Polystore Systems for Data Integration of Large Scale Astronomy Data Archives

A DISSERTATION

SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

MANOJ POUDEL

Graduate Department of Computer and Information Systems

The University of Aizu

2022

The thesis titled

Polystore Systems for Data Integration of Large Scale Astronomy Data Archives

by

Manoj Poudel

is reviewed and approved by

Chief referee

Professor

Yutaka Watanobe Yutaka Watanobe 2022/8/16

Professor

Incheon Paik Incheon Paik Aug. 22, 2022

Professor

Maxim Mozgovoy M. Mozgovoy 2022/8/22

Professor

Yoshiko Ogawa Yoshiko Ogawa 2022/8/19

Professor Emeritus

Subhash Bhalla Subhash Bhalla

The University of Aizu

2022

This thesis is dedicated to My Family

GRADUATE DEPARTMENT OF COMPUTER AND INFORMATION SYSTEMS

UNIVERSITY OF AIZU

Aizu Wakamatsu, Fukushima-ken, 965-8580, JAPAN

ACKNOWLEDGEMENT

I would like to thank my esteemed supervisor, Dr. Yutaka Watanobe for his invaluable supervision, support, and tutelage during my Ph.D. My gratitude extends to the University of Aizu for providing me with the opportunity to undertake my studies in the Department of Computer Science and Information Systems. Additionally, I would like to express my gratitude to Prof. Subhash Bhalla for his treasured support, which was influential in shaping my experimental methods and critiquing my results.

I would also like to thank my company (Software Research Associates, Inc. (SRA)) for their support in my study. I would also like to thank Dr. Maxim Mozgovoy for his technical support during my study. I also thank Dr. Shashank Shrestha and Dr. Rashmi P. Sarode for their support. Finally, I express my gratitude to my parents and my wife. Without their tremendous understanding and encouragement over the past few years, it would be impossible for me to complete my study.

Abstract

Over the past few decades, changes in model and data types have created difficulties in managing heterogeneous data. Various scientific data archives employ diverse methods to manage such data efficiently. Similar to many other scientific fields, astronomy has data archives containing vast quantities of data, diverse data models, and various data types. Images, texts, key-and-value pairs, and graphs comprise a vast amount of available data in the astronomical domain. Scalability, growth, and performance issues may arise when managing such data in a single database. It is important to manage heterogeneous open data on the Internet and develop a query language to combine web services and data repositories.

Polystores can aid in the integration of disparate heterogeneous data stores for information retrieval, data visualization, and the creation of useful web applications. This dissertation proposes a web-based query system based on the Polystore database architecture and attempts to provide a solution to expand the size of astronomical data. The proposed system based on Polystore directly unifies the querying of multiple datasets, eliminating the need to translate complex queries, and simplifying the work for astronomical domain users. This dissertation articulates and analyzes data integration models, and incorporates them into a system for managing linked open data provided by the astronomical domain. Data integration integrates data from various sources and presents a unified view of all sources. The proposed system is scalable, and its model can be applied to various other heterogeneous data management systems. Hence, we present a workflow web-based polystore system architecture based on a top-down rather than a bottom-up strategy that emphasizes language translation. Using a web-based query system, a technique for managing a local data store and connecting it to a remote cloud store was developed.

Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BigDAWG	Big Data Working Group
CCD	Charge – Coupled Device
CDS	Strasbourg astronomical Data Center
DB	Database
DBEXPID	Database Exposures ID
DBFID	Database Filters ID
DBMS	Database Management System
DBNID	Database Nights ID
DBPID	Database Processed Images ID
DBS	Database System
DW	Data Warehouse
ETL	Extract – Transform – Load
EXTASCID	Extensible system for Analyzing Scientific Data
FDBMS	Federated Database Management System
FDBS	Federated Database System
FITS	Flexible Image Transport System
FQA	Federated Query Agents
GLADE	Generalized Linear Aggregate Distributed Engine
GUI	Graphical User Interface
GVA	Global View to Application
HDFS	Hadoop Distributed File System
hk	Header Key
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
iPTF	Intermediate Palomar Transient Factory
IPAC	Infrared Processing and Analysis Center
IRE	Information Requirements Elicitation
IRSA	Infrared Science Archive
IRTF	Infrared Telescope Facility
IT	Information Technology
JSON	JavaScript Object Notation
KDD	Knowledge discovery in database
LAN	Local Area Network

LSST	Legacy Survey of Space and Time
LOD	Linked Open Data
MDBS	Multi – Database System
MIMIC	Multiparameter Intelligent Monitoring in Intensive Care
MIT	Massachusetts Institute of Technology
NASA	National Aeronautics and Space Administration
OBS-DATE	Observation Date
OLTP	Online Transaction Processing
PDW	Parallel Data Warehouse
PTF	Palomar Transient Factory
RDBMS	Relational Database Management System
RDF	Resource Description Framework
SDSS	Sloan Digital Sky Survey
SIMBAD	Set of Identification, Measurements, and Bibliography for Astronomical Data
SOFIA	Stratospheric Observatory for Infrared Astronomy
SPARQL	SPARQL Protocol and RDF Query Language
SMOKA	Subaru-Mitaka-Okayama-Kiso-Archive
SQL	Structured Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VCRO	Vera C. Rubin Observatory
ZTF	Zwickly Transient Facility
ZTFFIELD	ZTF Field ID
ZSDS	ZTF Science Data System

Chapter 1

INTRODUCTION

1.1 Introduction

Developments in the types of models and variety of data over the past few decades have given rise to issues in managing heterogeneous data. Data integration is a technique for merging data from different sources and offering a single perspective over all sources[6]. There are two types of models for data integration: physical and virtual models. Data ware-housing is a physical architecture in which data is compiled from multiple data repositories to form a new central database. A data warehouse is a subject-oriented, integrated, time-variable, and non-volatile collection of data to support management decision-making processes [7][8]. Other models include virtual ones. The fundamental components of the virtual architecture are Federated Database Systems (FDBS), Mediation, and the newly proposed Polystores. The Federated Database Management System (FDBMS) is self-sufficient database management system used for viewing and querying several other databases via APIs. In contrast, the mediator process query is specified against the federation's integration schema [9] [10]. The mediator stores metadata by detailing the integration schemes for each data resource in the federation. Using an API, the proposed system merges several databases that contain heterogeneous data. The use of API in different databases relates to the data-integration mediation paradigm [11].

1.2 Big Data Integration

With an emphasis on data exploration and discovery, big data integration has been presented as a novel method of conceiving data integration across massive, growing data sources. The need for data analysis drives integration; hence, data discovery was conducted to aid in this analysis. Data analysis requires the identification of data that correctly combines, aggregates, or joins existing data, a paradigm labelled query-driven data discovery [12]. Various databases and data-sets are stored in multiple scientific data repositories. It is laborious to unify searches across databases and data models.

1.2.1 Big Data in Astronomical Domain

Astronomy is the study of the physics, chemistry, and evolution of celestial objects and events outside Earth's atmosphere, such as supernovae explosions, gamma-ray bursts, and cosmic microwave background radiation [13]. Big data in the astronomical field primarily consists of four Vs: volume, variety, velocity, and value.

Volume refers to the amount of data stored. Terabytes, petabytes, and even exabytes of data exist. Capturing, curating, integrating, storing, processing, indexing, searching, sharing, transmitting, mining, analyzing, and visualizing massive amount of data presents obstacles. Traditional tools are incapable of handling such large amounts of data. Numerous ground- and space-based large-scale sky survey programs have generated a deluge of data in all areas of astronomy.

Variety indicates data complexity. The most common types of astronomical data include pictures, spectra, time-series, and simulation data. Most of the information is stored in catalogues or databases. The fact that numerous telescopes or projects have their own formats makes it challenging to integrate data from various sources during the analysis phase. Each data item typically includes thousands or more features, resulting in significant dimension issues. In addition, data may be structured, semi-structured, unstructured, or mixed.

Velocity refers to the rates of data production, transmission, and analysis. The LSST will produce one SDSS per night for a decade in terms of data volume. Batch, stream,

near-time, and real-time data analysis are necessary. The LSST anticipates the discovery of 1,000 new supernovae each night for a decade, implying that at least 10 to 100,000 alerts will be requested. Astronomers face a challenging problem in determining how to mine, correctly categorize, target supernovae candidates, and conduct follow-up observations within the next decade.

