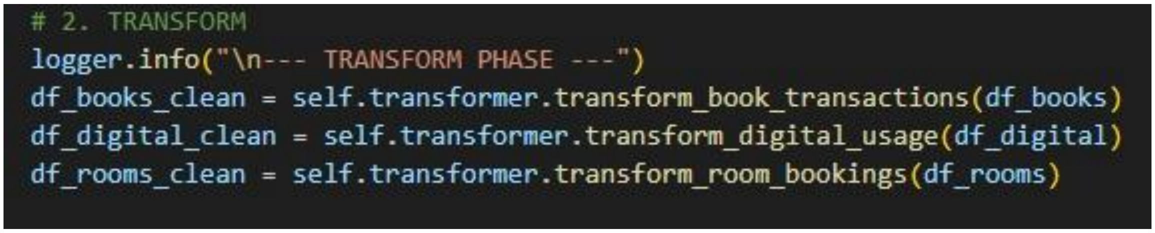


```
# 1. EXTRACT
logger.info("\n--- EXTRACT PHASE ---")
df_books = DataExtractor.extract_book_transactions()
df_digital = DataExtractor.extract_digital_usage()
df_rooms = DataExtractor.extract_room_bookings()
```

Figure 4:Extract

5.2 Transforms (ETL – Transform Phase)

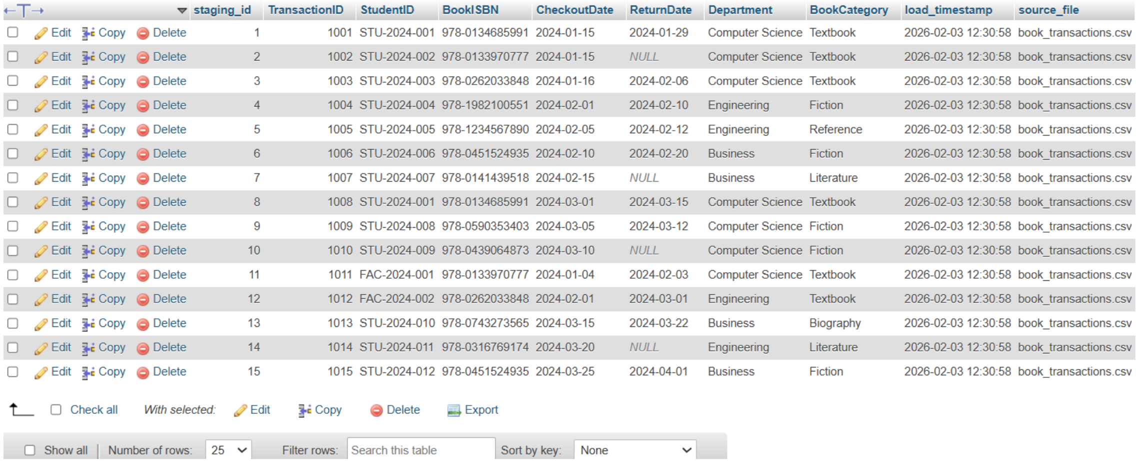
“Transforms” refers to the process of cleaning, standardizing, and converting the extracted data into a consistent format that fits the data warehouse structure.



```
# 2. TRANSFORM
logger.info("\n--- TRANSFORM PHASE ---")
df_books_clean = self.transformer.transform_book_transactions(df_books)
df_digital_clean = self.transformer.transform_digital_usage(df_digital)
df_rooms_clean = self.transformer.transform_room_bookings(df_rooms)
```

Figure 5:Transforms

x

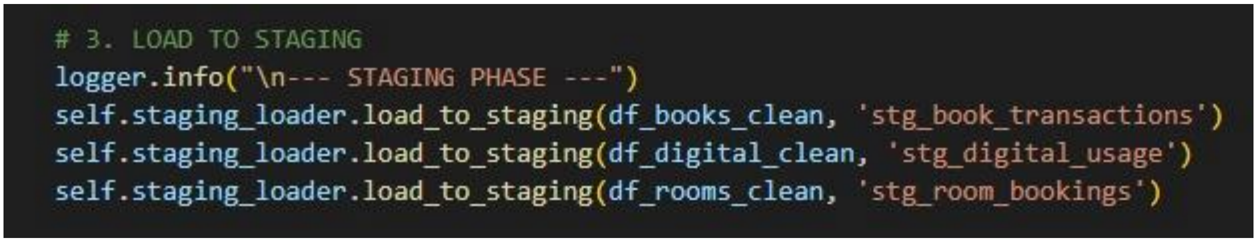


←T→	staging_id	TransactionID	StudentID	BookISBN	CheckoutDate	ReturnDate	Department	BookCategory	load_timestamp	source_file
☐ Edit Copy Delete	1	1001	STU-2024-001	978-0134685991	2024-01-15	2024-01-29	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	2	1002	STU-2024-002	978-0133970777	2024-01-15	NULL	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	3	1003	STU-2024-003	978-0262033848	2024-01-16	2024-02-06	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	4	1004	STU-2024-004	978-1982100551	2024-02-01	2024-02-10	Engineering	Fiction	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	5	1005	STU-2024-005	978-1234567890	2024-02-05	2024-02-12	Engineering	Reference	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	6	1006	STU-2024-006	978-0451524935	2024-02-10	2024-02-20	Business	Fiction	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	7	1007	STU-2024-007	978-0141439518	2024-02-15	NULL	Business	Literature	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	8	1008	STU-2024-001	978-0134685991	2024-03-01	2024-03-15	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	9	1009	STU-2024-008	978-0590353403	2024-03-05	2024-03-12	Computer Science	Fiction	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	10	1010	STU-2024-009	978-0439064873	2024-03-10	NULL	Computer Science	Fiction	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	11	1011	FAC-2024-001	978-0133970777	2024-01-04	2024-02-03	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	12	1012	FAC-2024-002	978-0262033848	2024-02-01	2024-03-01	Engineering	Textbook	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	13	1013	STU-2024-010	978-0743273565	2024-03-15	2024-03-22	Business	Biography	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	14	1014	STU-2024-011	978-0316769174	2024-03-20	NULL	Engineering	Literature	2026-02-03 12:30:58	book_transactions.csv
☐ Edit Copy Delete	15	1015	STU-2024-012	978-0451524935	2024-03-25	2024-04-01	Business	Fiction	2026-02-03 12:30:58	book_transactions.csv

Figure 6:sample output after cleaning

5.3 Load to Staging

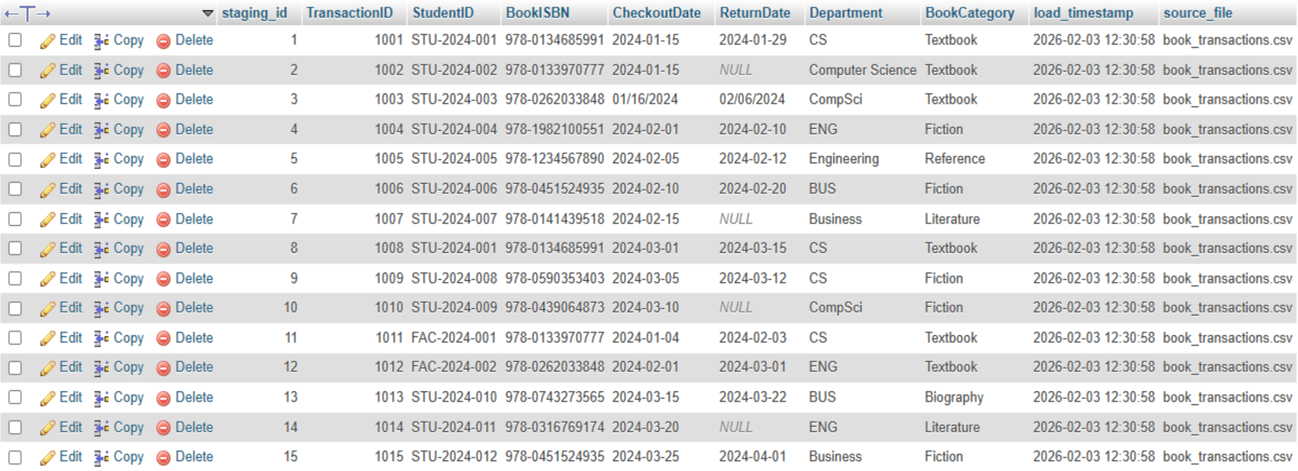
The staging area serves as a temporary workspace where data inconsistencies-like missing values, duplicate rows, and inconsistent formats-can be detected and corrected before moving into the data warehouse. This step ensures the warehouse receives clean, standardized, and validated data for analysis



```
# 3. LOAD TO STAGING
logger.info("\n--- STAGING PHASE ---")
self.staging_loader.load_to_staging(df_books_clean, 'stg_book_transactions')
self.staging_loader.load_to_staging(df_digital_clean, 'stg_digital_usage')
self.staging_loader.load_to_staging(df_rooms_clean, 'stg_room_bookings')
```

Figure 7:Staging

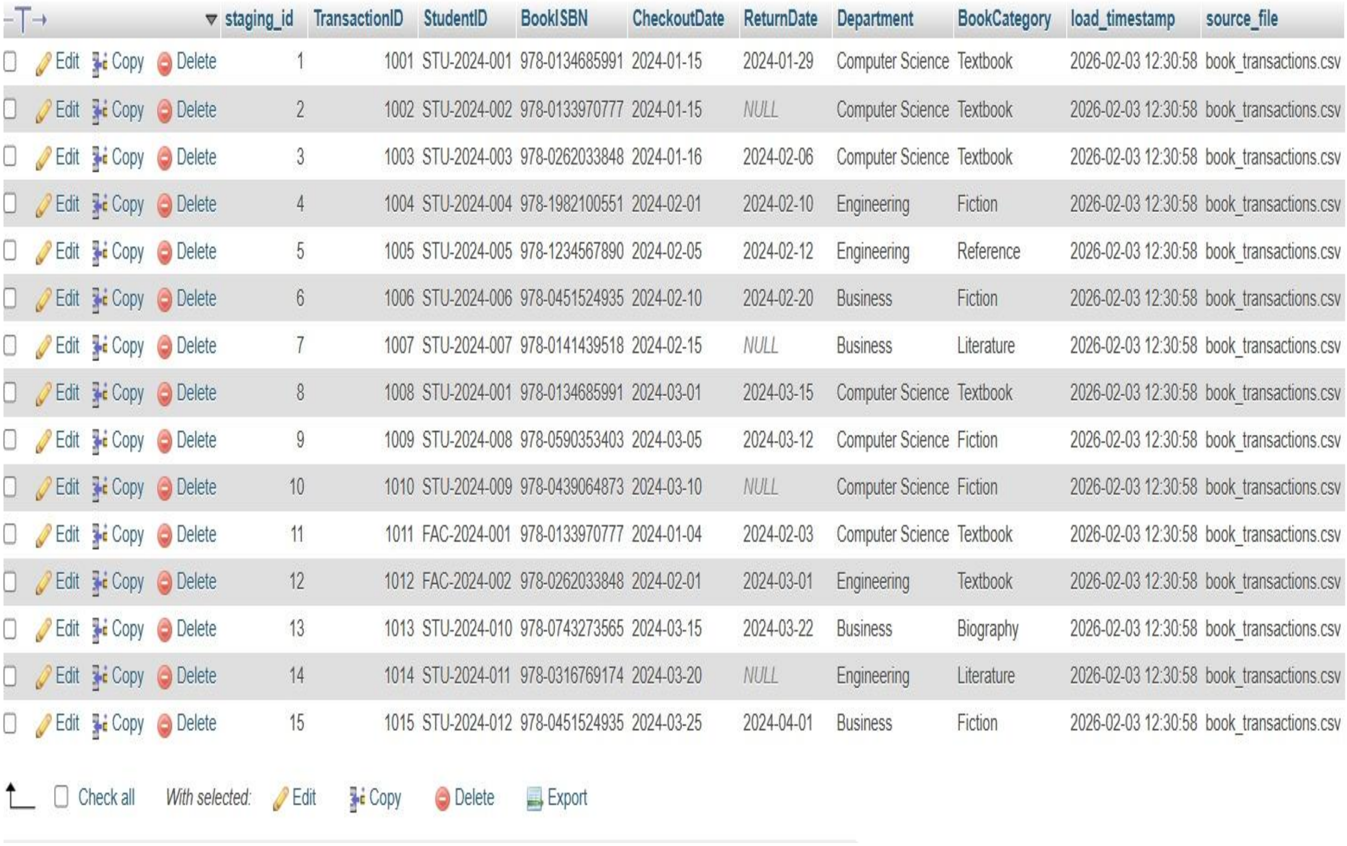
staging book transactions Before ETL: Data had inconsistent formats, missing values, and non-standardized text.



