DESIGN AND IMPLEMENTATION OF UNIVERSITY DATA WAREHOUSE

Presented By

Priyanka Motwani

Vinay Kiran Manjunath

Project Group: A

Instructor: Dr. C.I. Ezeife

Emerging Non-Traditional Database Systems: Data Warehousing and Mining (COMP-8390)

Winter 2020

School of Computer Science

University of Windsor

DESIGN AND IMPLEMENTATION OF UNIVERSITY DATA WAREHOUSE

Table of Contents

Abstract

Data warehousing is a concept used to describe the methodology and techniques used to extract, transform and load data from multiple sources and provide detailed analysis of the resulting data. A common representation of the data in the data warehouse is the multidimensional approach using the facts and dimensions tables that an OLAP approach can design and implement. The facts refer to the specific values of interest in the analysis while the dimensions are used to provide a meaning to the facts and are built up within hierarchies by multiple levels. One of the biggest concerns of the universities so far is the management of huge amount of data that is gathered with the increasing number of students. Moreover, managing the information of faculties, their respective departments and their growth rates across a global level is a challenging task. This project report focuses on an application which considers source databases of universities built in different relational database management systems for three modules namely student, faculty and course and builds a data warehouse for the same. The data warehouse stores the historical data and with the help of facts and dimension tables, the important information can be retrieved using database views and GUI. Database schema is generated with the web page given to the mining tool. Data types are chosen with the analysis of the values for an attribute. The input to the data warehouse will be the source databases of student, faculty and course modules along with their relationship tables. The ETL will then convert these into facts and dimension tables for data warehouse. Lastly, the output will be the results of queries that are retrieved based on the views on the Web page

Acknowledgment

We would like to express our gratitude and appreciation to Dr. C.I. Ezeife for the guidance, support and motivation.

Introduction

Database is a structured collection of detailed records, or data, usually stored in a computer system. A database is normally managed by a DBMS (Database Management System). Together, the data and the DBMS are referred to as a database system along with the applications associated with them, often simplified to just a database. Data within the most common types of databases in use today is usually modelled in a series of tables in rows and columns to make processing and querying data efficient. They can then easily access, manage, change, update, monitor and organize the data. Most databases use the Structured Query (SQL) language to write and query data.

Data Warehouse is a central data repository that can be analyzed to make more informed decisions. Data flows from transactional networks, relational databases, and other sources into a datacenter, usually at daily cadences. Business analysts, data scientists and decision- use business intelligence (BI) software, SQL clients and other analytics applications to access the data. A data warehouse architecture consists of three tiers. The bottom tier of the architecture is the database server, where data is loaded and stored. The middle tier consists of the analytics engine that is used to access and analyze the data. The top tier is the front-end client that presents results through reporting, analysis, and data mining tools. A data warehouse adapts OLAP (Online Analytical Processing) technique for decision support systems and it stores and accesses the data into a data cube. A fact table contains the measurements/facts and foreign key to dimension tables. Dimension tables are denormalized tables containing descriptive information of a fact.

ETL (Extract, Transform and Load) is a process that extracts the data from different source systems, then transforms the data (like applying calculations, concatenations, etc.) and finally loads the data into the Data Warehouse system. ETL provides a method of moving the data from various sources into a data warehouse. As data sources change, the Data Warehouse will automatically update. ETL helps to migrate data into a Data Warehouse and convert it to the various formats and types to adhere to one consistent system.

This project aims to integrate all the above-mentioned concepts and build a **University Data Warehouse** that contains different modules like student, faculty and course. It is basically a combination of operational databases of universities that have their data sources built in different relational database systems. It uses MySQL and MSSQL as two different RDBMS for different universities to keep the student, faculty and course information and includes these in each of the

source databases. Data Warehouse answers multiple queries related to modules which a single source database cannot answer. ETL helps in cleaning and combining the data from source databases and a GUI is given to provide a user-friendly environment for different views and queries. Moreover, evaluations have been done and the results show a fairly good performance. This report provides an overview of the tools and extends the structured automatic data extraction technique. It includes the studies about the idea of modeling the data warehouse and learning multiple techniques for obtaining better results.

Problem Statement

In this millennial era where education is the most important factor, the universities across the countries need a well-managed system for information storage and useful data extraction. With the increasing number of students getting enrolled in these universities, a huge amount of data is gathered that contains their personal and educational information. Moreover, the need to accumulate the data of the faculties that teach multiple courses to the students along with their personal information is one another task itself. Additionally, to retrieve any information related to this data stored is one good job. The data warehouse helps in solving such scenarios where all the data from different data sources that have any formats can be stored in one single central repository in one single format. The data warehouse schema can help in retrieving historical and current data of any student and faculties regarding their courses, gpa, rating and much more. Moreover, handling different source operational databases on a large scale and global level for multiple universities and extracting updated information from a data warehouse helps in quick decisions. Different source data may be operating on different hardware and software platforms. It is the responsibility of the transformation tool (called extraction, transformation and Loading or ETL tool) to convert data on these platforms to a uniform platform where the data warehouse or operational data are stored. So, in order to achieve these goals, there is a need to design and implement a University Data Warehouse that can give a proper structure to this huge amount of data and store it on a central repository which contains historical and current information of all the operational source databases and answer all the queries with a user friendly interface.