Value: The term value characterizes the high value of data in astronomy. Discovering strange, rare, unanticipated, and novel objects or events in astronomy is fascinating and motivating. Similarly, identifying a novel distribution pattern or law is of considerable significance.

The data rate and volume in optical time-domain astronomy are on the cusps of exponential development. By 2022, the data are anticipated to have been multiplied 300 times. A significant increase in detected sources will necessitate efficient and well-structured databases. To classify source types to manage these data, extremely efficient machine learning algorithms are necessary.

1.3 Motivation

A variety of data have their own language, so managing these data with a federated database management system (FDBMS) may result in inefficiency and poor performance. Therefore, we require a Polystore architecture that utilizes various types of data from many databases. Polystores facilitate uniform querying across many data models [14]. Polystores are required to manage information efficiently across several data models. It is a database management system (DBMS) comprising of heterogeneous database engines that communicate via an application programming interface (API) [15]. The primary objective of this study was to handle astronomical data using Polystore technology. These data sources are the Zwickly Transient Facility (ZTF) data sources [16].

1.4 Thesis Contributions

This dissertation presents two technical and conceptual contributions in the form of a Polystore system for maintaining large-scale astronomical data archives, and a workflow-based query system for visualizing and downloading astronomical images.

This thesis focuses on developing web-based query management tools and methodologies for open data from the astronomical domain, extracting data information from unstructured data and connecting available data to other datasets to enable the discovery of further data information.

With the ultimate objective of empirically validating our research hypothesis, the empirical contributions of this dissertation include the development of web-based Polystore applications and methodologies within a variety of case studies. The development phase comprises data downloading, cloud server connection, and API visualization.

1.5 Thesis Organization

This section provides an outline of the organization of the thesis. This thesis comprises four main chapters, each of which includes a discussion of related work and is tailored to the specific problem addressed in the corresponding chapter. This thesis is comprised of two major sections.

The first section of this thesis provides an overview of the research problem, literature review, and study of historical and current data management models.

The second section of this study covers the design, development methods, and tools for web-based Polystore systems. In addition, we provide a comprehensive analysis of data processing and access through workflow systems and a comparison of existing systems with other systems.

1.6 Overview of Chapters

Chapter 2 provides a brief overview of the polystore system techniques and their historical context. A survey of data integration, schema matching, and link data models was presented. The structure of these models is discussed in detail in this section.

Chapter 3 provides an overview of all types of data, including big data, astronomical data, open data, and linked open data. Various models have been proposed to manage these data effectively. As these models cannot handle these data efficiently, the Polystore concept was elaborated to solve this problem.

Chapter 4 presents data sets, ZTF data processing, a brief description of the system, and a query system with numerous databases through query formulation, query transformation, query execution, and visualization. In addition, we discuss the proposed method with other polystore system data integration models for managing an extensive data archive via mediation and API connection to a remote cloud database.

Chapter 5 describes the proposed system's data integration model with other polystore system data models for managing a big data archive via mediation and API connections to a remote cloud database. Workflow-based Polystore query management solutions employing a top-down methodology were also discussed.

Chapter 6 concludes the thesis by summarizing and highlighting the limitation of the current study and future research directions.

1.7 Publications

The research and experimental findings of this dissertation have been published in peer-reviewed journals. The finding of the following studies are presented in Chapters 2, 3, and 4 of this dissertation.

- **Manoj Poudel**, Rashmi P. Sarode, Yutaka Watanobe and Subhash Bhalla, A Survey of Big Data Archives in Time-Domain Astronomy. Applied Sciences: Applied Science, (2022), 12(12), 6202.

- **Manoj Poudel**, Rashmi P. Sarode, Yutaka Watanobe and Subhash Bhalla, Processing Analytical Queries over Polystore System for a Large Astronomy Data Repository. Applied Sciences: Applied Science, (2022), 12(5), 2663.

Chapter 2

BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

The "one size fits all" approach employed by (Federated Database Management System) FDBS encounters issues in supplying data management solutions for heterogeneous data, as illustrated in Figure 2.1 [14]. If FDBMS is used to manage multiple types of data, performance and efficiency difficulties may arise. As a result, we require a Polystore architecture, which is a means of combining data from many databases. Polystores provide uniform querying across different data models and are required to manage data across several data models quickly and efficiently. Polystores are a form of database management system comprising several interconnected heterogeneous database engines communicating via an application programming interface (API) [15]. Polystore systems, also known as multi-store systems, provide integrated access to heterogeneous cloud data storage, such as NoSQL and RDMS.

A previous study [17] examined the taxonomy of a Polystore system and classified it as either loosely or tightly coupled. A mediator or wrapper is similar to a loosely connected multi-store architecture. The data store has a unified user interface and can operate independently of the multi-store locally. The wrapper communicates with the data store via an API to generate queries, transform, execute, and return the results to the operator engine. The tightly coupled multi-store system allows for local user engagement and task sharing across multiple systems, resulting in higher performance. Data are also collected from various sources. Benefits of loosely coupled multi-store are combined with the help

of a tightly coupled system in the hybrid system. It uses native sub-queries and operator orders to optimize the querying of various cloud-based data storage sources.

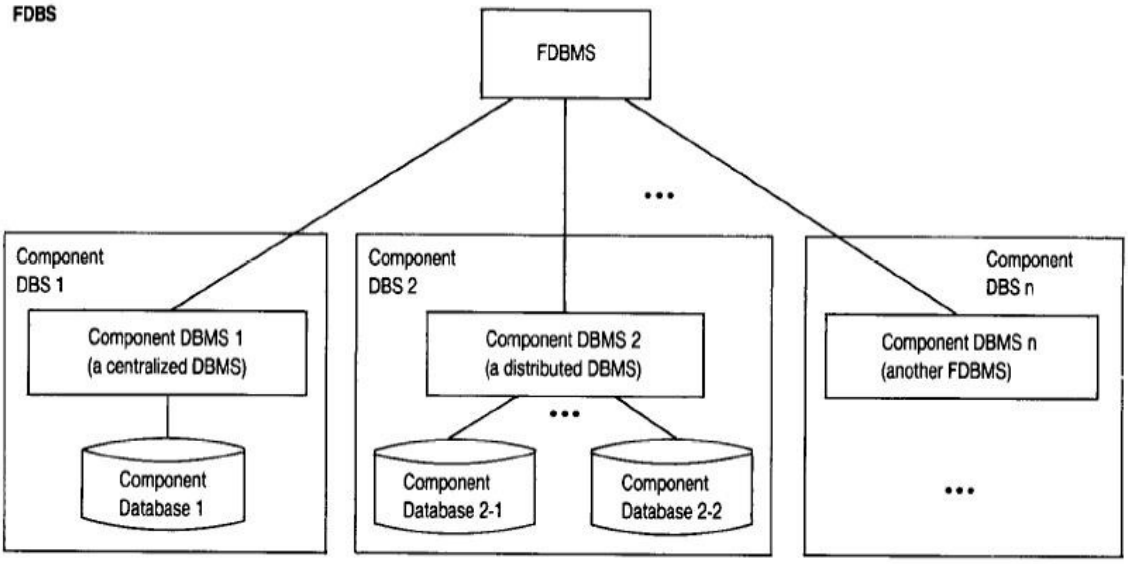

Figure 2.1: Federated Database Management System Architecture [1]

2.2 Polystore System

Many scientific data companies have recently faced difficulties in offering data management solutions for large, heterogeneous data sets with variable data and models. Several solutions have been proposed to address these issues. The Polystore system is one of the solutions that integrate several data and database management systems. Polystore systems, also known as multi-store systems, enable integrated access to numerous heterogeneous cloud data stores, NoSQL RDMS, etc.

The authors of [17] discussed the classification of polystore systems as weakly coupled, tightly coupled, and hybrid systems.

A loosely coupled multi-store system utilizes a mediator or a wrapper concept. It offers a familiar user interface and the ability to locally control data storage independently from multiple stores. Similarly, a wrapper communicates with a data store via an API for query formulation, query transformation, and query execution and returns the result to the operator engine.

Finally, the tightly coupled multi-store system enables a local user interaction interface for enhanced performance by sharing the workload among multiple systems. Additionally, it permits the merging of data from disparate repositories.

The hybrid system combines a multi-store design with loose coupling and a system with tight coupling. It enables native sub-queries and operator ordering to query several cloud-based data store sources [17].