staging_id	TransactionID	StudentID	BookISBN	CheckoutDate	ReturnDate	Department	BookCategory	load_timestamp	source_file
1	1001	STU-2024-001	978-0134685991	2024-01-15	2024-01-29	CS	Textbook	2026-02-03 12:30:58	book_transactions.csv
2	1002	STU-2024-002	978-0133970777	2024-01-15	NULL	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
3	1003	STU-2024-003	978-0262033848	01/16/2024	02/06/2024	CompSci	Textbook	2026-02-03 12:30:58	book_transactions.csv
4	1004	STU-2024-004	978-1982100551	2024-02-01	2024-02-10	ENG	Fiction	2026-02-03 12:30:58	book_transactions.csv
5	1005	STU-2024-005	978-1234567890	2024-02-05	2024-02-12	Engineering	Reference	2026-02-03 12:30:58	book_transactions.csv
6	1006	STU-2024-006	978-0451524935	2024-02-10	2024-02-20	BUS	Fiction	2026-02-03 12:30:58	book_transactions.csv
7	1007	STU-2024-007	978-0141439518	2024-02-15	NULL	Business	Literature	2026-02-03 12:30:58	book_transactions.csv
8	1008	STU-2024-001	978-0134685991	2024-03-01	2024-03-15	CS	Textbook	2026-02-03 12:30:58	book_transactions.csv
9	1009	STU-2024-008	978-0590353403	2024-03-05	2024-03-12	CS	Fiction	2026-02-03 12:30:58	book_transactions.csv
10	1010	STU-2024-009	978-0439064873	2024-03-10	NULL	CompSci	Fiction	2026-02-03 12:30:58	book_transactions.csv
11	1011	FAC-2024-001	978-0133970777	2024-01-04	2024-02-03	CS	Textbook	2026-02-03 12:30:58	book_transactions.csv
12	1012	FAC-2024-002	978-0262033848	2024-02-01	2024-03-01	ENG	Textbook	2026-02-03 12:30:58	book_transactions.csv
13	1013	STU-2024-010	978-0743273565	2024-03-15	2024-03-22	BUS	Biography	2026-02-03 12:30:58	book_transactions.csv
14	1014	STU-2024-011	978-0316769174	2024-03-20	NULL	ENG	Literature	2026-02-03 12:30:58	book_transactions.csv
15	1015	STU-2024-012	978-0451524935	2024-03-25	2024-04-01	Business	Fiction	2026-02-03 12:30:58	book_transactions.csv

Figure 8: staging book transaction Before ETL

After ETL Normalization: All staging tables were cleaned, standardized, and loaded into the warehouse dimensions and fact table, ready for analytics.



staging_id	TransactionID	StudentID	BookISBN	CheckoutDate	ReturnDate	Department	BookCategory	load_timestamp	source_file
1	1001	STU-2024-001	978-0134685991	2024-01-15	2024-01-29	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
2	1002	STU-2024-002	978-0133970777	2024-01-15	NULL	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
3	1003	STU-2024-003	978-0262033848	2024-01-16	2024-02-06	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
4	1004	STU-2024-004	978-1982100551	2024-02-01	2024-02-10	Engineering	Fiction	2026-02-03 12:30:58	book_transactions.csv
5	1005	STU-2024-005	978-1234567890	2024-02-05	2024-02-12	Engineering	Reference	2026-02-03 12:30:58	book_transactions.csv
6	1006	STU-2024-006	978-0451524935	2024-02-10	2024-02-20	Business	Fiction	2026-02-03 12:30:58	book_transactions.csv
7	1007	STU-2024-007	978-0141439518	2024-02-15	NULL	Business	Literature	2026-02-03 12:30:58	book_transactions.csv
8	1008	STU-2024-001	978-0134685991	2024-03-01	2024-03-15	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
9	1009	STU-2024-008	978-0590353403	2024-03-05	2024-03-12	Computer Science	Fiction	2026-02-03 12:30:58	book_transactions.csv
10	1010	STU-2024-009	978-0439064873	2024-03-10	NULL	Computer Science	Fiction	2026-02-03 12:30:58	book_transactions.csv
11	1011	FAC-2024-001	978-0133970777	2024-01-04	2024-02-03	Computer Science	Textbook	2026-02-03 12:30:58	book_transactions.csv
12	1012	FAC-2024-002	978-0262033848	2024-02-01	2024-03-01	Engineering	Textbook	2026-02-03 12:30:58	book_transactions.csv
13	1013	STU-2024-010	978-0743273565	2024-03-15	2024-03-22	Business	Biography	2026-02-03 12:30:58	book_transactions.csv
14	1014	STU-2024-011	978-0316769174	2024-03-20	NULL	Engineering	Literature	2026-02-03 12:30:58	book_transactions.csv
15	1015	STU-2024-012	978-0451524935	2024-03-25	2024-04-01	Business	Fiction	2026-02-03 12:30:58	book_transactions.csv

Figure 9: staging book transaction After ETL

6 TEAM MEMBER 3 REPORT

Team Member 3 was responsible for transforming the processed data into meaningful insights for decision makers. The focus was to implement OLAP operations, create analytical queries, design dashboards, and ensure system security and data protection. This role ensures that stakeholders can easily explore data, identify trends, and make informed decisions.

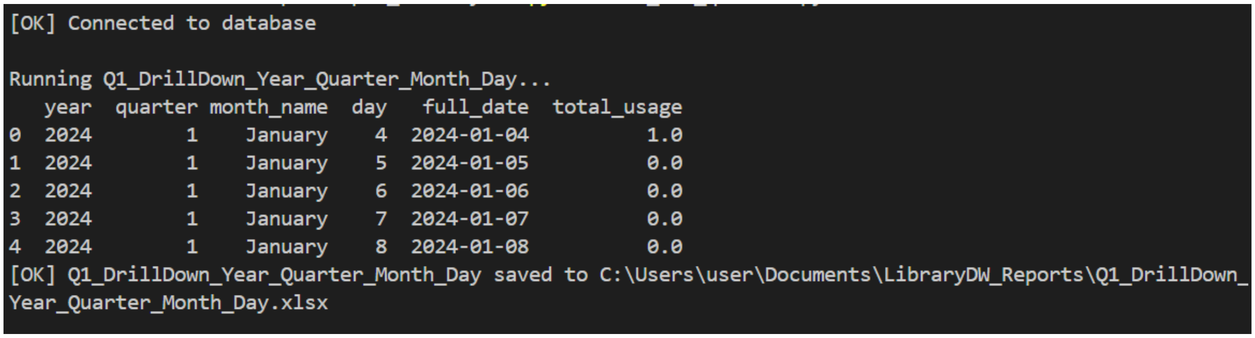
6.1 OLAP Implementation

OLAP (Online Analytical Processing) was implemented to allow multi-dimensional analysis of library data. This enables users to analyze information from different perspectives such as time, department, resource type, and student category.

6.2 Drill-Down Implementation (Year → Quarter → Month → Day)

Drill-down allows users to move from summarized data to detailed data.

By using python:



```
[OK] Connected to database

Running Q1_DrillDown_Year_Quarter_Month_Day...
   year  quarter month_name  day   full_date  total_usage
0  2024        1    January    4  2024-01-04          1.0
1  2024        1    January    5  2024-01-05          0.0
2  2024        1    January    6  2024-01-06          0.0
3  2024        1    January    7  2024-01-07          0.0
4  2024        1    January    8  2024-01-08          0.0
[OK] Q1_DrillDown_Year_Quarter_Month_Day saved to C:\Users\user\Documents\LibraryDW_Reports\Q1_DrillDown_
Year_Quarter_Month_Day.xlsx
```

Figure 10:Drill-Down Implementation (Year → Quarter → Month → Day

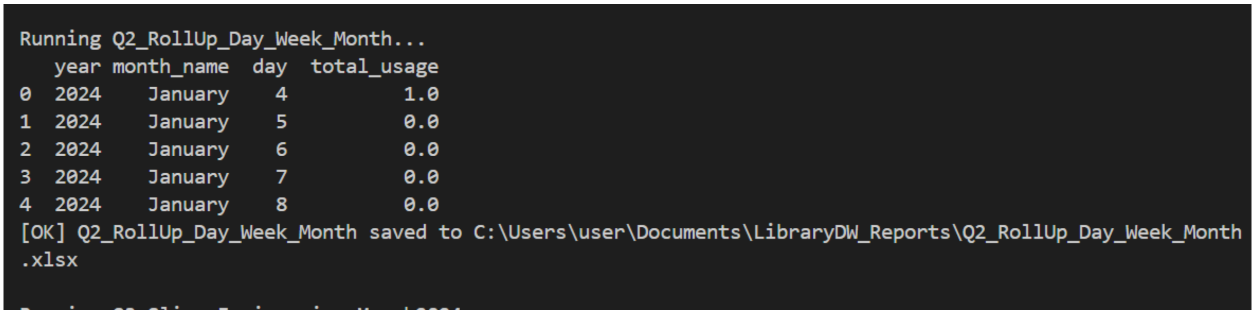
)

Library managers can view total yearly usage and then drill down to see usage per quarter, month, and individual day.

6.3 Roll-Up Implementation (Day → Week → Month)

Roll-up is the opposite of drill-down. It summarizes detailed data into higher-level totals.

By using python:



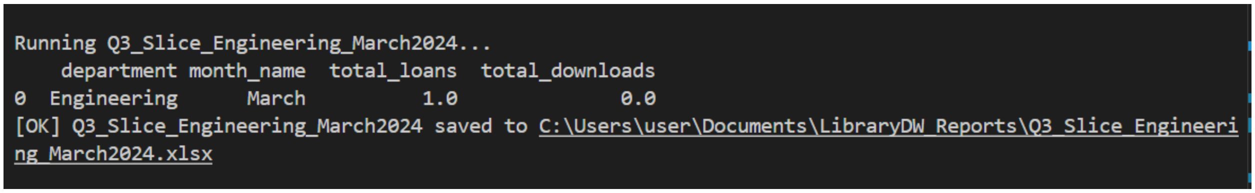
```
Running Q2_RollUp_Day_Week_Month...
   year month_name  day  total_usage
0  2024    January    4          1.0
1  2024    January    5          0.0
2  2024    January    6          0.0
3  2024    January    7          0.0
4  2024    January    8          0.0
[OK] Q2_RollUp_Day_Week_Month saved to C:\Users\user\Documents\LibraryDW_Reports\Q2_RollUp_Day_Week_Month
.xlsx
```

Figure 11:Roll-Up Implementation (Day → Week → Month)

6.4 Slicing Operation

Slicing filters data by selecting a single dimension value.