Purpose of the application

This application serves to implement a warehouse for two different universities. It uses the concept of databases, data warehousing along with an ETL process to combine the data from multiple

source databases. Lastly, GUI is integrated for the user-friendly environment and for viewing the results of the queries.

The student, faculty and course modules are built in MySQL databases for two different universities. These modules are built for two operational source databases and then the data warehouse is built in MSSQL which combines the data from the two sources. The data warehouse is built in Star Schema for Student and Snowflake Schema for faculty where there are two fact tables for student and faculty and the dimension tables contain the information of the fact tables.

This application creates a schema by analyzing the structure and the content which is related to the universities for now. This application is created after the analysis of different university web sites and their structures used to populate the information about the products in their web sites for users' access. The study revealed the fact that almost all the university web sites provide the information about the students and the faculties in the form of a systematic process including collection of data, cleaning it, processing it, storing it and displaying the results of fired queries on an appropriate user –friendly environment.

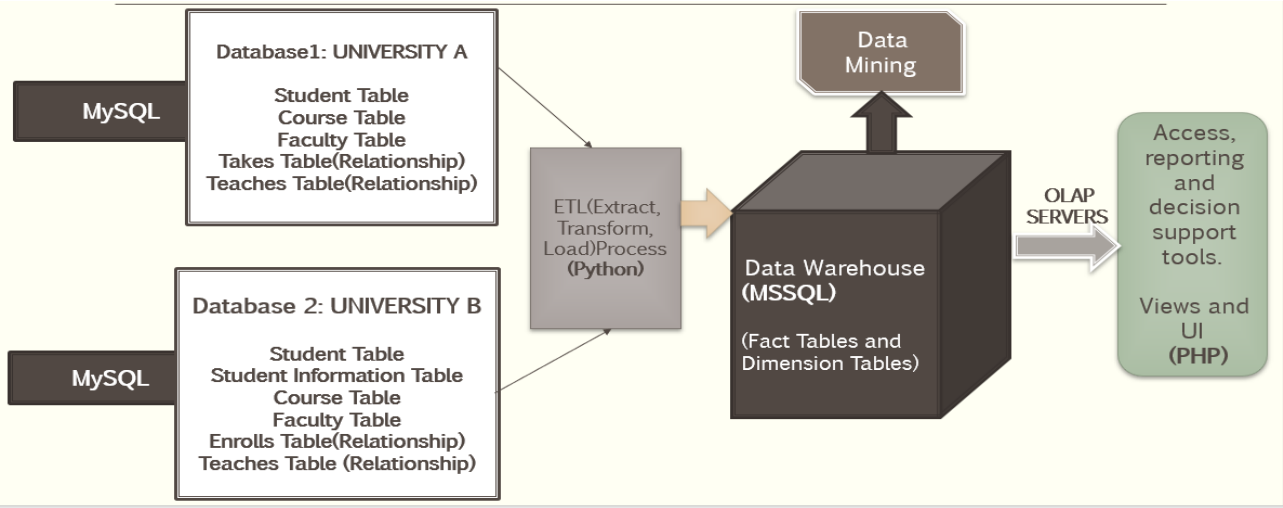


Figure 1:University Data Warehouse System Architecture

Implemented Modules

I. Student Module

The student module for both the universities contain the student information. It has student table for each university in different databases (MySQL). The attributes in both the tables of the databases differ in the schema, data types and syntax having null and not null values. The student

table in each database is connected to the course module by a relationship table which will link the constraints in both the modules of the two databases. These two source databases will be combined into the warehouse (MSSQL) and student fact table and its corresponding dimension tables are prepared for extracting information and running multiple queries in the views which can't be answered with single source databases.

II. Course Module

The course module for both the universities contain the course information. It has course table for each university in different databases (MySQL). The attributes in both the tables of the databases differ in the schema, data types and syntax having null and not null values. The course table in each database is connected to the student and faculty modules by a relationship tables which will link the constraints in all the modules of the two databases. These two source databases will be combined into the warehouse (MSSQL) and included in both the fact tables of student and faculty modules and its corresponding dimension tables are prepared for extracting information and running multiple queries in the views which can't be answered with single source databases.

III. Faculty Module

The faculty module for both the universities contain the faculty information. It has faculty table for each university in different databases (MySQL). The attributes in both the tables of the databases differ in the schema, data types and syntax having null and not null values. The faculty table in each database is connected to the course module by a relationship table which will link the constraints in both the modules of the two databases. These two source databases will be combined into the warehouse (MSSQL) and faculty fact table and its corresponding dimension tables are prepared for extracting information and running multiple queries in the views which can't be answered with single source databases.