2.3 General Architecture of a Polystore Database System

The BigDAWG (Big Data Analytics Working Group) Architecture at MIT includes a query function for multiple huge data sets in the MIMIC II medical domain. BigDAWG's architecture has four layers: database and storage engines, islands, middleware, API, and applications. Initial releases of BigDAWG supported PostgreSQL (SQL), Apache Accumulo (NoSQL), and SciDB as open-source database engines (NEWSQL). In addition, it supports relational, array, and text islands. The architecture of BigDAWG is shown in Figure 2.2. The client is linked to the middle-ware or API. The middle-ware receives a client query and forwards it to the appropriate island(s) for execution. Shim translated queries from each island and forwarded them to an appropriate database. Casts are used for data migration between different databases [15].

The constituents of BigDAWG middle-ware or API consist of four components: the planner, monitor, executor, and migrator. This is illustrated in Figure 2.2. The planner element parses the incoming query into a collection of objects and generates query plan trees. Additionally, the planner aspect emphasizes the potential data engines for each set of objects. These trees are subsequently transmitted to the monitor elements, which determine the optimal tree for each object group. The trees are then passed to the executor elements, that assemble the collection of objects to execute the query. The executor element can use the migrator element to move items across islands and engines if required by the query plan [15].

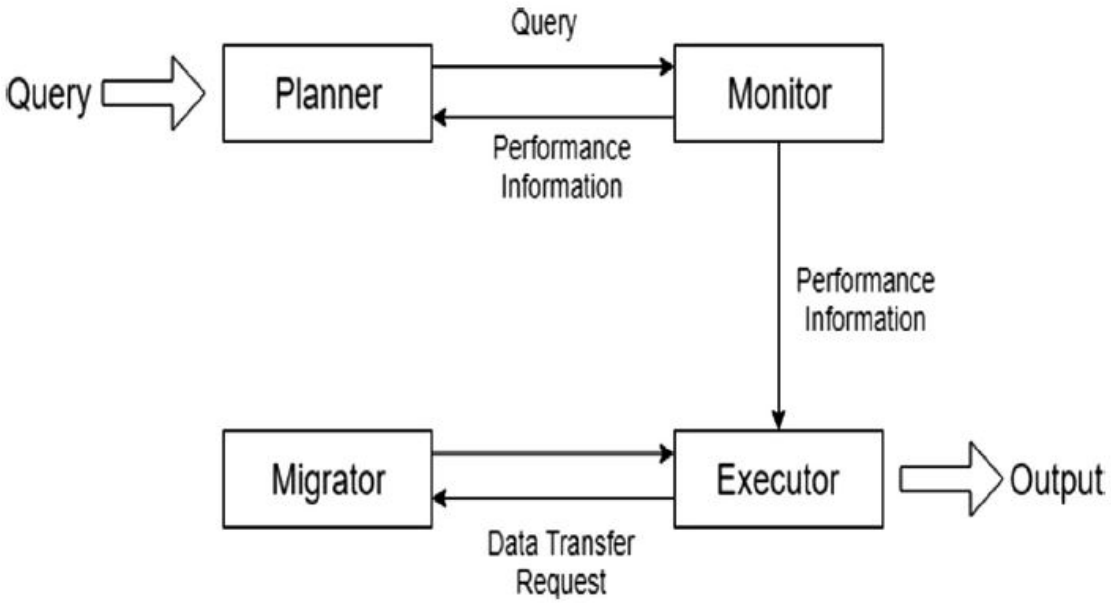

Figure 2.2: BigDAWG Architecture [2]

2.4 Data Integration for Data Science

Data integration has been achieved using a framework known as query discovery. The primary goal is to locate a query (or transformation) that translates the data from one format to another. The purpose was to determine the optimal operators for data joining, nesting, grouping, connecting, and twisting. A strong focus has been placed on data science and analysis. Frequently, data science is performed on massive repositories, also known as data lakes, that include a significant number of diverse data sets. There may be few or no schema in the data-sets [12]. Consider the following configuration example for an integration application with many sources.

There are two data sources, S1 and S2, including data information in Figure 2.3. M12 is the schema-mapping matching operator for sources S1 and S2. G is the mediated schema produced by a schema merge operator on source schema S1 and S2 and mapping M12. The merge operator generates a schema comprising all data from sources S1 and S2. The merge operator also provide mapping from source S1 and source S2 to G, M1G, and M2G.

Now if we add another source, S3, to our system, suppose that S3 is similar to S1 in figure 2.4.

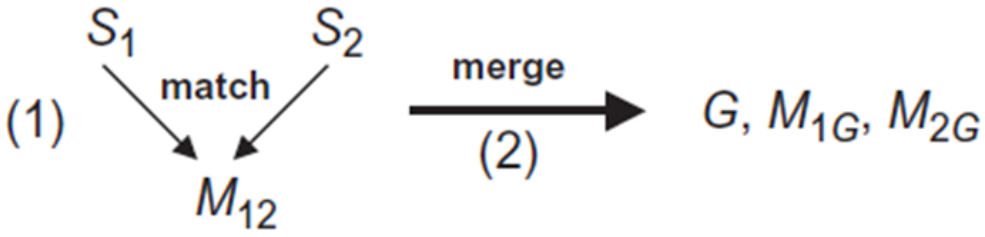

Figure 2.3: Common tasks in data integration with two source

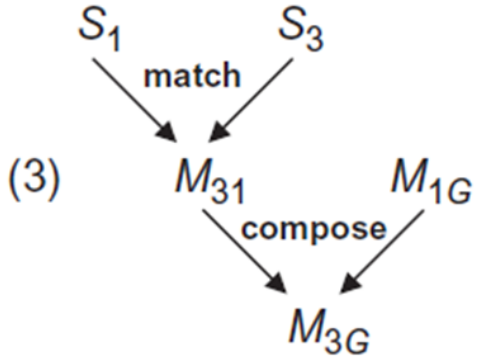

Figure 2.4: Composing mappings by adding new source

From sources S1 and S3, the match helps create a mapping M31. By composing mappings, M31 and M1G help create a mapping M3G.

2.4.1 Open Data

Globally, governments across the globe are seeking to harness the benefits of technology to improve the lives of their citizens. The use of data provided through open government data platforms has the potential to enable innovative services, improve the lives of individuals, and increase the effectiveness of the government and society. Open data helps bridge the gap between the government and the people by enabling public institutions to operate as open, interactive systems. Open data is based on the principle that specific data should be freely available for general usage without restrictions on previously published data. Open data were first meant to make government-provided data accessible to anybody, but they are now utilized by many businesses, organizations, and researchers [18]. The idea behind open data is that the data must be accessible in its entirety and at a reasonable cost for reproduction, preferably via Internet download. Additionally, the data must be

accessible to the user-friendly and editable format. The data must be made accessible under terms that permit reuse and redistribution, and in combination with other data sets [19]. Everyone must be able to use, reuse, and redistribute, and no discrimination against fields of effort, individuals, or groups should be permitted.

2.4.2 Schema Matching

Schema mapping specifies how data are converted between the schema of an external data source and the integrated session schema, as depicted in Figure 2.5. The mapping translates relational database tables and columns into session schema classes and attributes. The schema of the session is derived from all the data stores from which the data has been opened.