Viewing only data for **March 2024**.

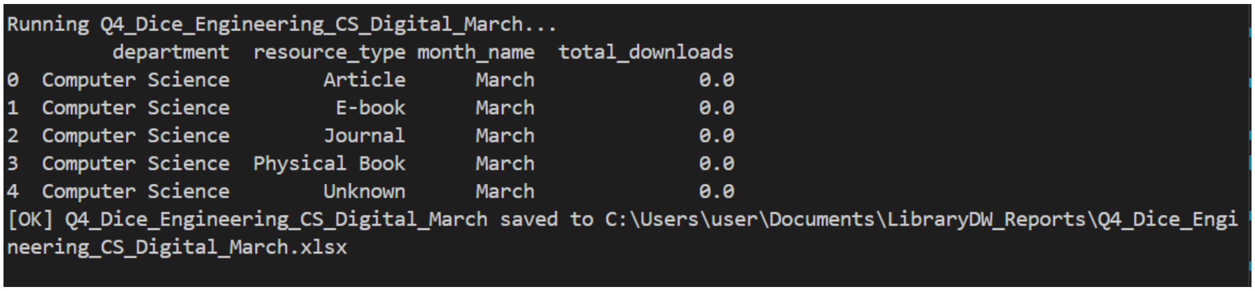


```
Running Q3_Slice_Engineering_March2024...
    department month_name  total_loans  total_downloads
0  Engineering      March          1.0              0.0
[OK] Q3_Slice_Engineering_March2024 saved to C:\Users\user\Documents\LibraryDW_Reports\Q3_Slice_Engineeri
ng_March2024.xlsx
```

Figure 12:Slicing Operation

6.5 Dicing Operation

Dicing filters data using multiple dimensions at the same time.

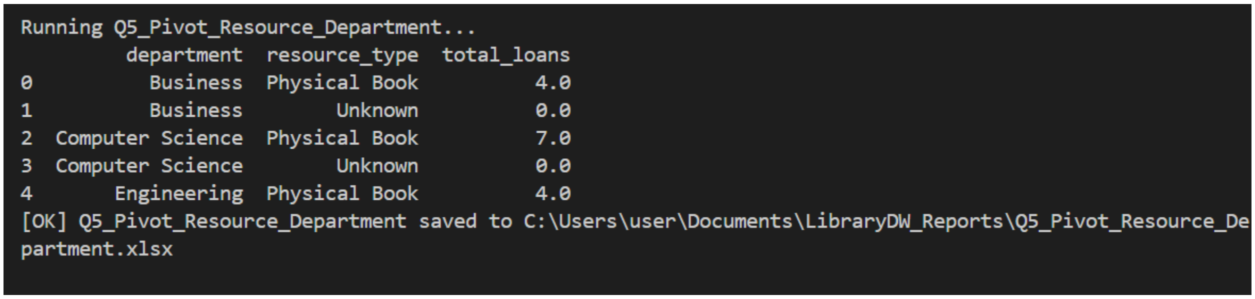


```
Running Q4_Dice_Engineering_CS_Digital_March...
         department  resource_type month_name  total_downloads
0  Computer Science        Article      March              0.0
1  Computer Science         E-book      March              0.0
2  Computer Science        Journal      March              0.0
3  Computer Science  Physical Book      March              0.0
4  Computer Science        Unknown      March              0.0
[OK] Q4_Dice_Engineering_CS_Digital_March saved to C:\Users\user\Documents\LibraryDW_Reports\Q4_Dice_Engi
neering_CS_Digital_March.xlsx
```

Figure 13:Dicing Operation

6.6 Pivot Analysis

Pivot analysis reorganizes data into a matrix format.

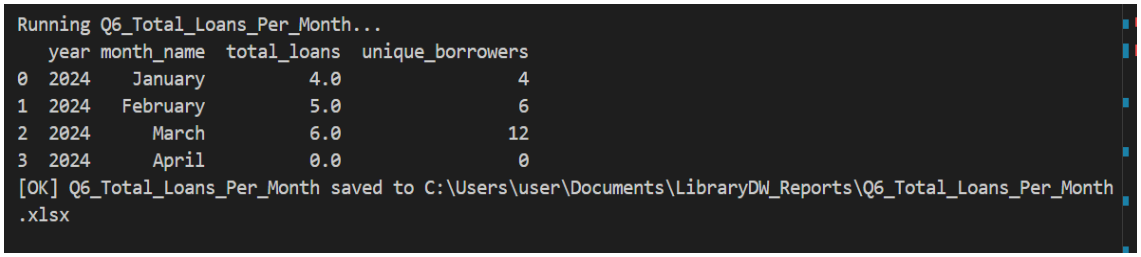


```
Running Q5_Pivot_Resource_Department...
         department  resource_type  total_loans
0          Business  Physical Book          4.0
1          Business        Unknown          0.0
2  Computer Science  Physical Book          7.0
3  Computer Science        Unknown          0.0
4       Engineering  Physical Book          4.0
[OK] Q5_Pivot_Resource_Department saved to C:\Users\user\Documents\LibraryDW_Reports\Q5_Pivot_Resource_De
partment.xlsx
```

Figure 14: Pivot Analysis

SQL Query Development: More than 10 business intelligence queries were developed to answer library management questions.

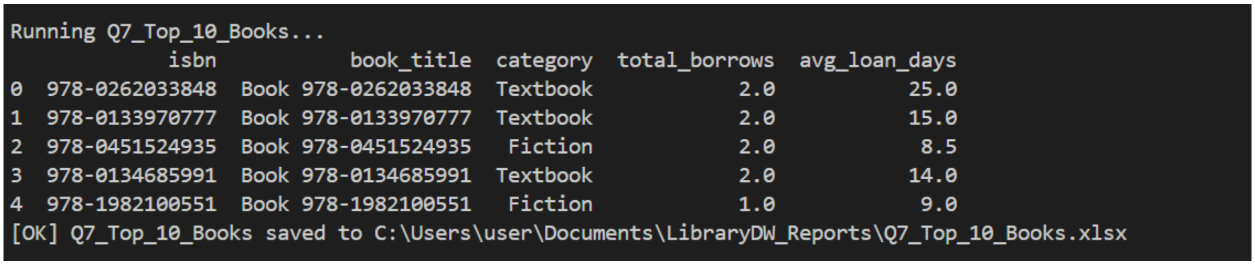
Aggregation Queries



```
Running Q6_Total_Loans_Per_Month...
   year month_name  total_loans  unique_borrowers
0  2024    January          4.0                 4
1  2024   February          5.0                 6
2  2024      March          6.0                12
3  2024      April          0.0                 0
[OK] Q6_Total_Loans_Per_Month saved to C:\Users\user\Documents\LibraryDW_Reports\Q6_Total_Loans_Per_Month
.xlsx
```

Figure 15:total loans

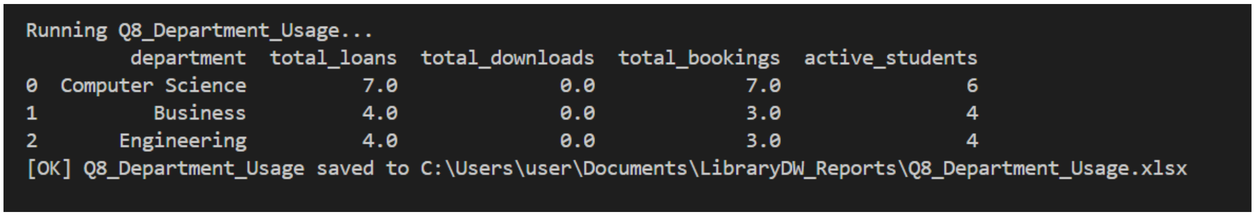
Ranking Queries



```
Running Q7_Top_10_Books...
             isbn           book_title  category  total_borrows  avg_loan_days
0  978-0262033848  Book 978-0262033848  Textbook            2.0           25.0
1  978-0133970777  Book 978-0133970777  Textbook            2.0           15.0
2  978-0451524935  Book 978-0451524935   Fiction            2.0            8.5
3  978-0134685991  Book 978-0134685991  Textbook            2.0           14.0
4  978-1982100551  Book 978-1982100551   Fiction            1.0            9.0
[OK] Q7_Top_10_Books saved to C:\Users\user\Documents\LibraryDW_Reports\Q7_Top_10_Books.xlsx
```

Figure 16: top 10 borrowed books

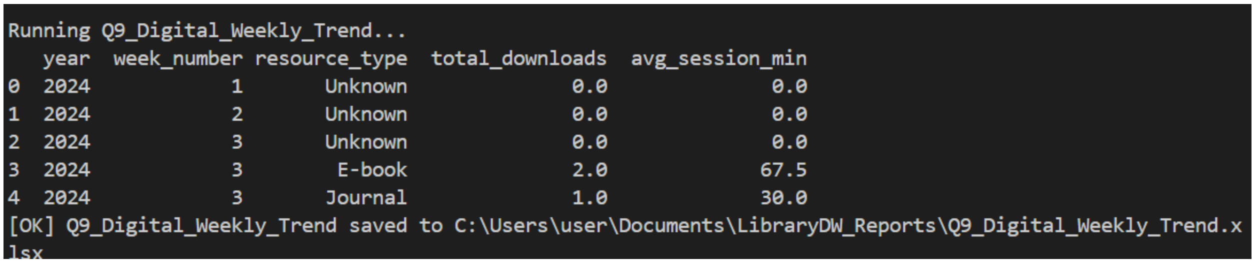
Most active departments



```
Running Q8_Department_Usage...
         department  total_loans  total_downloads  total_bookings  active_students
0  Computer Science          7.0              0.0             7.0                6
1          Business          4.0              0.0             3.0                4
2       Engineering          4.0              0.0             3.0                4
[OK] Q8_Department_Usage saved to C:\Users\user\Documents\LibraryDW_Reports\Q8_Department_Usage.xlsx
```

Figure 16: departments' usage

Trend Analysis Queries



```
Running Q9_Digital_Weekly_Trend...
   year  week_number resource_type  total_downloads  avg_session_min
0  2024            1       Unknown              0.0              0.0
1  2024            2       Unknown              0.0              0.0
2  2024            3       Unknown              0.0              0.0
3  2024            3        E-book              2.0             67.5
4  2024            3       Journal              1.0             30.0
[OK] Q9_Digital_Weekly_Trend saved to C:\Users\user\Documents\LibraryDW_Reports\Q9_Digital_Weekly_Trend.x
lsx
```

Figure 17: weekly digital trend

6.7 Dashboards and Data Visualization

Three interactive dashboards were created to support different users.

6.8 Executive Dashboard

Target Users: Library Director and Senior Management

Contains:

- Total library usage
- Monthly trends
- Digital vs physical usage comparison
- KPIs (Key Performance Indicators)

Purpose:

Provides high-level overview for strategic decision making.