IV. Relationships between the modules

Student - Course via *Enroll/ Takes* Relationship:

Student table is linked to course table through the "enrolls/studies" relationship table in both the source databases with all the foreign key constraints that can be the primary keys or the table attributes in student and course tables. The student fact table in the warehouse contains the attributes from all the tables including student, course and their relationship

tables along with certain other attributes (e.g. historical information of previous students) required to extract information that source databases can't answer single handedly.

Faculty - Course via *Teaches* Relationship:

Faculty table is linked to course table through the "teaches" relationship table in both the source databases with all the foreign key constraints that can be the primary keys or the table attributes in faculty and course tables. The faculty fact table in the warehouse contains the necessary attributes from all the tables including faculty, course and their relationship tables along with certain other attributes (e.g. Faculty evaluation over years) required to extract information that source databases can't answer alone.

Database and Data Warehouse Schema

The schema diagrams of source databases and data warehouse for all three modules (student, faculty and course) are completely implemented in MSSQL for visualization purpose.

A. Student Schema for databases and data warehouse

Source database 1: UWindsor Schema

Student (Sid, Sname, major, gpa)

Course (Cid, Ctitle, Credithr)

Takes (Sid, Cid, grade) - Relationship Table

Source Database: - UWindsor

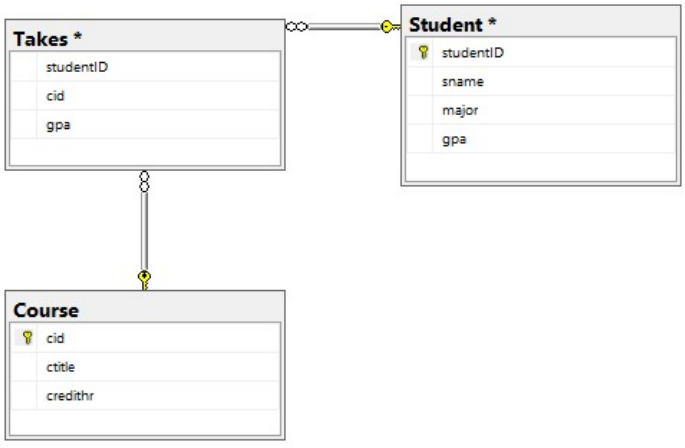


Figure 2: Source Database for UWindsor (Student)

Source database 2:

Student (Stuid, gpa)

Studentinfo (Stuid, Stuname, dept)

Course (Cid, Cname, Credithr)

Enrolls (Stuid, Cid, Grade) - Relationship Table

Source Database: - Utoronto

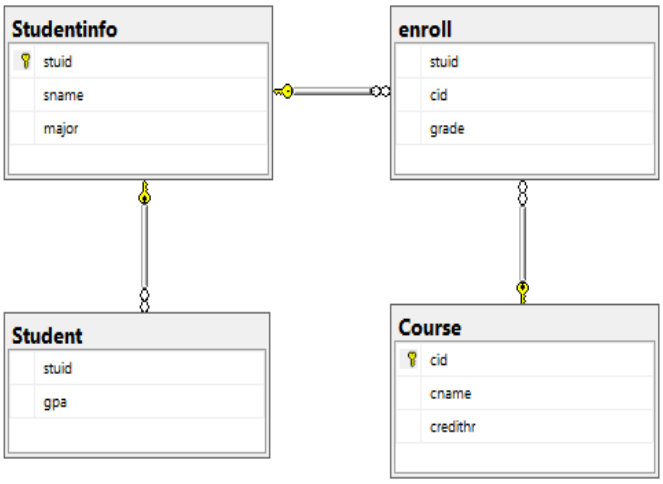


Figure 3: Source Database for UToronto (Student)

Data Warehouse Star Schema for Student and Course Module

Fact Table:

Student_fact (student_id, cid, univID, termID, gpa)

Dimension Tables:

Student (Sid, Sname, Smajor)

Course (Cid, Ctitle, Credithr)

Terminfo (Termid, Term, Year)

University (Univ_id, uname, location)