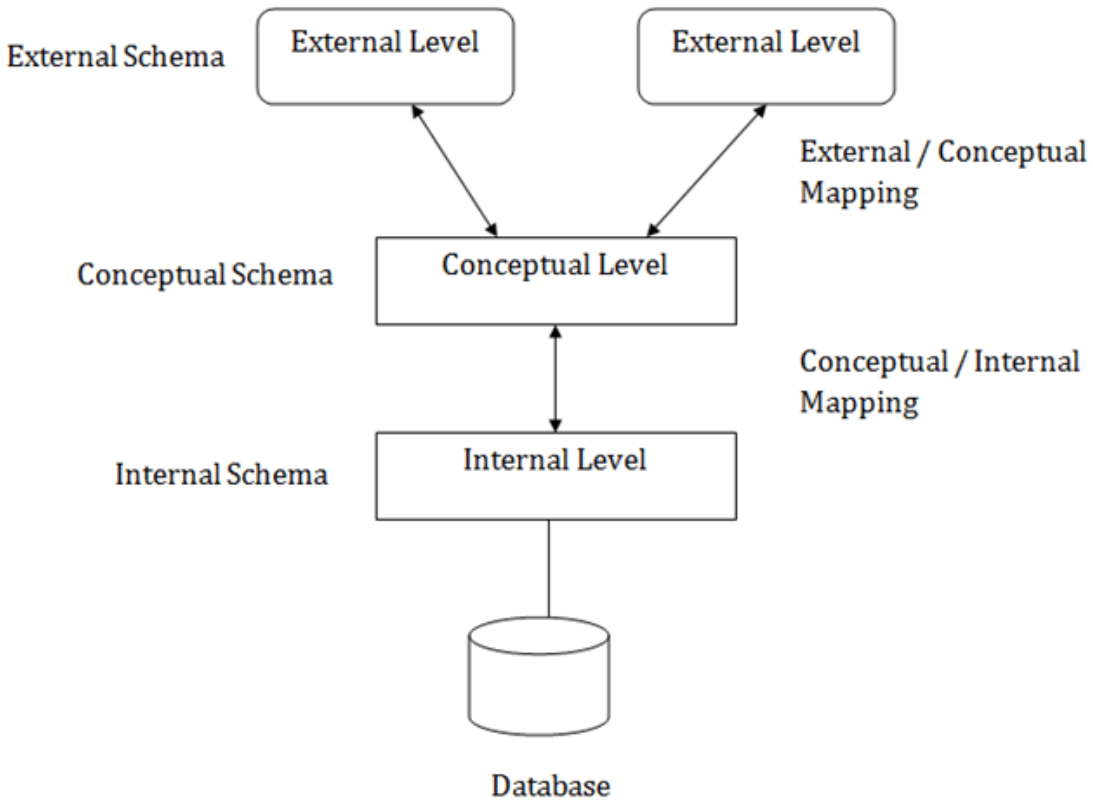

Figure 2.5: Schema Matching Architecture [3]

Mapping transforms requests and responses between different database design levels. Mapping is ineffective for small DBMS because it is time consuming. Conceptual/internal mapping is situated between the conceptual and internal levels. Its purpose is to specify

the correlation between the conceptual level records and fields and the internal level's files and data structures. Exterior/conceptual mapping is positioned between the external and conceptual levels. Its function is to define the relationship between a specific external and conceptual view.

2.4.3 Linked Data

Linked data refers to the best practices for publishing and connecting structured data on the internet. Over the past three years, many data providers have adopted these best practices, resulting in the establishment of a global data space containing billions of assertions, known as the Web of Data [20]. The Semantic Web encompasses much more than merely uploading data to the internet. It involves creating connections so that a person or machine can inspect the data network. When linked data are available, related data can be discovered. Similar to hypertext web, a data web was developed using web documents. In contrast to the hypertext web, where links are anchors for relationships in hypertext pages written in Hypertext Markup Language (HTML), data links are between arbitrary entities defined by the Resource Description Framework (RDF) [20]. URIs can be used to identify anything or notion. However, the exact needs apply to HTML and RDF to help expand the web.

- Utilize URIs to identify objects
- Utilize HTTP URIs to enable users to look up those names.
- When someone searches for a URI, relevant information is provided using industry standards (RDF*, SPARQL) [21].
- Include hyperlinks to additional URIs. That they can continue to learn new things

Chapter 3

MANAGEMENT OF BIG DATA ARCHIVES IN TIME-DOMAIN ASTRONOMY

3.1 Introduction

Considerable research has been conducted on the management of heterogeneous data as a result of the abundance available data sources. Large or complex data that canot be processed using typical methods are known as "big data." For a long time, large amounts of data have been stored and accessed for analytical purposes [22]. Structured and unstructured data are the two types of big data. A large portion of structured data consists of data that have already been entered into databases and spreadsheets by the organization. The term "unstructured data" refers to data not being arranged in a model or format in advance. It incorporates information gleaned from social media platforms, which helps organizations learn more about their clients' wants and needs. Personal electronics and apps, surveys, product purchases, and electronic check-ins are ways to collect large amounts of publicly available data and data provided voluntarily by users. In smart devices, sensors and other inputs enable the collection of a wide range of data in various contexts. In most cases, big data are stored in computer databases and analyzed using software specifically designed to deal with large and complex data sets [23].

Interlinking data in a machine-readable format is at the core of linked data and, a set of design principles for sharing data across the web. This called linked open data when combined with open data (data that can be freely used and distributed). Graph

DB from onto text is an example of an LOD database. It can handle large datasets from various sources and link them to open data for efficient data-driven analytics and knowledge discovery. Linked data is one of the central pillars of the Semantic Web, often known as the Web of data. The Semantic Web is about creating machine- and human-understandable linkages between data sets, and linked data provides the best practices for making these links possible. Linked Data is, in other words, a set of design principles for exchanging machine-readable interrelated data on the Web [24].

In recent years, open data has received significant attention due to: a constant increase in the number of openly published data sets, primarily by governments and public institutions can be considered as the demand for open data increases. However, many potential suppliers are still reluctant to disclose their data sets, and users frequently encounter challenges when seeking to implement such data in practice. This implies that there are still several hurdles surrounding the use and publication of open data, but it is difficult for researchers to systematically collect and evaluate the impact of these obstacles [25].

As in many other scientific fields, astronomy is confronted with a data deluge that requires modifications to the techniques and methods employed in scientific studies. This new era of astronomy has dramatically enhanced research on the entire universe. The study of astronomical topics such as the nature of dark energy and dark matter, the origin and evolution of galaxies, and the structure of our own Milky Way, is advancing rapidly. Research in astronomy is shifting from hypothesis-driven to data-driven and data-intensive [13].

Similar to other data-rich disciplines such as physics, biology, geology, and oceanography, astronomy is facing a data avalanche as a result of advances in telescope and detector technology, the exponential increase in computing capabilities, improvements in data-collection methods, and successful applications of theoretical simulations. An effective federation of database technologies is required for the proper management and processing of large data collections. However, the ultimate objective is to extract knowledge from massive quantities of data, making the development of data mining tools essential.

Knowledge discovery in databases (KDD) is the extraction of valuable information from data. Data mining, the application of certain algorithms to identify uncommon or previously unidentified types of objects or phenomena, is a specific step in the process [26]. In Section 3.4, mining in astronomy is discussed.

Here, we explore big data archives in time-domain astronomy and propose a method for efficiently managing large data repositories.

3.2 Big Data in Time-domain Astronomy

Big data refers to the ever-increasing quantity of information available in various forms and formats. Traditional relational data are not the only source of increasing unstructured data. Large data sets commonly originate from numerous sources. Machine-generated data, for instance, develops quickly and contains a wealth of information that will be unearthed in the future. However, even if human-collected data are predominantly textual, they can still yield valuable insights [27].

Optical time-domain astronomy is close to the tipping point in terms of data rate and volume. By 2022, it is anticipated that the volume of data will increase by a factor of three. As recognized sources increase, efficient and well-designed databases are required. To effectively manage these data, highly effective machine learning algorithms for categorizing source types are required [4].

Numerous scientific disciplines such as astronomy, have become data-intensive in the era of big data and archiving. The rapid development of technology, especially in computer hardware (with low-cost, high-capacity storage and processing) and microelectronics (such as CCD: Charge-coupled Device) devices, has revolutionized most natural science through an explosion in the number of measurements and simulation data [28].

The discovery of an optical telescope and its application to the study of the night sky significantly advanced astronomy in the early 1600s. The objective of global astronomical research projects is to satisfy the data volume and computational challenges associated with solving cutting-edge research issues. The virtual observatory has been proposed as an astronomical community response to the new challenges posed by massive and complex

data sets [29].

Palomar Transient Factory (PTF) is the primary focus of astronomical surveys. One of the primary objectives of this survey was to monitor the northern night sky, observe changes in astronomical bodies, and study optical transients and variable sources, such as stars, supernovae, asteroids, comets, fast-moving solar system objects, and other stellar explosions, using a variety of telescopes. The PTF performs two types of data processing: real-time stream processing and data maintenance with an image archive. Real-time processing of data for sky information updates. It has been maintained as an archive for research studies on various heavy domains. Grant agencies require all astronomy data to be made accessible to astronomers worldwide. These archival data are accessible to the public via a web-based infrared science archive (IRSA/IPAC) system [4]. The PTF has been in operation since 2009 using a 7.2 deg2 camera mounted on the Palomar Samuel Oschin 48-inch (1.2 m) Schmidt telescope [30].