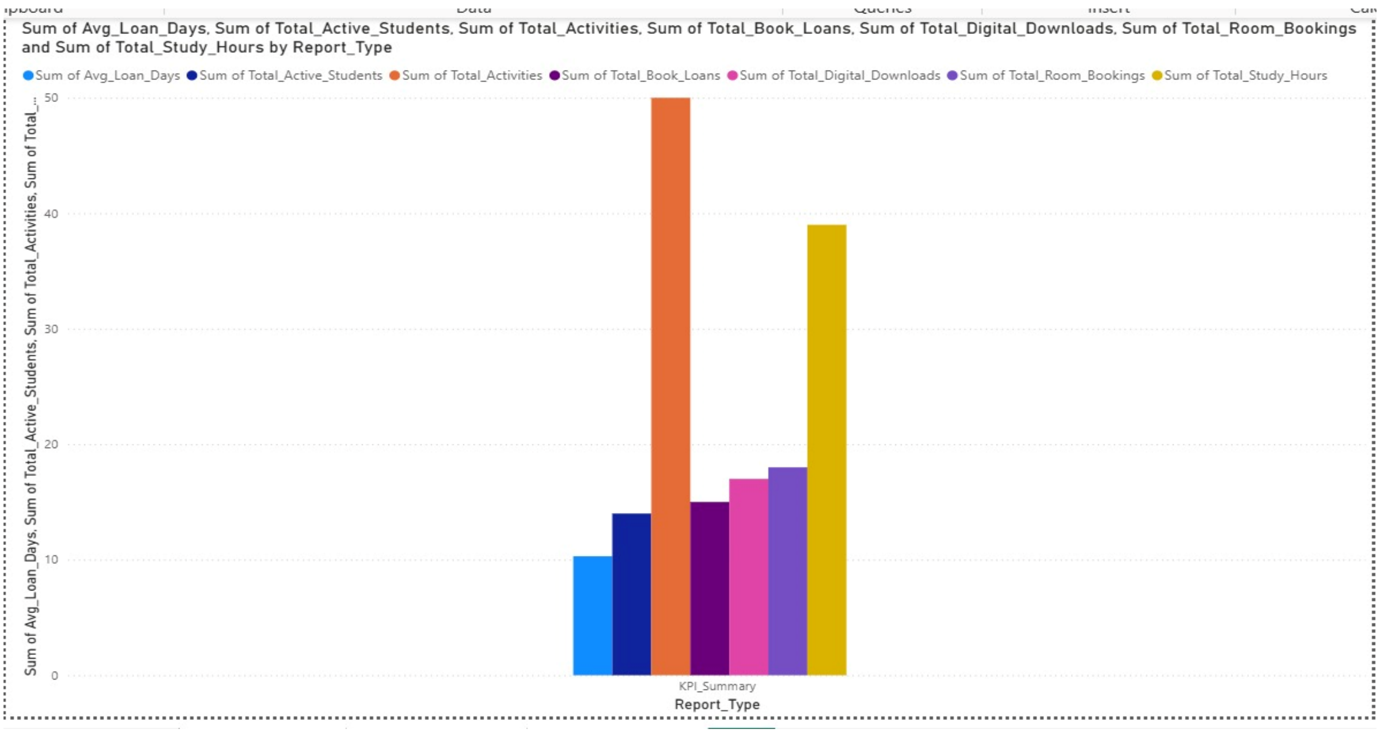


Figure 18:Figure 19: Executive Dashboard bar chart

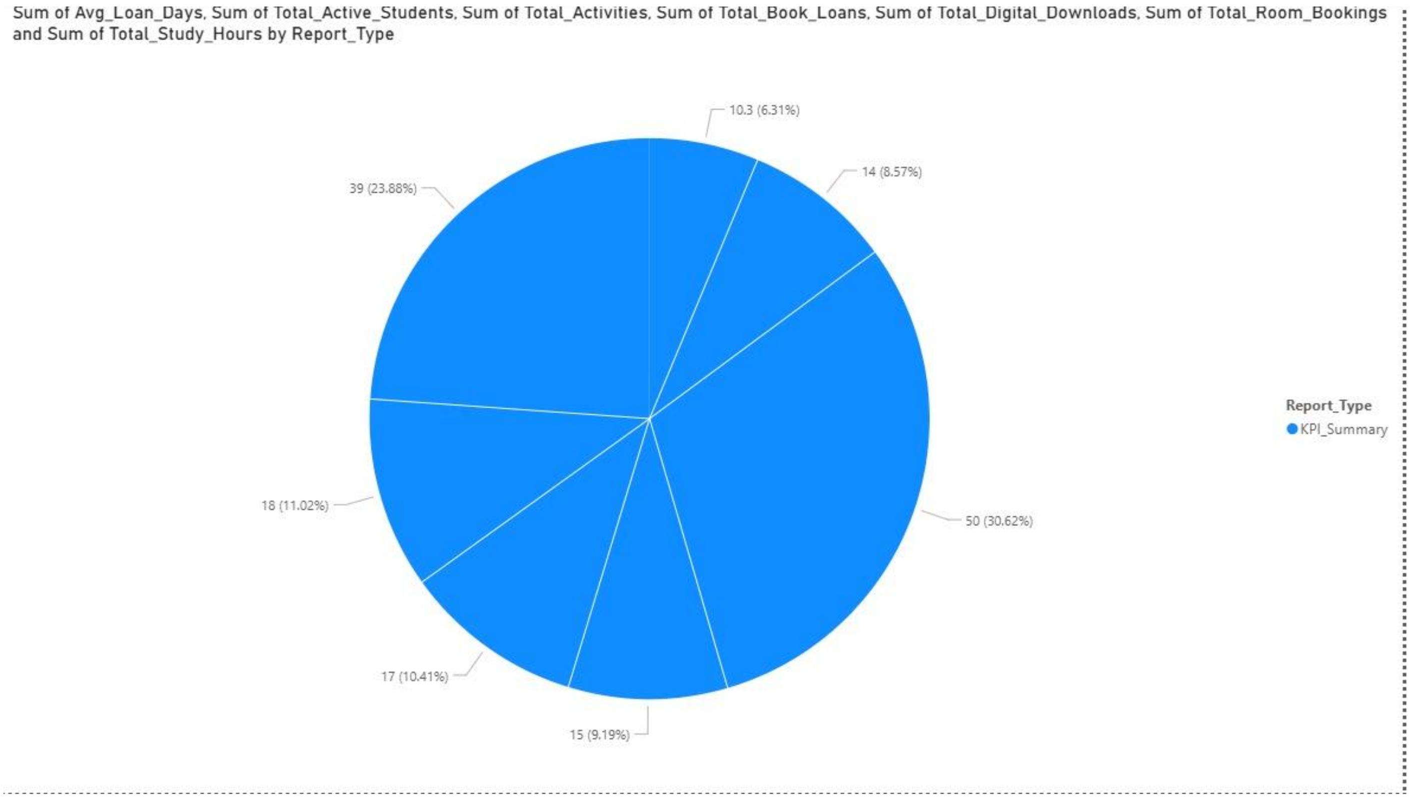


Figure 19:Figure 19: Executive Dashboard pie chart

6.9 Department Dashboard

Target Users: Department Heads

Contains:

- Usage per department
- Resource consumption by faculty
- Student engagement metrics

Purpose:

Helps departments monitor their resource utilization.