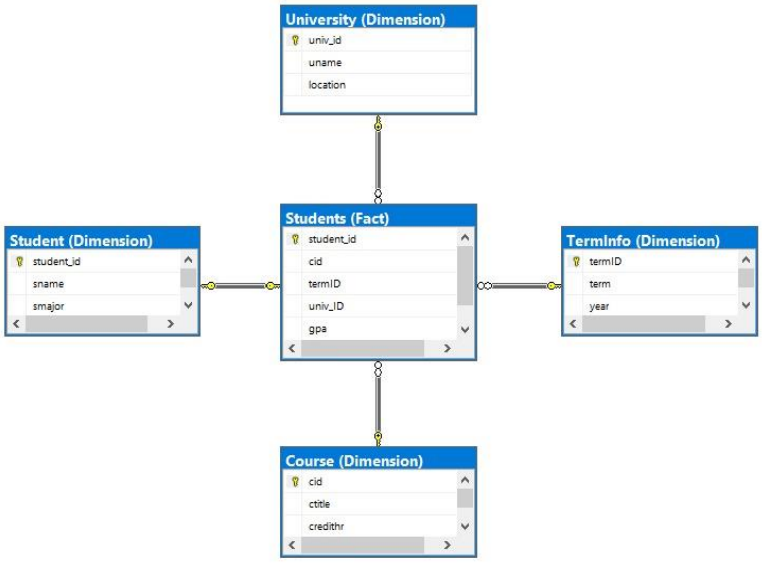


Figure 4: Student Data Warehouse Star Schema

B. Faculty Schema for databases and data warehouse

Source database 1: UWindsor Schema

Instructor (fac_id, fac_name, fac_dept_name, fac_research_area, fac_term, fac_rating)

Course (C_id, C_name, C_credithr)

Teaches (fac_id, C_id, Rating) - Relationship Table

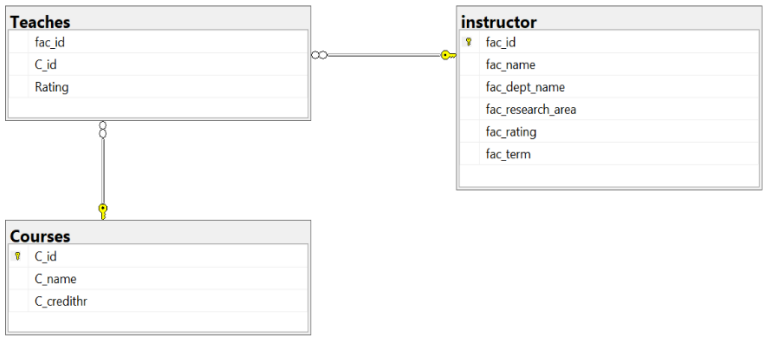


Figure 5:Source Database for UWindsor (Faculty)

Source database 2: UToronto Schema

Faculty (f_id, f_name, f_dept_name, f_research_area, f_term, f_rating)

Course (C_id, C_name, C_cr_hr)

Teaches (f_id, C_id, Rating) - Relationship Table

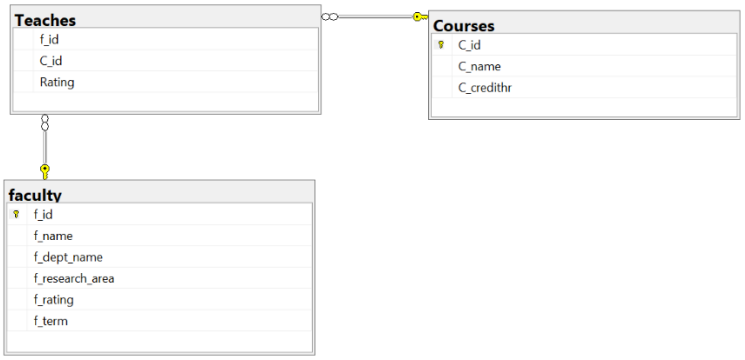


Figure 6: Source Database for UToronto (Faculty)

Data Warehouse Star Schema for Faculty and Course Module

Fact Table:

Fact_faculty(Fac_id, Univ_id, Term_id, Course_id, Rating)

Dimension Tables:

Dim_Faculty (Fac_id, Fac_name, Fac_Dept, Fac_Research_Area, Sem_id)

Course (Course_id, Course_name, Course_hr)

Terminfo (Termid, Term, Year)

University (Univ_id, uname, location)