Beginning in 2013, the Intermediate Palomar Transient Factory (iPTF) project expanded upon the legacy of the Palomar Transient Factory (PTF) led by Caltech. Through the historical Palomar Transient Factory data and rapid follow-up studies of transient sources, the iPTF enhanced software was used for data reduction and source classification [4]. The image processing and differencing pipeline innovations made it possible to receive transient candidates significantly more quite quickly than usual (from 30 min to 60 min to 10 min in iPTF). The PTF/iPTF generated approximately 0.05 petabytes per year or 1 gigabyte of data every 90 s [31].

In 2017, an iPTF telescope was transformed into the Zwicky Transient Factory (ZTF) telescope. ZTF utilizes a new camera to generate new reference image catalogs, lightcurves, and transient candidate images [4]. Using a 48-inch Schmidt Telescope, ZTF has the largest instantaneous field of view of any camera on a telescope with an aperture greater than 0.5 meters [16]. The ZTF Observing System provides time-domain astrophysics analysis with high-speed, wide-field-of-view, and multi-band optical imagery. Large amounts of data from the ZTF will serve as a reference for the next Legacy Survey of Space and Time (LSST). The LSST will conduct measurements of position, fluxes, and shapes, as

well as lightcurves and calibrated images [4].

3.3 A Research Agenda

A machine learning-based classification broker for petascale mining of large-scale astronomy sky survey databases requires a number of major research ideas to be addressed. Data mining and computational science experts have already taken on several research ideas in their own work [32].

Classification labels are useful only if the community as a whole agrees with the correct set of semantic ontological, taxonomical, and classification terminology. Research into the completeness, utility, and usability of ontology's currently being developed in astronomy is required. To design, develop, and implement a user-oriented petascale data mining system, we must do research on user requirements and scientific use cases. A comprehensive set of classification standards for all conceivable astronomical events and objects must be developed and explored. Robust rules and classifiers are required to detect outliers and novelty in objects and events that are currently unknown [32].

To classify all the different types of training sets, we will have to conduct extensive research and gather a large number of classes. These samples were used to train and validate the classification brokers. Research, development, and validation of algorithms for web service-based classification and mining of distributed data are required. A text-and-numerical data mining technique may be the most effective and should thus be researched. Prototypes and demonstrations of the classification broker user interface and interaction models are required [32].

The astroDAS system components must be integrated in a sturdy manner. Various forms of interaction and integration, such as grids, web services, RSS feeds, ontology's, and linked databases, will have to be researched in this regard. The functioning, usefulness, bottlenecks, failure modes, and scalability of a working classification broker on a real-time astronomical event message stream are tested. Security (from the current few events per night to many tens of thousands of events per night in the coming decade). Interestingly, similar event message feeds are already accessible, albeit on a considerably more limited

scale than that projected to be provided by the LSST in the near future [32].

3.4 Existing Astronomical Data Mining

Owing to the rapid growth in data volume from various sky surveys, the size of data repositories has increased from gigabytes to terabytes and even petabytes. This field of study analyzes massive astronomical collections and surveys using data mining technologies. Large-scale data analysis is referred to as "big data analysis." Data mining is a collection of techniques used to reduce, enhance, and purify large quantities of data. These methods include summarization, classification, regression, clustering, association, time-series analysis, and outlier/anomaly detection [27].

The primary emphasis of the data-mining overview is knowledge discovery in databases (KDD). Nevertheless, the term 'database' encompasses all machine-readable astronomical data [33].

Numerous terms are associated with data mining, and we begin by defining them as follows:

- **Data collection:** Data collection encompasses all the actions necessary to gather the desired data in a digital format. As part of the research procedure, data collection methods include collecting new observations, querying existing databases, and completing data mergers (data fusion). Cross-referencing massive data sets can result in confusing matches, disparities in the point spread function (object resolution) within or between data sets, adequate processing time, and data transit requirements. A few arc seconds of astrometric tolerance were employed when each database item lacked a specific identifier.

- **Processing of Data:** Preprocessing may be necessary during the data collection process, such as sample cuts in database searches, may be necessary during the data collection process. It is crucial to exercise caution when preprocessing the data because the input data can substantially affect numerous data-mining techniques. Preprocessing can be divided into two types for a given algorithm: procedures that

make the data readable and processes that alter the data in some way.

- **Selection of Attribute:** Some of an object's numerous properties are not required for a proper operation. It is possible to utilize all the qualities to maximize the performance. This has created numerous low-density habitats and voids. Data cannot be easily mined for novel concepts. Dimension reduction is essential for retaining as much information as possible while employing fewer attributes algorithms are hindered by the presence of superfluous, redundant, or otherwise unimportant characteristics. The location of a survey with a uniform mask is an example of an unnecessary characteristic, because the color observed in two apertures with the same waveband would be extremely redundant.

- **Use of machine learning algorithms:** Methods for machine learning are classified as supervised or unsupervised. Semi-supervised approaches utilize two sets of objects for which the target property, such as classification, is known confidently. The algorithm is trained on these objects before being applied to others that lack target attribute. These additional items were included in the test set. In the majority of instances in astronomy, a photometric sample of an object can predict characteristics that generally require a spectroscopic sample. The parameter space spanned by the input attributes must encompass the application of an algorithm. This may initially appear limiting, but combining data sets can frequently circumvent this limitation [34] [35].

The ability to handle large and distributed data sets and execute complicated knowledge discovery tasks is a common requirement in a wide range of scientific fields. To complete a variety of data-mining jobs in various industries, data-mining experts have created numerous pieces of software and tools as described below. Scientists from a variety of fields are joining to build astronomical data mining software and tools [13].

- StatCodes is a Web meta site that provides hypertext links to a large variety of statistical codes that are beneficial for astronomy and related subjects [36].

- VOStat is a statistical web service hosted by Penn State University and its a R language-based GUI wrapper. The primary goal was to encourage astronomers to applied statistical methods and spread the use of R among astronomers. Interactive 3D visuals are also possible because of the program's ability to execute a variety of statistical studies including data smoothing and time series analysis, as well as a wide range of statistical tests [37].

- Weka used machine learning techniques to complete various data mining task, such as data pre-processing, classification, regression, clustering, association rules, and visualization. It can also be used to create new algorithms for machine learning. It is a user-friendly, open-source data mining tool that can be applied to a wide range of data mining jobs [38].

- AstroWeka is a set of enhancements to Weka specifically designed for astronomical data mining. For data loading, it uses the Astro Runtime and Starlink Tables Interchange Library [39].

- AstroML is a Python module for machine learning and data mining, based on astropy and other libraries. The purpose was to provide a repository of rapid Python implementations of common statistical data analysis tools and procedures used in the field of statistical astrophysics, as well as an interface for freely available astronomical datasets [40].

- Data Mining and Investigation (DAME) specializes in the exploration of enormous data sets using machine learning methods, and is a web-based and distributed data mining infrastructure. Photometric redshift, photometric quasar candidate extraction, globular cluster search, active galactic nuclei classification, photometric transient classification in multi-band, and multi-epoch sky surveys are some of the examples that have been used [41].

- Auton Lab led by Artur Dubrawski and Jeff Schneider, focuses on novel methods of statistical data-mining. They are interested in underlying computer science,

mathematics, statistics and artificial intelligence to find patterns in data and exploit them [42].

3.5 Time-domain Astronomical Archives

The following Scientific Archives and Services are accessible on the Web:

1. **SIMBAD (Set of Identification, Measurements, and Bibliography for Astronomical Data):** SIMBAD [43][44] is the primary database for the identification and bibliography of astronomical objects. e Centre de Donn'ees astronomiques de Strasbourg (CDS) has developed and managed the SIMBAD. Several astronomical objects are included in the database, bibliography, and observational measurements. Priority is given to catalogues and tables covering a broad spectrum of wavelengths and supporting large-scale research initiatives [35].

   A systematic review of the literature provides an overview of contemporary astronomical research, including its diversity and more significant trends. A WWW interface for SIMBAD is available at:http://simbad.u-strasbg.fr/Simbad.

2. **SMOKA (Subaru-Mitaka-Okayama-Kiso-Archive):**

   The SMOKA [45] science archive system contains data from multiple telescopes. Currently, the server stores almost 20 million astronomical frames, totaling more than 150 gigabytes. In addition, the search interface allows searches based on various search limitations and FITS-header keyword values for certain data sets.

   The search interface provides access to the following data from telescopes and observatories: Subaru (Subaru), OAO (Okayama), Kiso (Kiso), and MITSuME (MITSUME) equipment and reduction tools [35].