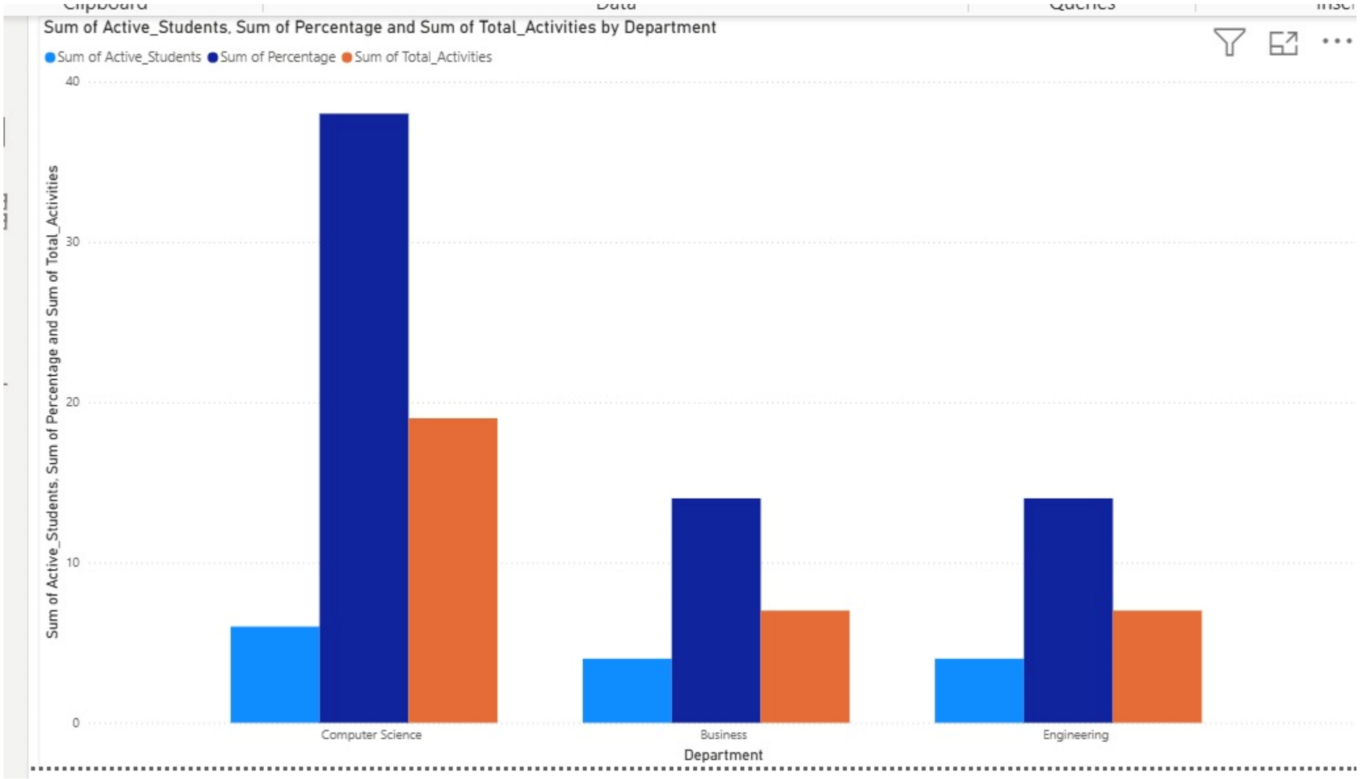


Figure 20: Department Dashboard bar chart

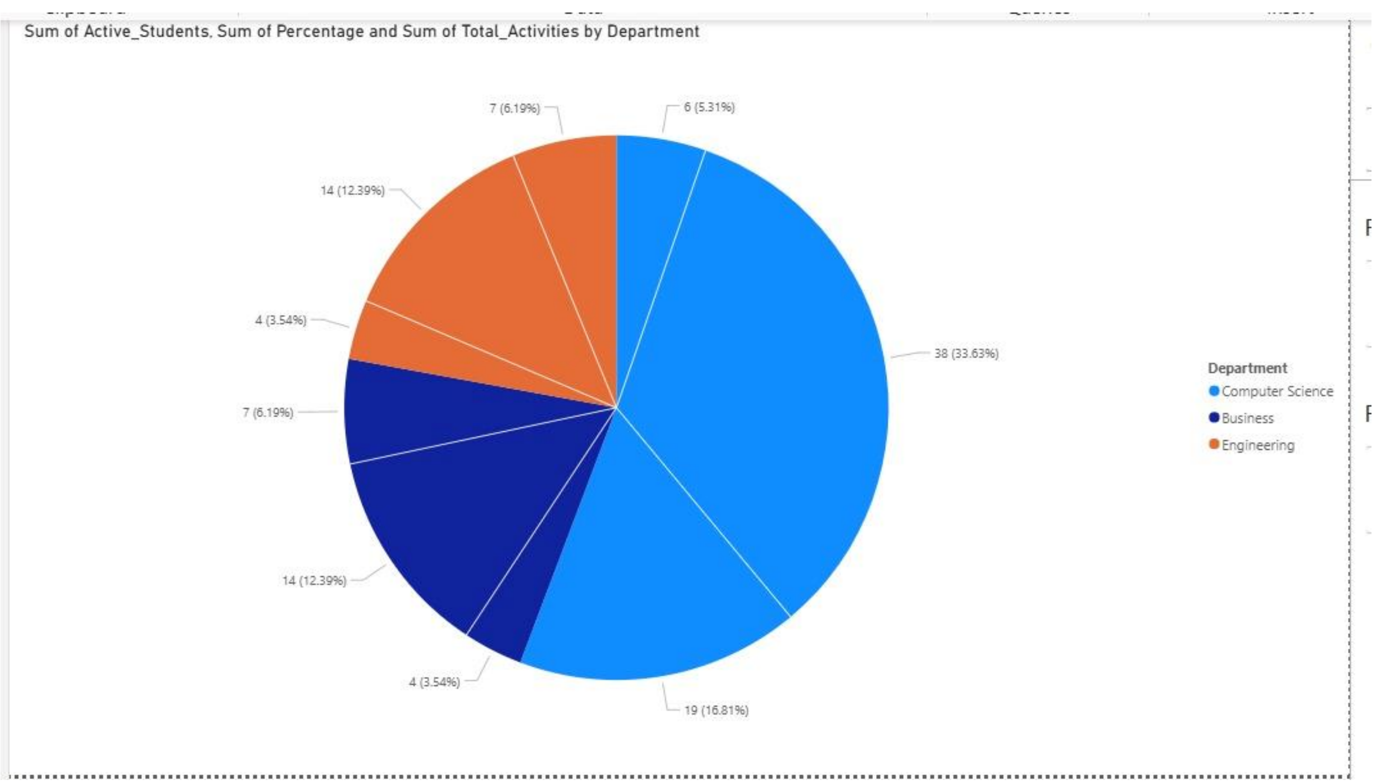


Figure 21: Department Dashboard pie chart

6.10 Operational Dashboard

Target Users: Library Staff

Contains:

- Daily room bookings
- Peak usage hours
- Active users

Purpose:

Supports daily operational planning and staffing decisions.

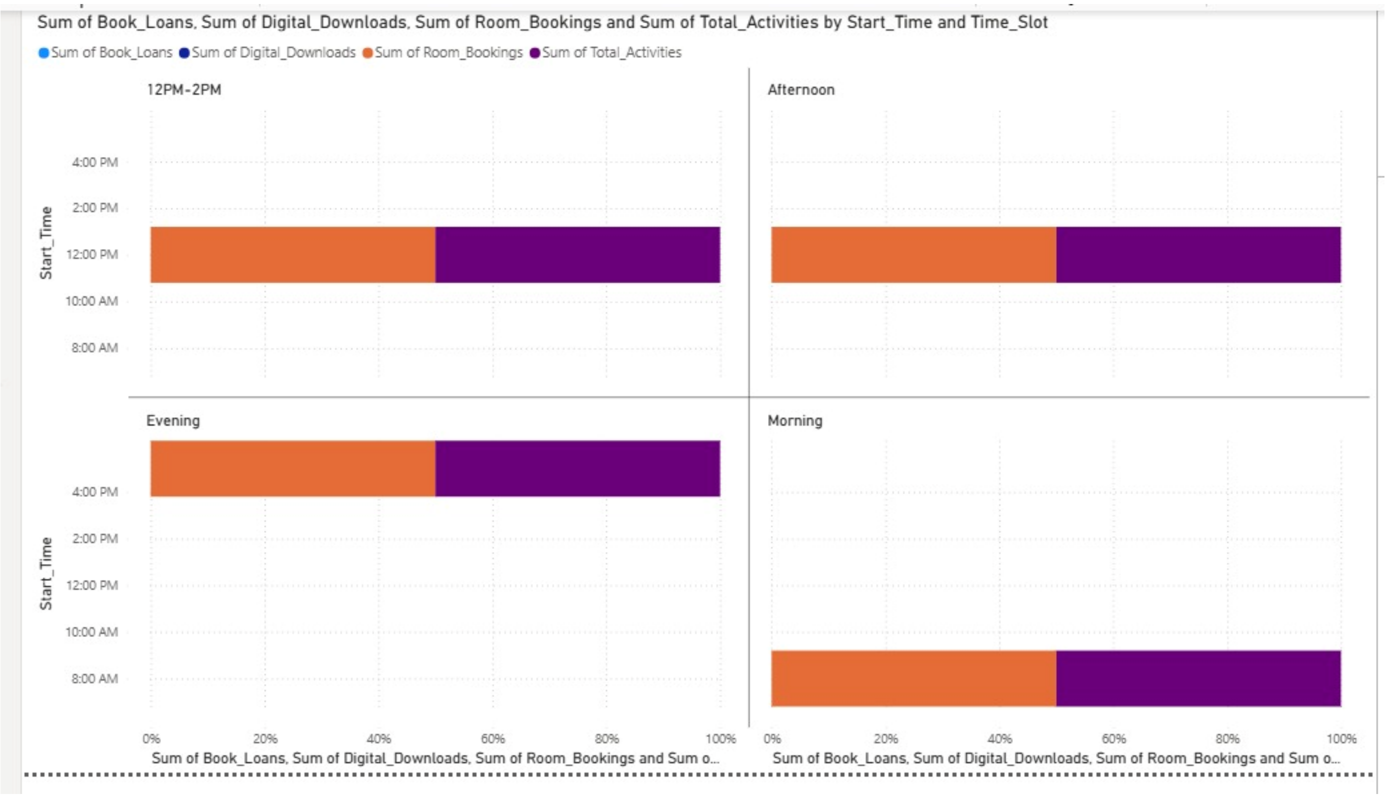


Figure 22: Operational Dashboard bar chart

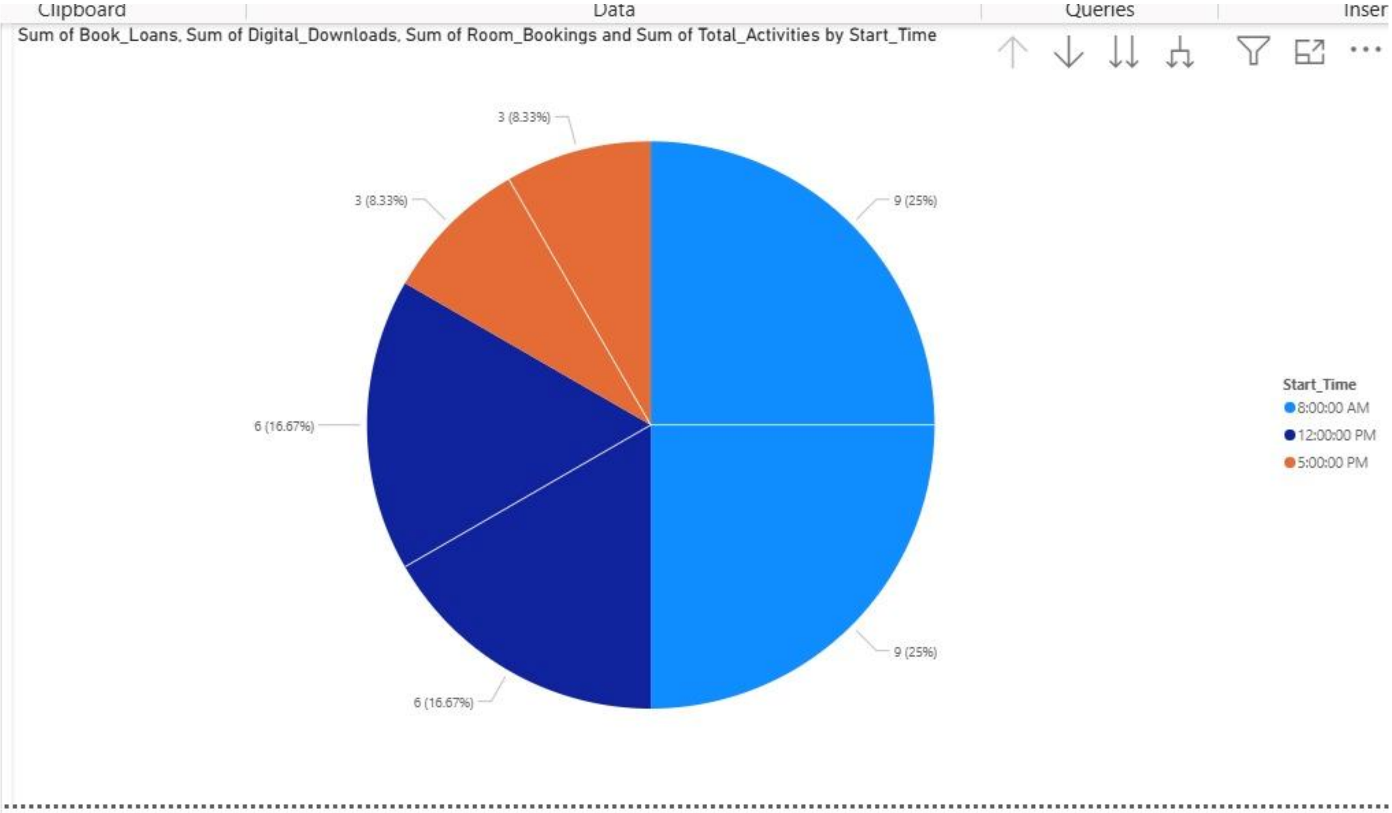
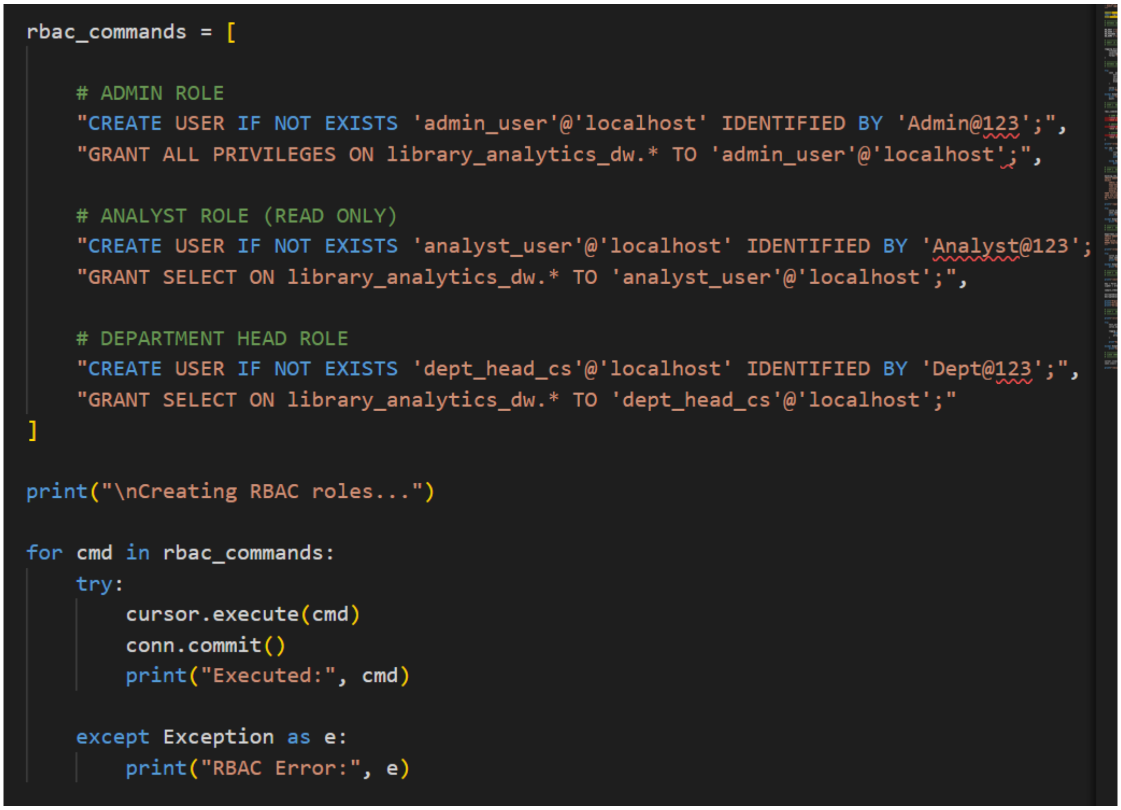