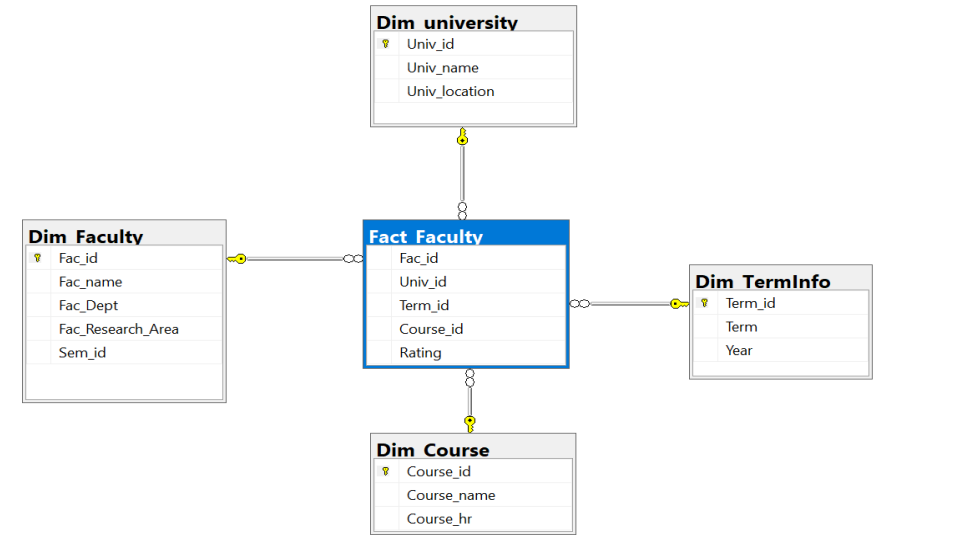


Figure 7: Faculty Data Warehouse Star Schema

Tools and Software:

The entire project is built on Linux operating system on Ubuntu v18.0.4. We have used MySQL v5.7.29 for building the two source databases. For data warehouse, Microsoft SQL Server (MSSQL) -2019 v15.0.4033.1 is used. For ETL process, we have used Python v3.7 and also imported “pyodbc” and “mysql.connector” for database connectivity. The GUI website of the project is built in PHP v7.5 with all the modules of databases, ETL modules and data warehouse integration. The final output of the views and queries is displayed on the website page.

Screenshots of the Implemented System

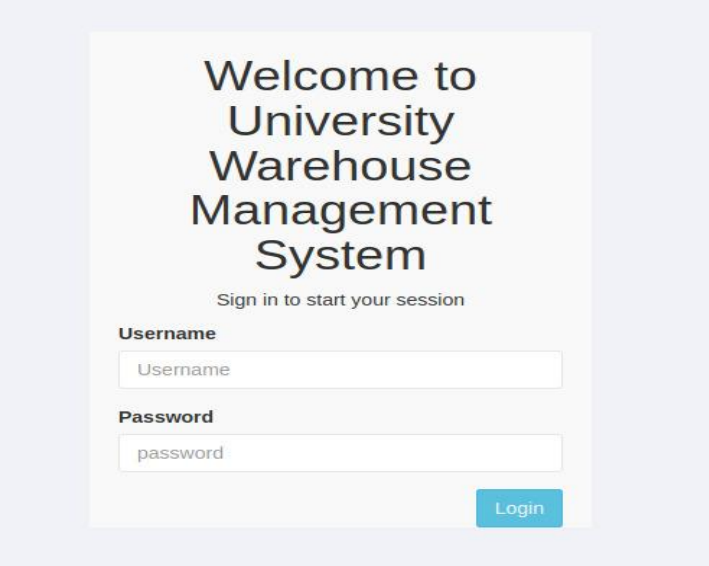


Figure 8: Welcome Screen Page

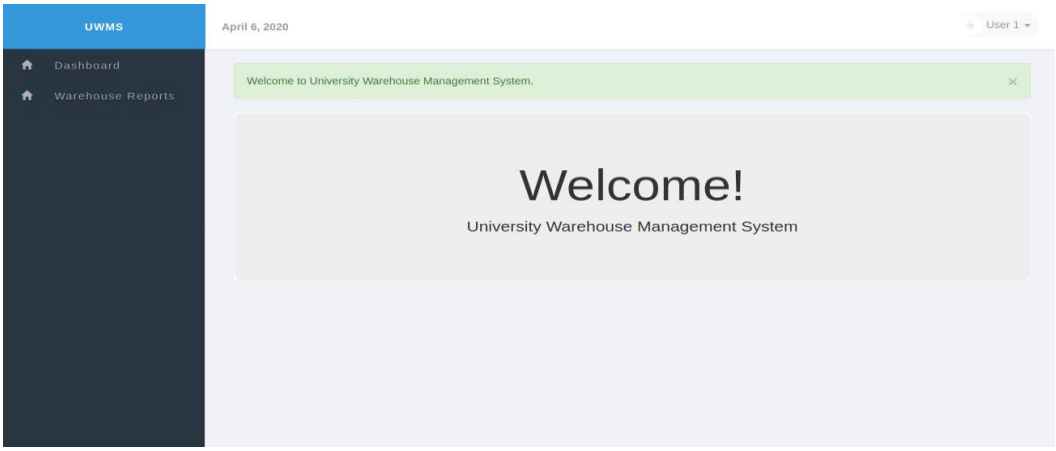


Figure 9: User Screen Page where only Warehouse Reports are accessible.