3. **IRSA (NASA/IPAC Infrared Science Archive):**

   The Infrared Processing and Analysis Center (IPAC) [46] provides support for several NASA [47] programs, including Spitzer, the (NEO) WISE and 2MASS satellites, and IRAS. IPAC manages NASA's data archives. In conjunction with NASA, IRSA

also provides access to data from ESA missions, including Herschel and Planck. Data from Infrared Telescope Facility (IRTF) and Stratospheric Observatory for Infrared Astronomy (SOFIA) are archived at IRSA. IPAC's non-NASA or non-infrared projects, including the Palomar Transient Factory (PTF), Zwicky Transient Facility (ZTF), and Vera C. Rubin Observatory (VCRO), benefit from IPAC's archiving technology (formerly known as LSST) [48]. The IRSA contains one petabyte of data from more than fifteen projects. IRSA provides access to more than 100 billion astronomical measurements, including coverage of the entire sky in 20 bands.

Astronomers can only retrieve data from various celestial bodies using query tools. Images and information linked to them are the most commonly researched topics. Depending on the circumstances, users may have a variety of needs. Users may wish to access information by querying a single item or a collection of objects. To locate exact information inside the astronomical domain, the user must create elaborate programs or formulate complex queries.

For decades, large-scale data management has relied on parallel DBMS. In addition to conventional relational DBMSs such as MySQL and Oracle, new data stores based on the ACID (Atomicity, Consistency, Isolation, Durability) [49] principles have been proposed to manage vast amounts of data. Numerous big data applications do not require strict ACID compliance and favor performance in terms of consistency and reliability. Systems for large-scale data storage and warehousing, such as Megastore, Mesa, and Spanner, are designed using SQL-based query languages. In addition, NewSQL databases are designed for high-throughput online OLTP while maintaining ACID properties [50].

3.6 Challenges of the Future Data Management in Existing Time-domain Archives

The current trends in database management necessitate the employment of numerous models and data repositories. Previous methods such as Federated Database Systems

(FDBS) and data warehouses work well with relational data but cannot store many data types (arrays, graphs, and images). Different data stores manage various data types, each have their own native language.

Federated database systems (FDBS) consist of cooperative but autonomous databases. With FDBS, decentralized local databases can exert greater control over the exchange of information. FDBS uses federated query agents (FQA) to process the queries. These agents function as intermediates (mediators) between the two to store and execute queries. Data in FDBS is stored in a relational database, the only data model the system supports [51] [35].

A data warehouse is a relational database designed for analysis instead of transaction processing. It often comprises historical data gathered from transactions, although it may also include information from other sources. It permits businesses to combine data from numerous sources while segregating analytical and transactional tasks. Several programs that manage to obtain and distribute data to business users may be found in a data warehouse environment. A central data warehouse or repository contains data warehouses [52].

Owing to the emergence of big data, models such as FDBS and data warehouse appear inefficient, as they can only integrate databases with a single data model that is no longer relevant. In addition, the expansion of the data volume and velocity cannot be accommodated. These models also lack the price and performance. It is now safe to say that multiple heterogeneous data management strategies provided by past data integration models have failed. Managing vast quantities of unstructured data from diverse data repositories is gaining increasing interest in the database community. Owing to the increase in data size, rate of data growth, and emergence of new data types in numerous scientific data archives, this issue has attracted more attention. In contemporary database engineering, the "one size fits all" approach [53] is no longer applicable. The underlying database management system must have full liberty to optimize queries. Using a unified query language, a model that can bridge heterogeneous data sources is required. In addition, data virtualization through mediation is required to achieve these requirements

[35].

3.7 Proposed Data Management Model to handle Time-domain Archives

A Polystore can span several data management systems without the need for an underlying data location or storage engines, and it can be queried using a single language [2]. Polystores will enable a many-to-many connection between information islands and data management systems across diverse data models and query languages [15]. Polystores also enable seamless access to several cloud data stores. The CloudMdsQL Polystore provides an SQL-like query language for accessing various data sources (relational, NoSQL, and HDFS) [54].

Polystore systems or multi-store systems have recently been introduced as a novel solution to data integration that allows integrated access to heterogeneous data stores via a unified single query language. In addition, Polystore Systems solves heterogeneity problems by providing a communication protocol within the underlying database management systems using islands/shims, mediation, or APIs (application programming interface).

Polystore aids astronomers in integrating database content into their own data portals, thereby offering scientific content to their respective communities of users. Polystores can help integrate sky surveys, robotic telescopes, and other data sources. Astronomers can view an integrated database for query analysis (either manually or through web services). Ontology's, semantics, dictionaries, annotations and tags are used along with data/text mining, machine learning and information extraction algorithms as part of Polystores. It can also access database repositories, grids, and web services. Finally, it can share databases in a collaborative, dynamic, and distributed manner [35].

3.8 Summary

Every type of data has been addressed, including big data, astronomical data, open data, and connected open data. These data are extensive, heterogeneous, and complex. In addition, we explored big data in astronomy and in existing archives. Several astronomical archives and their respective query languages have been described. The single-data-model models of the past, such as FDBS and Data warehouse, are ineffective for managing massive amounts of data. The Polystore concept, which uses a uniform query language to efficiently span multiple heterogeneous data models, can be used to address this issue.

It is necessary to manage heterogeneous open data on the Internet and create a query language to combine web services and data repositories. Polystores can facilitate the integration of disparate heterogeneous data stores for information retrieval, data visualization, and the development of useful online applications.

Chapter 4

DESIGN AND DEVELOPMENT OF WEBBASED POLYSTORE SYSTEMS

4.1 Zwicky Transient Facility (ZTF) Overview

The Zwicky Transient Facility (ZTF) is a new time-domain survey that uses a wide-field survey camera to detect and analyze supernovae, variable stars, binaries, active galactic nuclei (AGN), and asteroids [55]. The ZTF was design to detect near-Earth asteroids, unique and rapidly changing flux transients, and all types of variable galactic plane sources [56]. In 2013, the Intermediate Palomar Transient Factory (iPTF) project was initiated, building on the legacy of the Caltech-led Palomar Transient Factory (PTF) [57]. Through historical PTF data and rapid follow-up investigations of transient sources, the iPTF improved data reduction and classification tools. In 2017, the iPTF was transformed into the Zwicky Transient Factory (ZTF), utilizing a reconstructed version of the same telescope that the iPTF utilized. The Samuel Oschin 48-inch (1.2m) Schmidt telescope was outfitted with a brand-new camera with a 47-square-degree field of vision for ZTF observations. [58]. The camera comprises 16 CCDs separated into four quadrants for the readout. Consequently, each ZTF exposure generated 64 CCD quadrant images [59]. A CCD-quadrant is the primary image unit for pipeline processing and the origin of all the output scientific data. The Legacy Survey of Space and Time (LSST) uses ZTF's huge amount of data as a reference for its next endeavor [60]. The LSST will perform location surveys, flux and shape measurements, light curve analysis, and calibrated images. The

Table 4.1: ZTF science data product

Project name	**Duration**	**Data Download**	**No. of FITS File**	**Product**
PTF(Level 0, Level 1)	2009-2012	0.1 TB per night	Around 3 million	Epochal images, Photometric catalogs
iPTF(Level 2)	2013-2017	0.3 TB per night	Around 5 million	Deep reference, Lightcurves
ZTF(Data release 1 to 8)	2017-2021	1.4 TB per night	Around 50 million	New reference, Lightcurves, Transient canididates, catalog
LSST	2022-2024	3 TB per night	Around 500 million	Calibrated images,measure of position, flux and shapes, and Lightcurve

development of the PTF and ZTF projects, as well as the product specifications, are detailed in Table 4.1.

According to study [57], all image data are available in the Flexible Image Transport System (FITS) format, which includes epochal (single exposure) photos and photometric catalogs. The images were captured using 64 CCD (Charge-coupled Device) cameras outfitted with various image-quality filters. Photometric catalogs provide image information in key-value pairs and header information. These key-value pairs are convertible to relations and can be stored in relational database management systems (RDBMS). Only a subset of the data is downloaded for indexing, depending on the size of the data and the available resources. Consequently, header files containing astronomical image data were downloaded. The header files were downloaded from the hierarchical file system (HFS) of the IRSA online service and replicated on the local server. The catalogs also include header files containing the meta-data (HTML elements) required to connect to the IRSA web service and retrieve the images. The key values and header information for the images obtained from the IRSA/IPAC were stored in PostgreSQL. Images were accessible through the IRSA/cloud IPAC service. The repository currently houses data for 2017–2020, totaling approximately 50 Terabytes.