Figure 23:Operational Dashboard pie chart

6.11 Security and Access Control (RBAC)

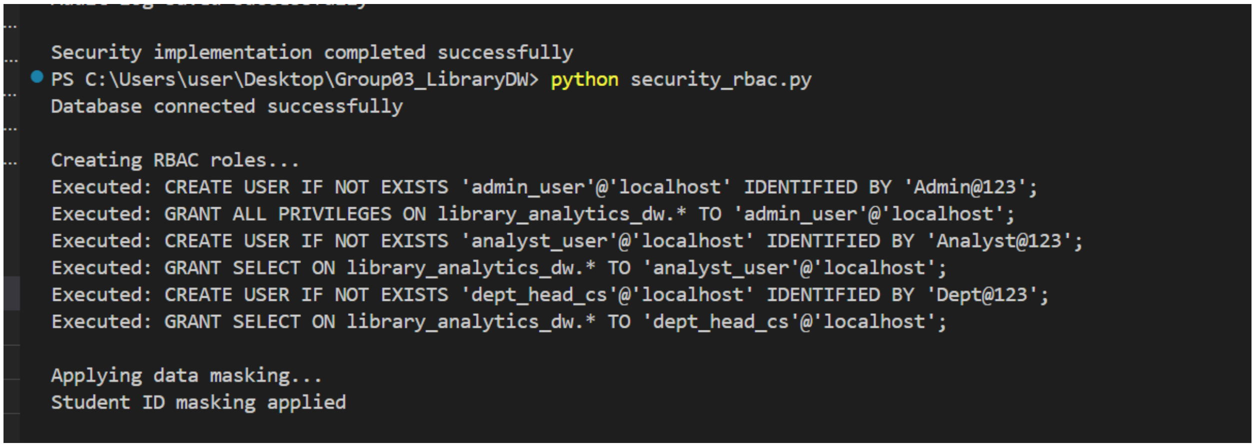
To protect sensitive data, Role-Based Access Control (RBAC) was implemented.

User Roles



```python
rbac_commands = [

    # ADMIN ROLE
    "CREATE USER IF NOT EXISTS 'admin_user'@'localhost' IDENTIFIED BY 'Admin@123';",
    "GRANT ALL PRIVILEGES ON library_analytics_dw.* TO 'admin_user'@'localhost';",

    # ANALYST ROLE (READ ONLY)
    "CREATE USER IF NOT EXISTS 'analyst_user'@'localhost' IDENTIFIED BY 'Analyst@123';
    "GRANT SELECT ON library_analytics_dw.* TO 'analyst_user'@'localhost';",

    # DEPARTMENT HEAD ROLE
    "CREATE USER IF NOT EXISTS 'dept_head_cs'@'localhost' IDENTIFIED BY 'Dept@123';",
    "GRANT SELECT ON library_analytics_dw.* TO 'dept_head_cs'@'localhost';"
]

print("\nCreating RBAC roles...")

for cmd in rbac_commands:
    try:
        cursor.execute(cmd)
        conn.commit()
        print("Executed:", cmd)

    except Exception as e:
        print("RBAC Error:", e)
```

Figure 24:Role-Based

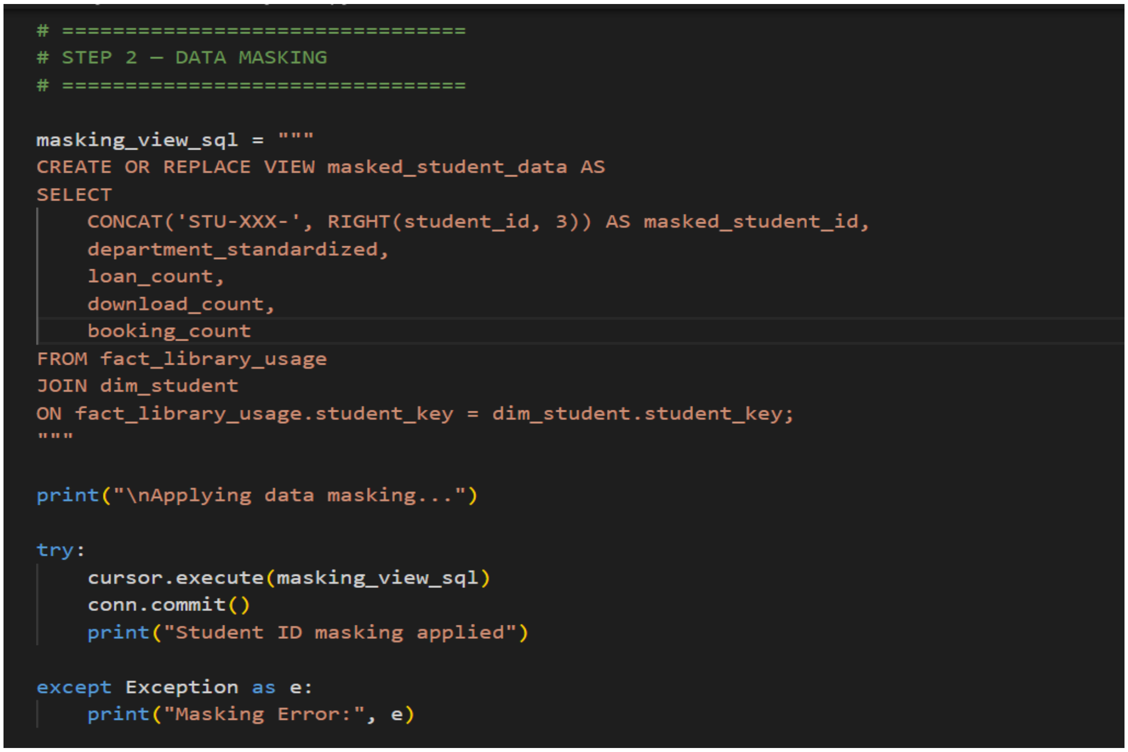


```
Security implementation completed successfully
PS C:\Users\user\Desktop\Group03_LibraryDW> python security_rbac.py
Database connected successfully

Creating RBAC roles...
Executed: CREATE USER IF NOT EXISTS 'admin_user'@'localhost' IDENTIFIED BY 'Admin@123';
Executed: GRANT ALL PRIVILEGES ON library_analytics_dw.* TO 'admin_user'@'localhost';
Executed: CREATE USER IF NOT EXISTS 'analyst_user'@'localhost' IDENTIFIED BY 'Analyst@123';
Executed: GRANT SELECT ON library_analytics_dw.* TO 'analyst_user'@'localhost';
Executed: CREATE USER IF NOT EXISTS 'dept_head_cs'@'localhost' IDENTIFIED BY 'Dept@123';
Executed: GRANT SELECT ON library_analytics_dw.* TO 'dept_head_cs'@'localhost';

Applying data masking...
Student ID masking applied
```

Figure 25:output of Role-Based

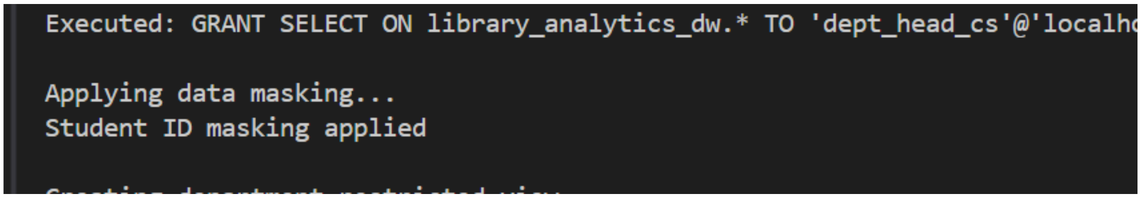
Data Masking

Sensitive fields such as StudentID were partially hidden.



```
# ================================
# STEP 2 — DATA MASKING
# ================================

masking_view_sql = """
CREATE OR REPLACE VIEW masked_student_data AS
SELECT
    CONCAT('STU-XXX-', RIGHT(student_id, 3)) AS masked_student_id,
    department_standardized,
    loan_count,
    download_count,
    booking_count
FROM fact_library_usage
JOIN dim_student
ON fact_library_usage.student_key = dim_student.student_key;
"""

print("\nApplying data masking...")

try:
    cursor.execute(masking_view_sql)
    conn.commit()
    print("Student ID masking applied")

except Exception as e:
    print("Masking Error:", e)
```

Figure 26:Data Masking



```
Executed: GRANT SELECT ON library_analytics_dw.* TO 'dept_head_cs'@'localh

Applying data masking...
Student ID masking applied

Creating department restricted view
```

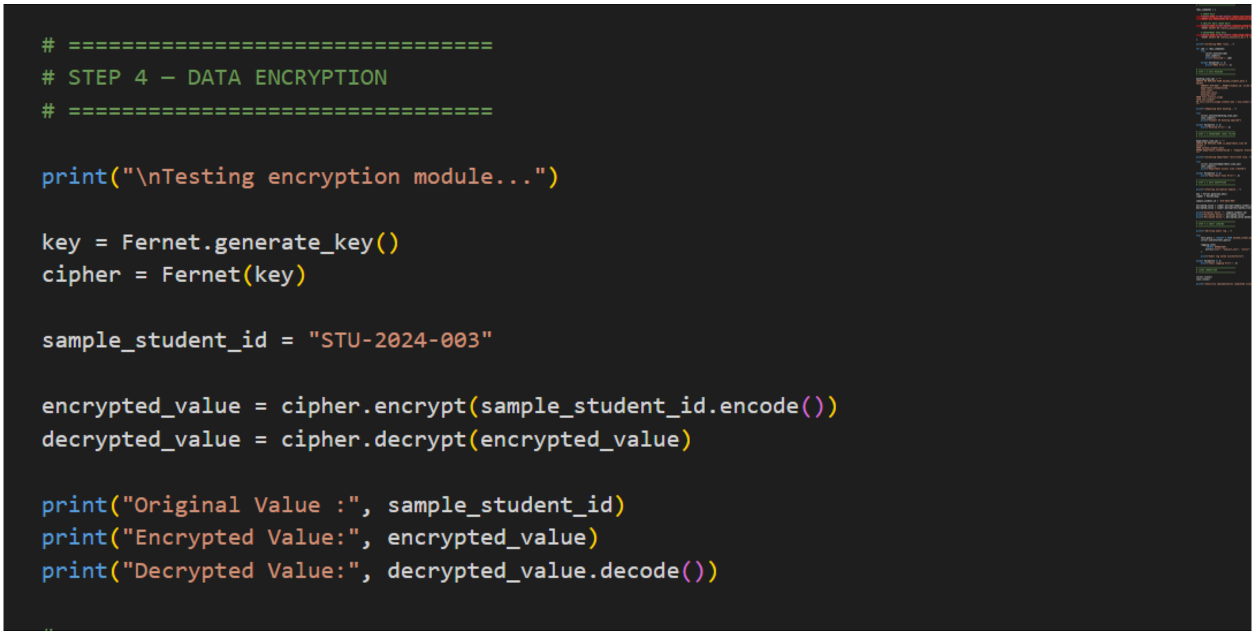
Figure 27: output Data Masking

Purpose:

Protects student privacy while allowing analysis.