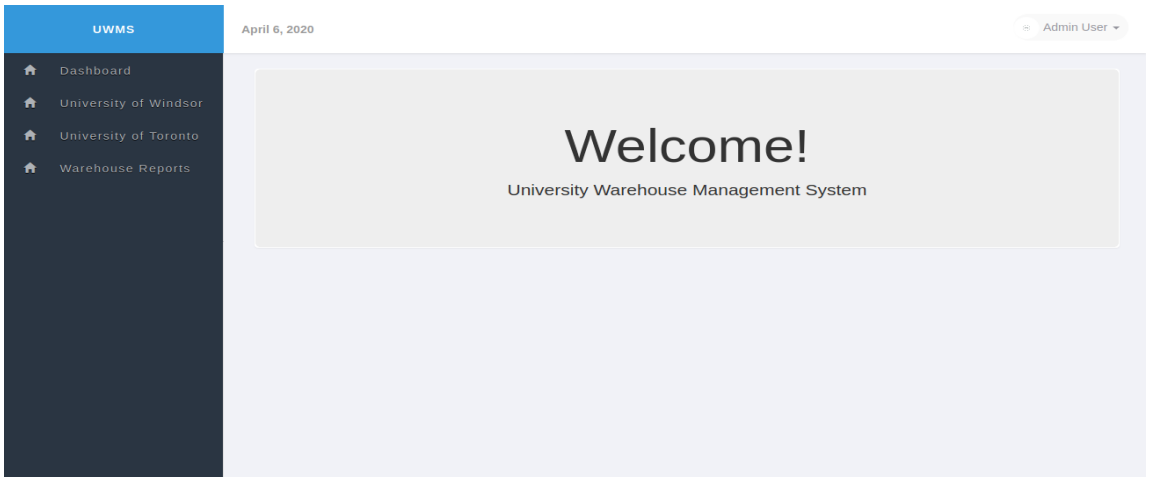


Figure 10: Admin user screen page where Source databases and Warehouse Reports are accessible

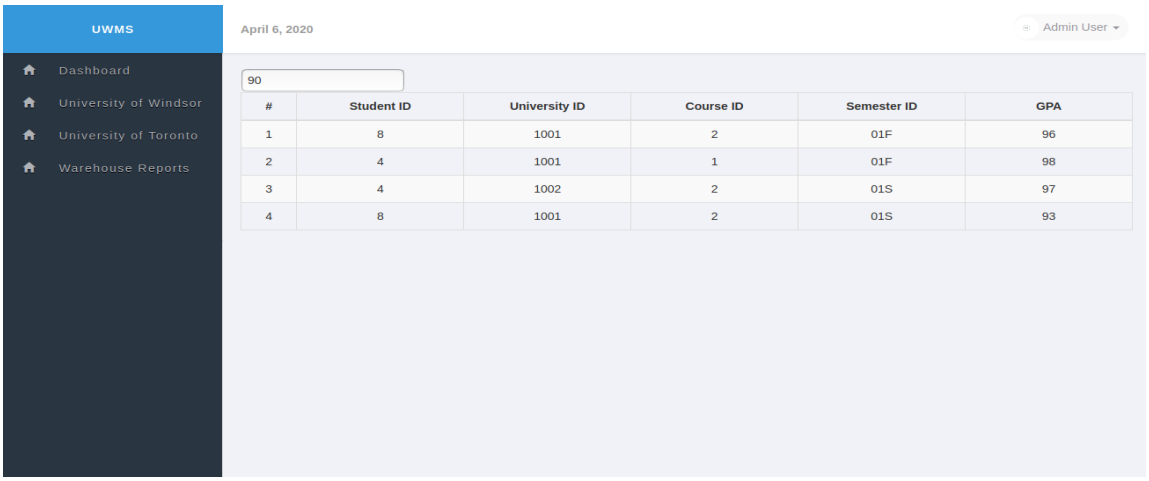


Figure 11: A query which returns studentID, UniversityID, CourseID, SemesterID whose GPA > 90

Table 1 – Grade Distribution Table

Course	Semester	A	B	C	D	F	W
MATH0700	00F	53	69	23	14	8	9
	01S	48	58	32	12	11	3
	01U	24	10	12	4	3	1
	01F	67	72	51	10	17	14
MATH0800	00F	30	44	26	12	5	6
	01S	46	24	13	16	7	4
	01U	13	25	27	3	8	3
	01F	38	51	35	14	6	15
ENGL1100	00F	63	38	19	3	4	5
	01S	41	12	28	15	2	7
	01U	24	6	7	4	3	2
	01F	74	22	27	9	6	3

Figure 12: Give me all the students who have scored less than 75 in courses MATH0700, MATH0800, ENGL1100 in the last 4 terms.