4.2 ZTF Data Processing Overview

The Infrared Processing and Analysis Center (IPAC) developed the PTF and ZTF Science Data Systems (ZSDS) [56]. The ZSDS was designed to provide a processing and archiving system capable of producing scientific-grade output. It is maintained as a repository for research in various astronomical fields. The data were analyzed in real-time to provide real-time updates to sky information. These data are processed in several ways to produce various outputs for use in other scientific endeavors. Example include data processing pipelines, data archives, long-term curation infrastructure, and data retrieval user services. Nine pipelines run on different timescales: raw data ingestion pipeline, image splitting pipeline, calibration derivation pipelines (Bias-image Generation, Flat-field Image Generation), instrumental and photometrical pipeline, reference image pipeline, real-time image subtraction and extraction pipeline, light curve pipeline, and ZMODE: ZTF moving object discovery engine [56]. As required by grant agencies, all astronomical data must be made publicly available to astronomers worldwide. These archival data are freely available to the public via the Infrared Science Archive's (IRSA/IPAC) browseable web directory [61]. The ZTF Science Exposure Metadata, calibration metadata, raw metadata, and reference image metadata can all be accessed via their online system. Algorithm 1 describes the steps involved in the real-time data processing in the ZTF archive.

4.2.1 IRSA Archive

IRSA/IPAC is responsible for curating and distributing the images and catalogues. In 2019, roughly 6.9 million single-exposure images, 135,000 co-added images, 106 billion source catalogue files, and 2 billion lightcurves will be generated using single-exposure extractions with previously available catalogues [62]. All image data, including scientific images, calibration image metadata, raw image metadata, and reference image metadata, are stored in the Flexible Image Transport System (FITS) format [63]. The ZTF archived web directory structure is presented in Table 4.2. The IRSA stores the ZTF FITS file in its web directory in the B-tree data structure defined in a previous study, as shown in

Algorithm 1: Real-time data processing in ZTF Archive

Input: New raw CCD-based data set (CCD_RAW) with corresponding raw image data (raw), calibration product (cal), epochal science product (sci) and reference science product (ref) are generated by checking some predefined requirement with all FITS HDUs in ZTF Archive

Output: New CCD-Based Images (CCD_RAW)

$CCD\_RAW \leftarrow$ Identify_New_CCD_Images (raw, cal, sci, ref)

for *Archive* ϵ CCD_RAW **do**

  if *Requirement_verified (Archive(raw, cal, sci, ref))* **then**

    $CCD\_RAW \leftarrow CCD\_RAW$+Archive

  end

end

for *Archive* ϵ CCD_RAW **do**

  $Requirement\_fails \leftarrow$ Fails_status_flag(Archive(raw, cal, cal, sci))

  if *(Requirement_fails) = (CCD_RAW)* **then**

    $CCD\_RAW \leftarrow CCD\_RAW - Archive$

  end

end

Figure 4.1 [4]. The top node is the root node of the IRSA web directory, and the index is stored in the leaf node. The ZTF FITS image data are stored in leaf nodes, where all insertions and updates occur.

The i^{th} record contains an absolute identity x_i used to identify the base record. The components with x_i are year, two-digit month, two-digit day, fractional day, field, and filtercode, CCDID, image type code, and quadrant ID, as shown in Figure 4.2. There are approximately 106 billion records in the ZTF archive. Each record has up to one billion data unit fields, which are identified as $y_{i,j}(y_{i,1}.....y_{i,1billion})$ where (i, j) indicate the record identity (x_i), and the position of the field $((y_{ij})$ in the individual record.

Table 4.2: ZTF FITS file index in web archive

Header	Data Unit	
x_i, y_{ij}, Size and index for the data (1, ... 50 million)	y_{ij}	
	Name, Size of Data	Data type
	Night, field,..	FITS,Log ...

In the Figure 4.2,

- /raw = raw image data file;
- /cal = calibration product file;
- /sci = epochal science product file and difference images;

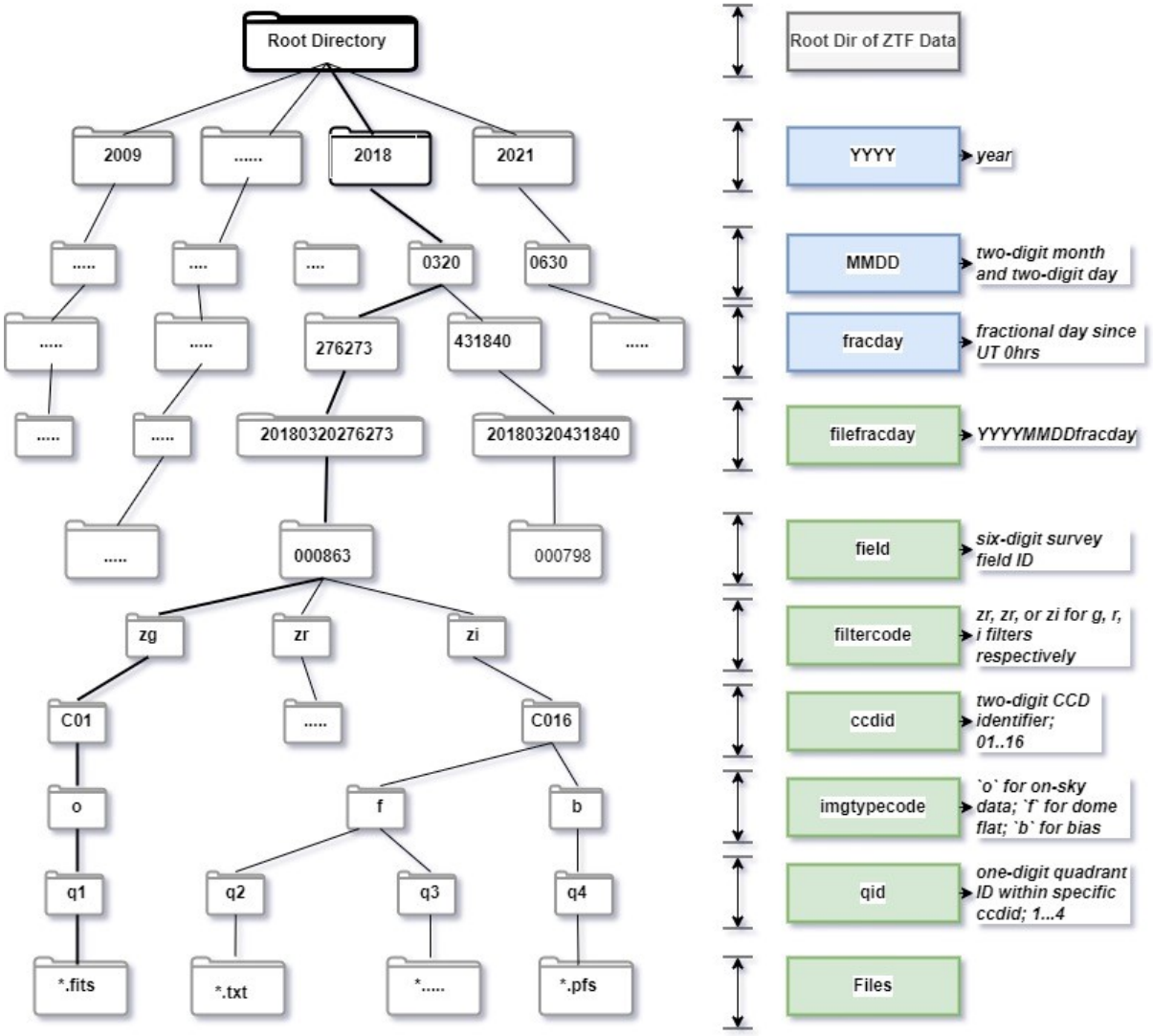

Figure 4.1: ZTF Science Images Generic Root Path in The Archive [4]

Figure 4.2: ZTF Archive [5]

- /ref = reference image (co-adds) and catalog files;
- <fff> = first three (leftmost) digits of <fieldID>;
- <caltype> = calibration type, e.g., "bias", "hifreqflat";
- <imgtype> = single character label in raw camera image file;
- <ptype> = product type suffix string from pipeline;
- YYYY = year;
- MMDD = month and day (all UT-based);
- dddddd = fractional time of day of exposure (all UT-based);
- fieldID = 6-digit survey field ID if targeted science (on-sky) exposure, otherwise "000000" for calibrations;
- filterID = 2-letter filter code:zg, zr, zi for exposure acquired in g, R, or i respectively (for on-sky & flat-fields) bi, dk for bias and dark images, respectively (filter neutral);
- ccdID = 2-digit detector chip ID: 01, 02, ... 16;
- quadID = 1-digit quadrant ID in ccd: 1, 2, 3, or 4 imgtype o: on-sky object observation or science exposure, b: bias calibration image, d: dark calibration image, f: dome/screen flatfield calibration image, c: focus image, g: guider image;
- ptype = fits, txt format image product type.