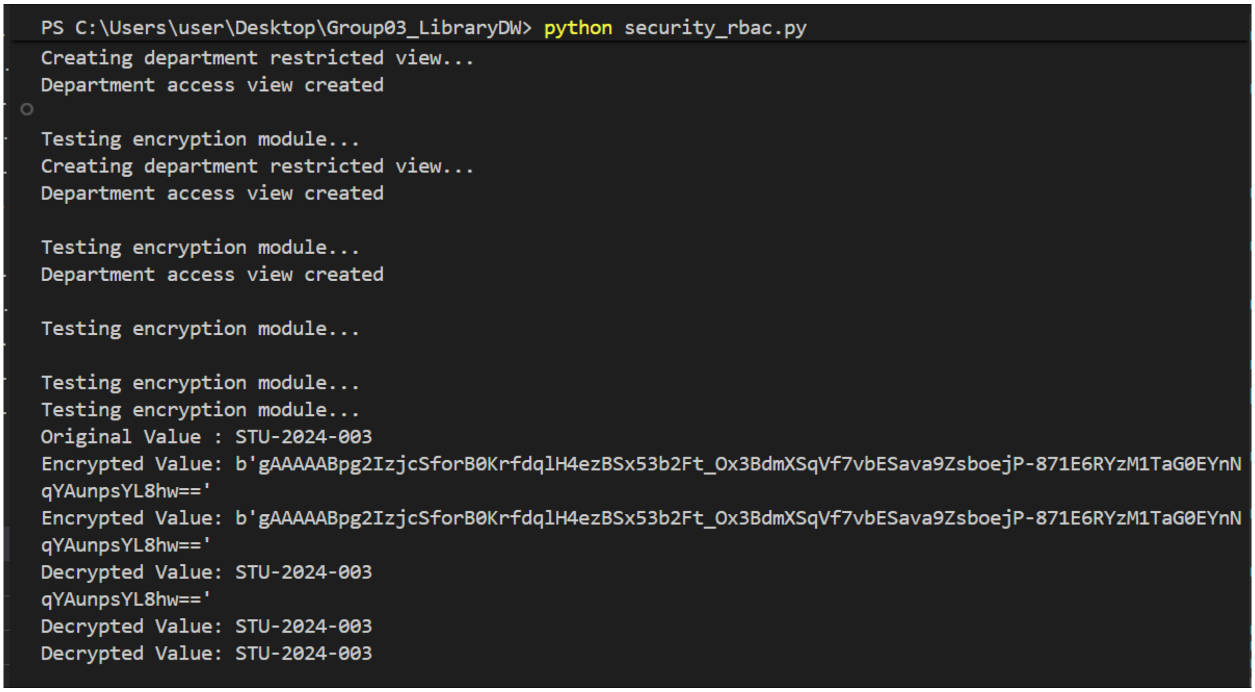
Encryption

- Data in transit protected using secure database connections.
- Data at rest is protected through database security configurations.



```
# ================================
# STEP 4 — DATA ENCRYPTION
# ================================

print("\nTesting encryption module...")

key = Fernet.generate_key()
cipher = Fernet(key)

sample_student_id = "STU-2024-003"

encrypted_value = cipher.encrypt(sample_student_id.encode())
decrypted_value = cipher.decrypt(encrypted_value)

print("Original Value :", sample_student_id)
print("Encrypted Value:", encrypted_value)
print("Decrypted Value:", decrypted_value.decode())
```

Figure 28:Encryption



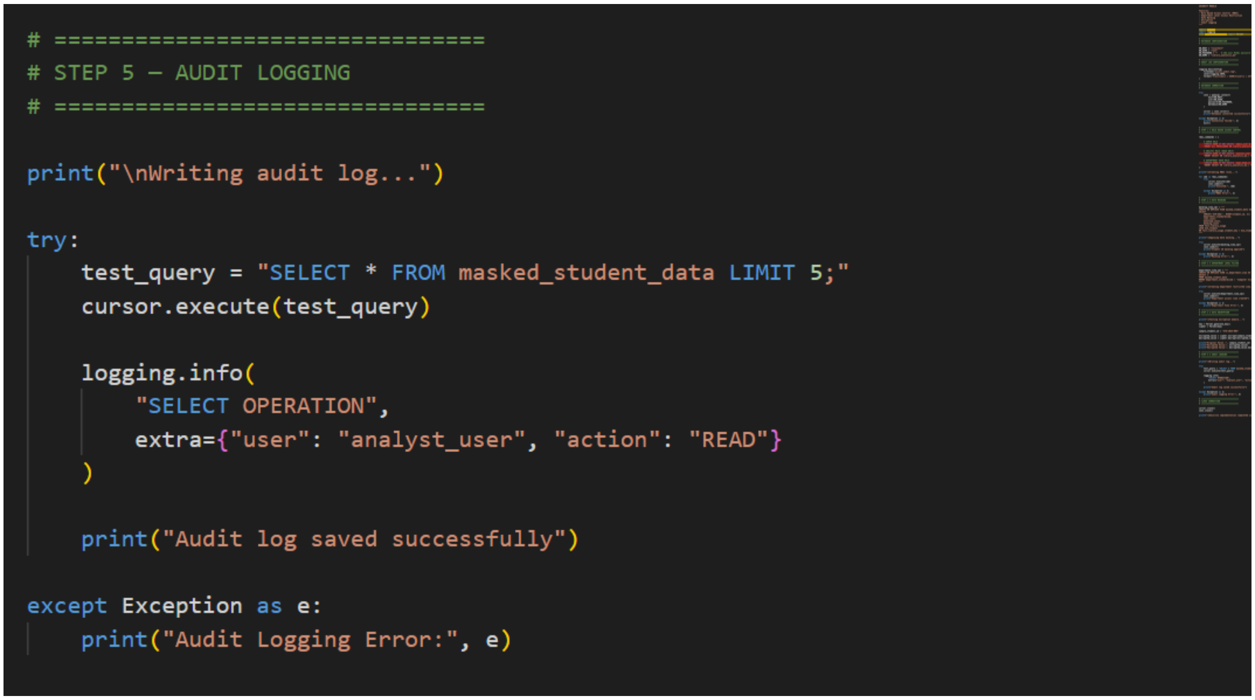
```
PS C:\Users\user\Desktop\Group03_LibraryDW> python security_rbac.py
Creating department restricted view...
Department access view created

Testing encryption module...
Creating department restricted view...
Department access view created

Testing encryption module...
Department access view created

Testing encryption module...

Testing encryption module...
Testing encryption module...
Original Value : STU-2024-003
Encrypted Value: b'gAAAAABpg2IzjcSforB0KrfdqlH4ezBSx53b2Ft_Ox3BdmXSqVf7vbESava9ZsboejP-871E6RYzM1TaG0EYnN
qYAunpsYL8hw=='
Encrypted Value: b'gAAAAABpg2IzjcSforB0KrfdqlH4ezBSx53b2Ft_Ox3BdmXSqVf7vbESava9ZsboejP-871E6RYzM1TaG0EYnN
qYAunpsYL8hw=='
Decrypted Value: STU-2024-003
qYAunpsYL8hw=='
Decrypted Value: STU-2024-003
Decrypted Value: STU-2024-003
```

Figure 29:output of Encryption

Audit Logging

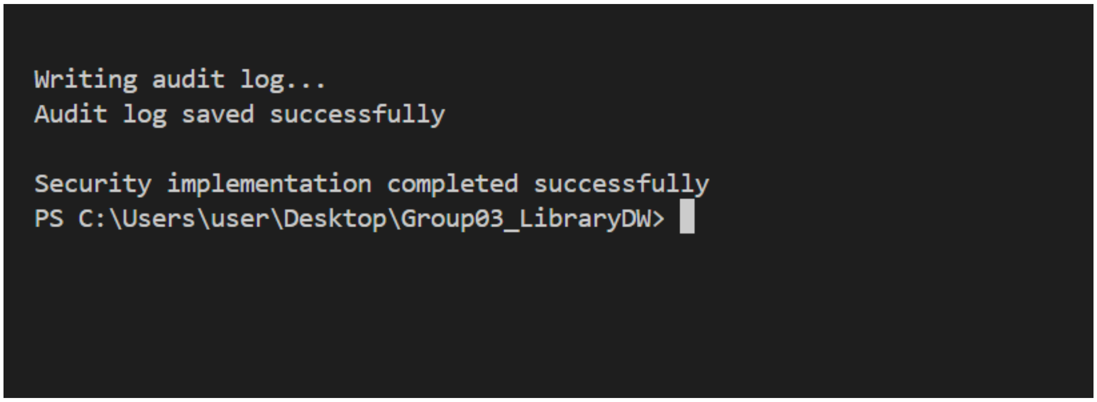
Database logs were enabled to:

- Track who accessed data
- Monitor query execution
- Detect suspicious activities



```
# ================================
# STEP 5 — AUDIT LOGGING
# ================================

print("\nWriting audit log...")

try:
    test_query = "SELECT * FROM masked_student_data LIMIT 5;"
    cursor.execute(test_query)

    logging.info(
        "SELECT OPERATION",
        extra={"user": "analyst_user", "action": "READ"}
    )

    print("Audit log saved successfully")

except Exception as e:
    print("Audit Logging Error:", e)
```

Figure 30:Audit Logging



```
Writing audit log...
Audit log saved successfully

Security implementation completed successfully
PS C:\Users\user\Desktop\Group03_LibraryDW> 
```

Figure 31:output of Audit Logging

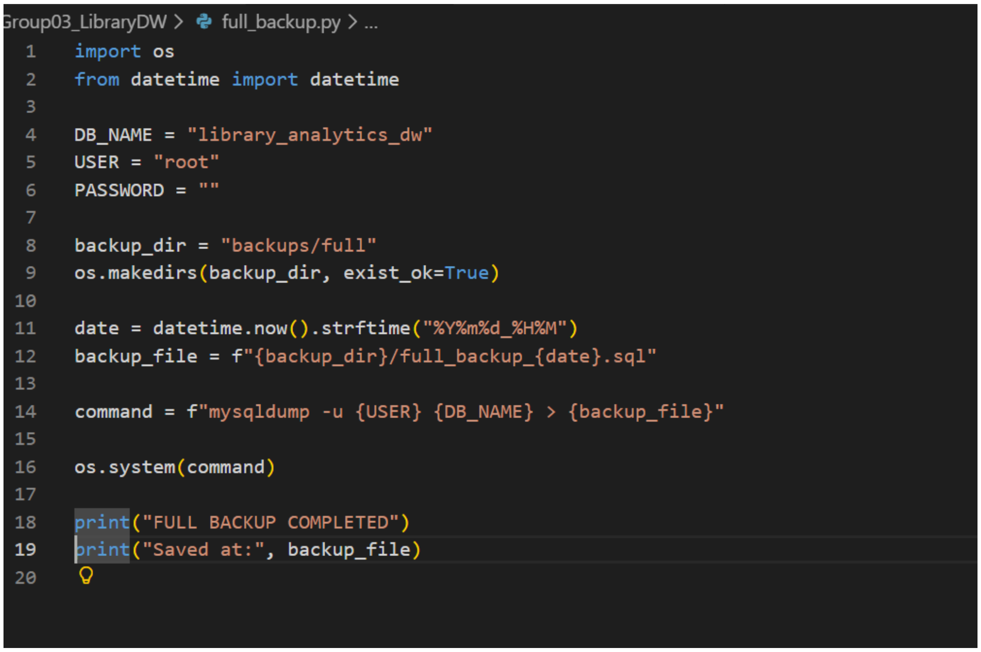
6.12 Backup and Recovery Strategy

To prevent data loss, a backup and recovery plan was designed.