Student ID	Student Name	Major	University ID	Course ID	Semester ID	GPA
1	Rivers, Seth D.	MAC	U001	COMP 8100	2019F	81
4	Charles, Hilda E.	CS	U001	COMP 8100	2019F	85
7	Thomas, Hilel X.	MAC	U001	COMP 8100	2019F	87
2	Jefferson, Jessamine P.	MAC	U001	COMP 8100	2019F	92
5	Mayer, Jamal G.	CS	U001	COMP 8100	2019F	95

Figure 13: Give me Top (5) student names and their ID who have taken Literature Survey course in the Fall term whose GPA > 80 order by ascending

Conclusion

This project implements a University Data Warehouse for student, course and faculty module and integrates data from multiple source databases. It is considering data from two different universities and two different DBMS systems like MySQL for source databases and MSSQL for data warehouse. Since the data used here is structural, ETL process is used in order to move the data from source to target. The warehouse helps in keeping the historical and current data maintained in a central repository and answers multiple queries related student's gpa or faculty's rating calculation, etc. from the views. GUI has been built for user friendly environment and for searching the data warehouse while firing the queries or calling the views in the back end. The project aims to suffice fair requirements of a university warehouse and can be used for two different universities.

Future Scope

This project can be extended to combine the databases from more than two universities or with multiple other databases or a combination of both. Right now, it is built for three modules namely

student, course and faculty but can also be extended to several other modules including library, canteen, human resources, cashiers. Moreover, the project can be extended to work as a recommendation system (e.g. recommendation of new courses for further semester based on previously selected courses by the student) with frequent patterns observed.

Contributions

Source Database Creation:

- SDB1 – UWindsor – Priyanka Motwani
- SDB2 – UToronto – Vinay Kiran Manjunath
- Triggers – Worked by both.

Data Warehouse Creation:

- Dimension and Fact Tables Creation with Referential Integrity Constraints – Vinay Kiran Manjunath
- Viewing and Indexing – Priyanka Motwani

ETL Design:

- Extraction of data from Source Database – Priyanka Motwani
- Loading the data into Warehouse – Vinay Kiran Manjunath

Web Application Design:

- Admin User Page – Vinay Kiran Manjunath
- Normal User – Priyanka Motwani
- Login Page and Results of the Query – Worked by both

SQL queries: Worked by both

User Manual

First, for the backend, we start with building the student, course and faculty database tables along with the relationship tables like enrolls and teaches in MySQL databases for two different universities. Then we create the triggers on the different modules. After that the data warehouse is created that includes the facts and the dimension tables. ETL cumulates the process of two data

sources and then the warehouse views are created to query and get the results. Lastly, the frontend is created in php for user friendly environment and information retrieval.

Certain examples (student module) have been shown below in the steps how the tasks should be done and then in the end the project installation guide provides how to download and run the project.

1. **Database Creation**

   Following is the schema, that can be referred to create tables in MySQL source database:

Student

```
CREATE TABLE Student(
    stuID int NOT NULL,
    gpa INT,
    PRIMARY KEY (stuID)
);
```

Student Info

```
CREATE TABLE StudentInfo(
    stuID int NOT NULL,
    stuName varchar(255),
    gpa int not null,
    dept varchar(255) not null,
    FOREIGN KEY (stuID) REFERENCES Student(stuID)
);
```

Course

```
CREATE TABLE Course(
    cid int NOT NULL,
    cname varchar(255),
    Credithr int not null,
    PRIMARY KEY (cid)
);
```

Enroll

```
CREATE TABLE Enroll(
    stuID int not null,
    cid int not null,
    Grade int not null,
    FOREIGN KEY (stuID) REFERENCES Student(stuID),
    FOREIGN KEY (cid) REFERENCES Course(cid)
);
```

Once all the schemas are created, populate them with data.

2. **Database Trigger Creation:**

SQL trigger is a stored program that invoked by the database automatically when any change in the event occurs. This trigger is created in the student module as soon as a student's gpa is updated. When a student data will be inserted at the first time of admission, the grade won't be entered and as soon as the student is enrolled in a course and gets its grade, this trigger will update that student's gpa in the Student table.

Trigger created in the MySQL database:

```
CREATE TRIGGER [dbo].[Student_gpa] ON [dbo].[Takes]
AFTER INSERT
AS
declare @newGPA as INT;
set @newGPA = (SELECT AVG (I.gpa) from INSERTED I group by (I.s_id));
BEGIN
    UPDATE s
    SET s.gpa = @newGPA
    FROM dbo.Student s
    INNER JOIN Inserted i ON s.s_id = i.s_id
END
```

3. **Data Warehouse Creation from the source databases:**

Fact tables and Dimension tables are created for the data warehouse in MSSQL that combines both the source databases.

Following are the operations that can be done on Student Fact Table:

```
CREATE TABLE [Fact].[Students](

[student_id] [int] NOT NULL,

[cid] [int] NOT NULL,

[termID] [int] NOT NULL,

[univ_ID] [int] NOT NULL,

[gpa] [varchar](100) NOT NULL,

 CONSTRAINT [PK_Student] PRIMARY KEY CLUSTERED

([student_id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
 OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [Fact].[Students]  WITH CHECK ADD  CONSTRAINT [FK_Fact_Student_cid_Dimension_Course] FOREIGN KEY([cid])

REFERENCES [Dimension].[Course] ([cid])

GO

ALTER TABLE [Fact].[Students] CHECK CONSTRAINT [FK_Fact_Student_cid_Dimension_Course]

GO

ALTER TABLE [Fact].[Students]  WITH CHECK ADD  CONSTRAINT [FK_Fact_Student_student_id_Dimension_Student] FOREIGN KEY([student_id])

REFERENCES [Dimension].[Student] ([student_id])

GO
```

Following is the schema for Student Dimension Table

```
CREATE TABLE [Dimension].[Student](

[student_id] [int] NOT NULL,

[sname] [varchar](1) NOT NULL,

[smajor] [varchar](1) NOT NULL,

 CONSTRAINT [PK_Student] PRIMARY KEY CLUSTERED

([student_id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO
```

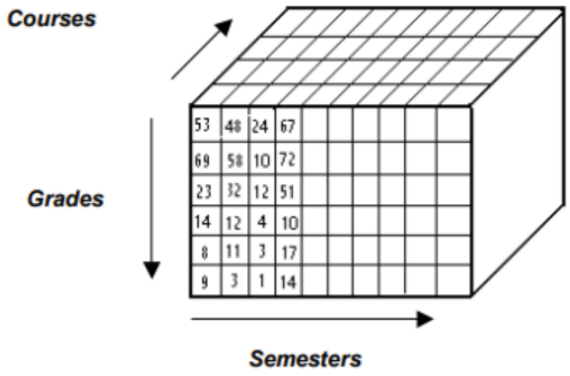
4. ETL process:

Extract, Transform and Load process to cumulate the data into the data warehouse from different source databases.

```
def etl(query, source_cnx, target_cnx):
  # extract data from source db
  source_cursor = source_cnx.cursor()
  source_cursor.execute(query.extract_query)
  data = source_cursor.fetchall()
  source_cursor.close()

  # load data into warehouse db
  if data:
    target_cursor = target_cnx.cursor()
    target_cursor.execute("USE {}".format(StudentDW))
    target_cursor.executemany(query.load_query, data)
    print('data loaded to warehouse db')
    target_cursor.close()
  else:
    print('data is empty')

def etl_process(queries, target_cnx, source_db_config, db_platform):
  # establish source db connection
  if db_platform == 'mysql':
    source_cnx = mysql.connector.connect(**source_db_config)
  elif db_platform == 'sqlserver':
    source_cnx = pyodbc.connect(**source_db_config)
  elif db_platform == 'firebird':
    source_cnx = fdb.connect(**source_db_config)
  else:
    return 'Error! unrecognised db platform'

  # loop through sql queries
  for query in queries:
    etl(query, source_cnx, target_cnx)

  # close the source db connection
  source_cnx.close()
```

5. **Data Warehouse View:**

It is a simple analysis of grade distributions for courses over several semesters. In contrast to the traditional normalized relational model used for databases, the dimensional model is denormalized for reporting speed, as it minimizes the number of table joins required to satisfy queries. OLAP cube is shown for visualization



6. **Installing the system for a naive user:**

   - Download the zip file named Website.

- Extract it to /var/www/html/ in Linux Operating System.
- You can find three .sql files named "MySQL1.sql", "MySQL2.sql", "MSSQLDW.sql" which is for creating source database and warehouse.
- Then in a web browser type: localhost/Website/index.php to get the web application and access the Warehouse as "admin user" or "user" roles.

References

[1] http://cezeife.myweb.cs.uwindsor.ca/courses/60-539/notes/539notes.pdf

[2] Bates, Emily, "UVM Big Data? Aggregating Campus Databases and Creating a Data Warehouse to Improve Student Retention Rates at the University of Vermont" (2015). UVM Honors College Senior Theses. 86. https://scholarworks.uvm.edu/hcoltheses/86

[3] Linkedin Learning – Implementing a data warehouse with Microsoft SQL server - https://www.linkedin.com/learning/implementing-a-data-warehouse-with-microsoft-sql-server-2012/physical-design-for-a-data-warehouse?u=56973065

[4] Create Login, User, assign Permission: SQL Server Tutorial - https://www.guru99.com/sql-server-create-user.html

[5] Walkthrough for the security features of SQL Server on Linux - https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-security-get-started?view=sql-server-ver15

[6] Building a Data Warehouse - With Examples in SQL Server | Vincent Rainardi | Apress - https://www.apress.com/us/book/9781590599310

[7] PHP Tutorials for beginners - https://www.w3resource.com/php/php-home.php

[8] A Quick Guide to Using the MySQL APT Repository - https://dev.mysql.com/doc/mysql-apt-repo-quick-guide/en/

[9] Using python script for data ETL - https://codeburst.io/using-python-script-for-data-etl-53138c567906

[10] Triggers In SQL Tutorial - https://appdividend.com/2019/06/24/sql-triggers-tutorial-with-example-triggers-in-sql/