Backup Types

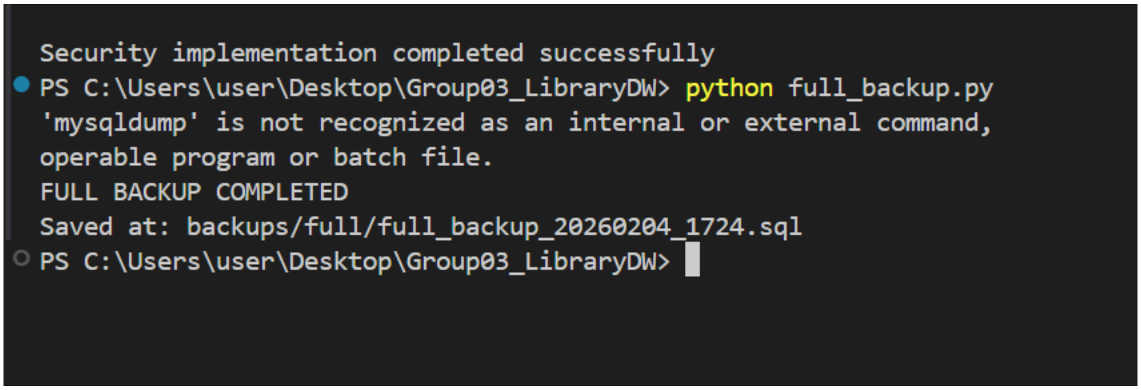
Full Backup

- Weekly full database backup.



```python
import os
from datetime import datetime

DB_NAME = "library_analytics_dw"
USER = "root"
PASSWORD = ""

backup_dir = "backups/full"
os.makedirs(backup_dir, exist_ok=True)

date = datetime.now().strftime("%Y%m%d_%H%M")
backup_file = f"{backup_dir}/full_backup_{date}.sql"

command = f"mysqldump -u {USER} {DB_NAME} > {backup_file}"

os.system(command)

print("FULL BACKUP COMPLETED")
print("Saved at:", backup_file)
```

Figure 32:full backup

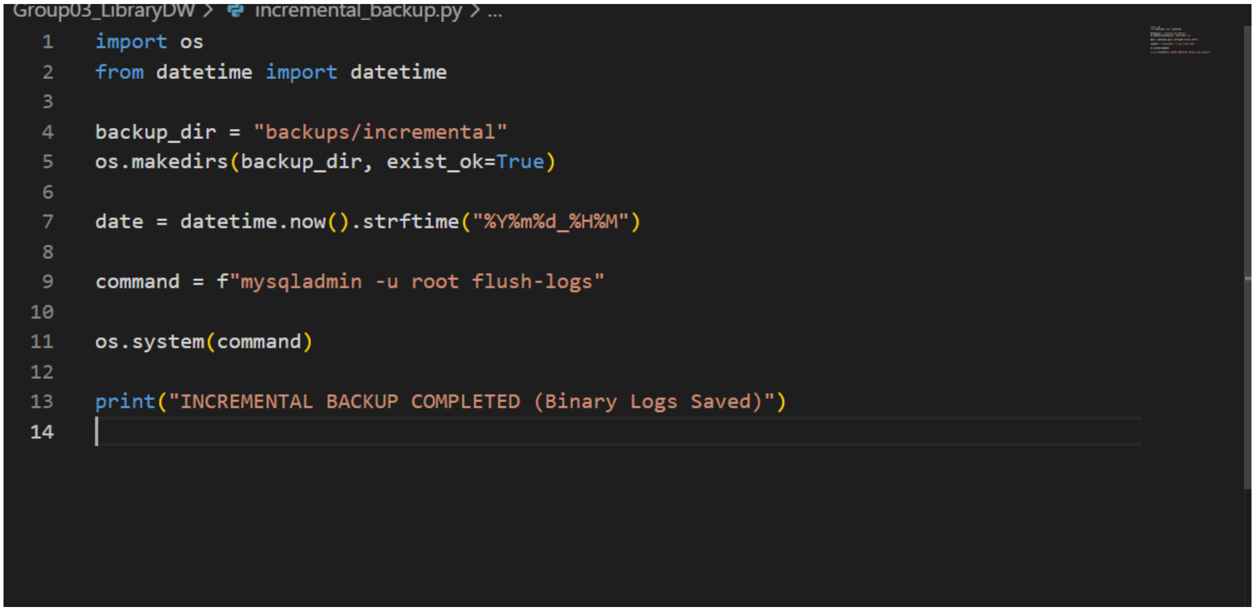


```
Security implementation completed successfully
PS C:\Users\user\Desktop\Group03_LibraryDW> python full_backup.py
'mysqldump' is not recognized as an internal or external command,
operable program or batch file.
FULL BACKUP COMPLETED
Saved at: backups/full/full_backup_20260204_1724.sql
PS C:\Users\user\Desktop\Group03_LibraryDW>
```

Figure 33:output full backup

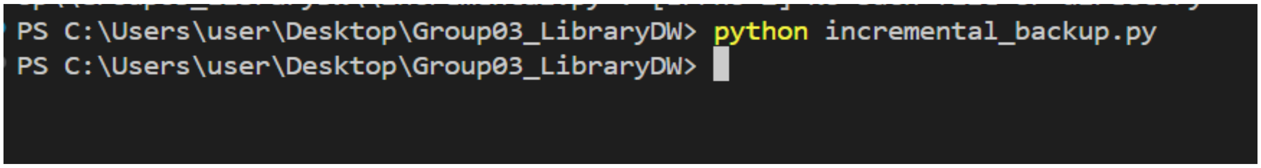
Incremental Backup

- Daily backup of changed data.



```python
import os
from datetime import datetime

backup_dir = "backups/incremental"
os.makedirs(backup_dir, exist_ok=True)

date = datetime.now().strftime("%Y%m%d_%H%M")

command = f"mysqladmin -u root flush-logs"

os.system(command)

print("INCREMENTAL BACKUP COMPLETED (Binary Logs Saved)")

```

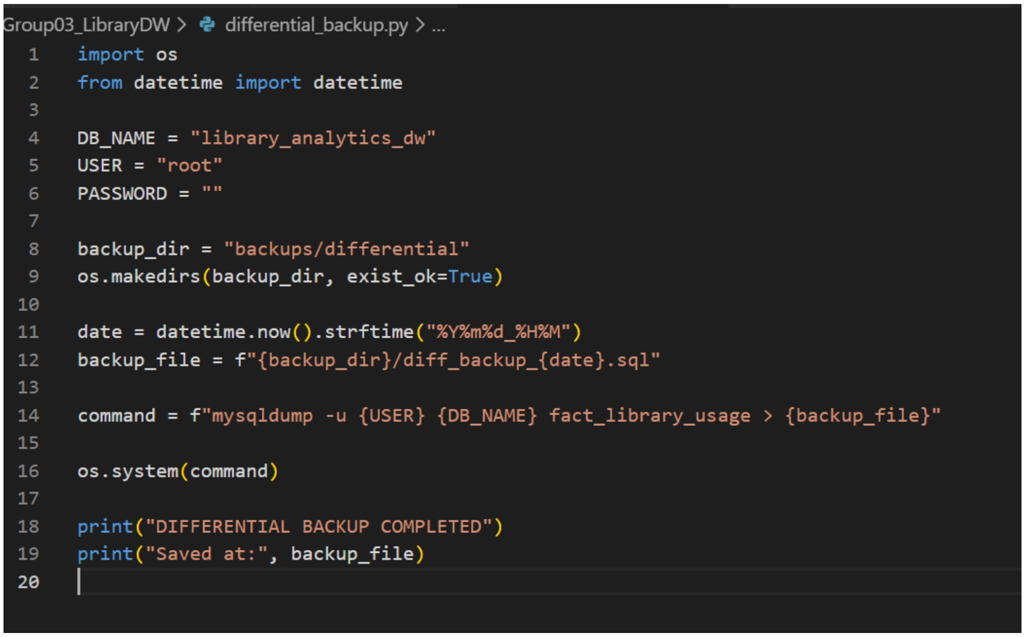
Figure 34:Incremental Backup



```
PS C:\Users\user\Desktop\Group03_LibraryDW> python incremental_backup.py
PS C:\Users\user\Desktop\Group03_LibraryDW>
```

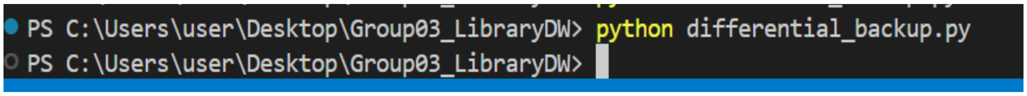
Figure 35:output of Incremental Backup

Differential Backup



```python
import os
from datetime import datetime

DB_NAME = "library_analytics_dw"
USER = "root"
PASSWORD = ""

backup_dir = "backups/differential"
os.makedirs(backup_dir, exist_ok=True)

date = datetime.now().strftime("%Y%m%d_%H%M")
backup_file = f"{backup_dir}/diff_backup_{date}.sql"

command = f"mysqldump -u {USER} {DB_NAME} fact_library_usage > {backup_file}"

os.system(command)

print("DIFFERENTIAL BACKUP COMPLETED")
print("Saved at:", backup_file)

```

Figure 36:Differential Backup

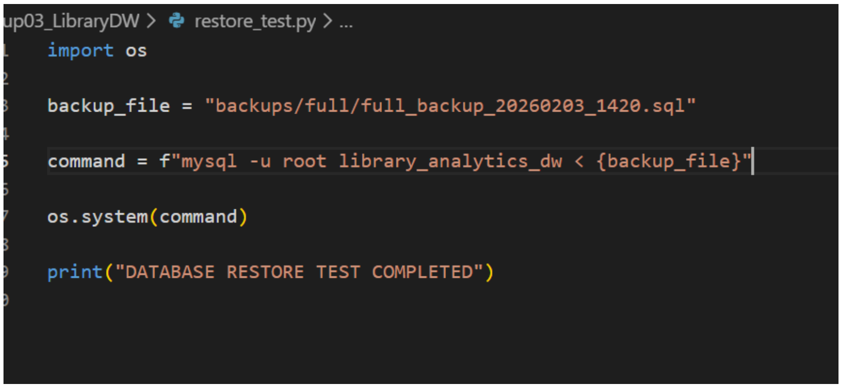


```
PS C:\Users\user\Desktop\Group03_LibraryDW> python differential_backup.py
PS C:\Users\user\Desktop\Group03_LibraryDW>
```

Figure 37: output Differential Backup

Testing

Backup restoration was tested to ensure data can be recovered successfully.



```
import os

backup_file = "backups/full/full_backup_20260203_1420.sql"

command = f"mysql -u root library_analytics_dw < {backup_file}"

os.system(command)

print("DATABASE RESTORE TEST COMPLETED")
```

Figure 38:testing

7 Conclusion

The University Library Analytics project successfully centralized fragmented library data into a unified Data Warehouse, enabling accurate, timely, and reliable reporting.

Through well-designed architecture, robust ETL processes, and a dimensional data model, data quality and consistency were significantly improved.

OLAP operations and analytical queries provided deep insights into usage trends, department performance, and resource utilization.

Interactive dashboards empowered management, departments, and staff to make informed strategic and operational decisions.

Overall, the system enhanced efficiency, data security, and scalability, supporting smarter, data-driven library management.