All image data, including scientific images, calibration image metadata, raw image metadata, and reference image metadata, are stored in the Flexible Image Transport System (FITS) format [64]. Access control is only applicable to password-protected, and non public data. There are instructions on the access control pages for using IRSA APIs with a password. Using the IRSA ZTF-LC-API form of HTTP URLs, ZTF Lightcurve data can be queried or retrieved via the specified ZTF objects by their identifiers (ID), by position (POS), by a collection of ZTF files (COLLECTION), and by the format of the output table (FORMAT) parameters.

The IRSA/IPAC directory indexes image URLs to support ZTF FITS files with API support for images visualization. This makes it simple to embed the archive directly into the user's software.

4.2.2 ZTF Released Products

IRSA has archived all raw metadata, processed data, and data archives and made them accessible to the public using data exploration tools. As stated in Table 4.1, ZTF image data products are made accessible to the public in a number and variety of formats. All image data were made publicly available in the Flexible Image Transport System (FITS) format, including CCD-based image metadata data files, CCD-quadrant-based image metadata files, single-exposure science images, source catalogues, and reference images per CCD-quadrant. According to Table 4.1, a CCD-quadrant is the core image unit for pipeline processing, from which all scientific data outputs (DR1, DR2, DR3, DR4, DR5, DR6, DR7, and DR8) are generated.

4.2.3 Access to ZTF Data

Using the IRSA ZTF graphical user interface (GUI), ZTF data can be retrieved, visualized, and analyzed from file-based products, such as single-exposure science images or reference images, and their catalogues or other files [65]. The IRSA offers two graphical user interfaces for gaining access to ZTF data, ZTF Images services and catalogue services. The IRSA/IPAC provides a download platform for these data via a web-based, navigable directory. Using a graphical user interface, users can query the ZTF images service to view and download ZTF images and search for astronomical objects by position, ZTF field ID, or solar system object/orbit. IRSA image services implement a low-level search method for metadata tables associated with astronomical images. IRSA image search services provide both single-object and multiple-object queries. The single-object search capability of the system enables the user to locate astronomical bodies by name or position. The query returns CCD-quadrant images that intersect these spots, together with metadata for additional filtering. The results are displayed in a web browser in a table with multiple

columns. The graphical user interface (GUI) offers previews and interactive analysis of selected images. The user must manually construct a complex SQL query and upload it to the system to conduct a multi-object search. The system permits the user to load a file from a local disk or workstation in an infrared scientific repository.

Users can retrieve, view, and assess catalogue services using GUI services in three stages. The ZTF catalogue services contain numerous tables, including ZTF DR1, DR2, DR3, DR4, DR5, DR6, DR7, and DR8 objects. Each table is independent and can be queried separately. The user can search for a single object using a single location, either a search radius or box size centred on that location. To locate an astronomical object, a single object search, multi-object search, or all-sky search can be used. Upon clicking "Run Query", all objects and related light curves were displayed in a web browser using a multi-column table based on user input. To obtain images from the multi-object search, the user must manually construct a complex query and submit it to the workstations of the infrared scientific archive. All-sky searches retrieve counts from the entire database table, regardless of whether they are in ASCII (IPAC-table), HTML, or XML format. The all-sky search option does not return light curves.

On the results screen that follows the query in step 1, the user can click the "Time Series Tool" to retrieve the lightcurves of user input items. In step 2, the user can select a specific object from the list of object IDs containing lightcurve data and transfer its lightcurve to the Time Series Tool by clicking on it. These tools provide an object centered epoch-based scientific image and period locator.

4.2.4 Retrieving the catalog file from IRSA Remote Resource

The real-time identification and classification of astronomical objects, such as variable stars and super-novae, is the major purpose of the ZTF project. This initiative also seeks to organize and analyze a database containing additional celestial objects for an in-depth examination. Astrometric calibration outputs (ZTF CCD-quadrant images) were in the FITS format. Each image within an FITS file consists of a header data unit (HDU) and image data. HDU is observational metadata, which is the information associated with an

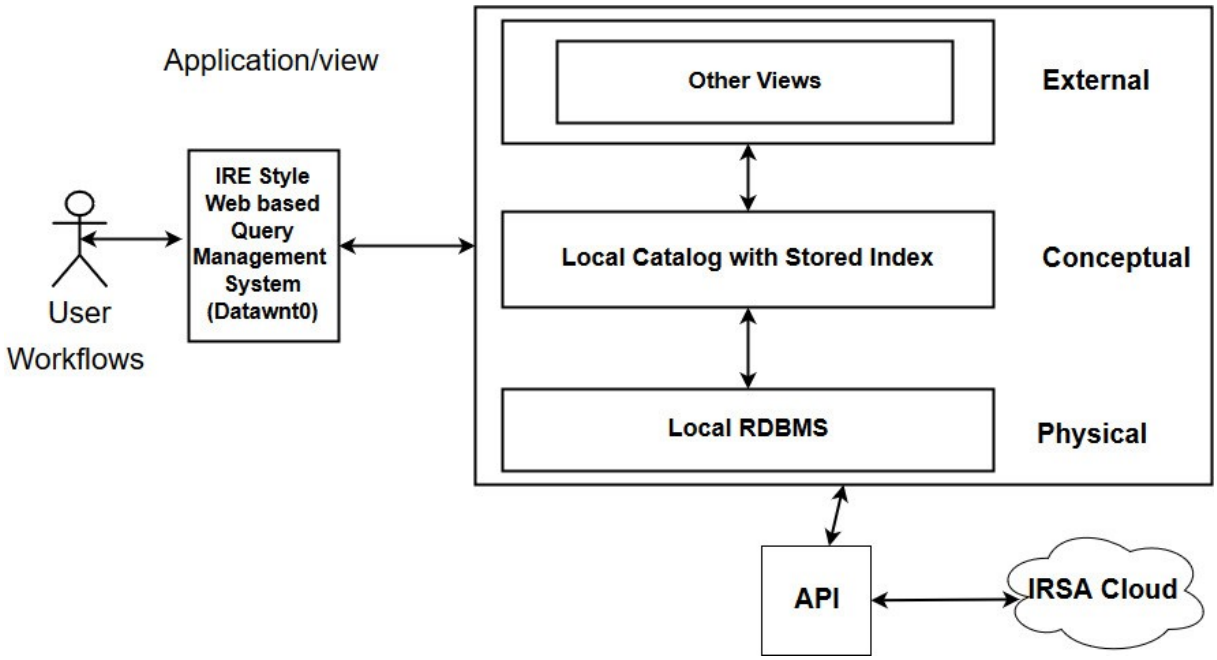

Figure 4.3: Local three-level architecture connected to remote cloud data source

image file.

To provide an Information Requirements Elicitation (IRE)-style query interface for ZTF data, a three-level architecture in the ANSI dictionary relational form is used to retrieve the data, as shown in Figure 4.3 [66]. The conceptual level comprises local catalogs with an index that is stored. A local RDBMS (Postgres) is used to store catalogs. In addition, the current three-level architecture is linked to a remote cloud service on the IRSA server to download and transfer data using various APIs (Astropy and JS9) [67]. Because of the identical schema design, any modifications to the remote data repository can be reflected in the local architecture. If necessary, the conceptual schema can be expanded by adding additional indices to the local catalogues. Additional external views that support additional user workflows can be added.

We developed a local repository for this study. The IRSA remote server retrieves the table header information, image data, and key-value pairs. The downloaded unstructured data are rearranged into a relational database schema. The download process was automated using a Python script that generates SQL on the fly, and then optimized and stored in a PostgreSQL database on a local LINUX server with 16GB of memory and 5TB of disk space. SQL script helped optimize and create a new entity-relationship-based tables

(key and value) from the local database. (see 4.4). These database tables are Nights, Exposures, CCD, Filters, Fields, procimages, and Host Galaxy.

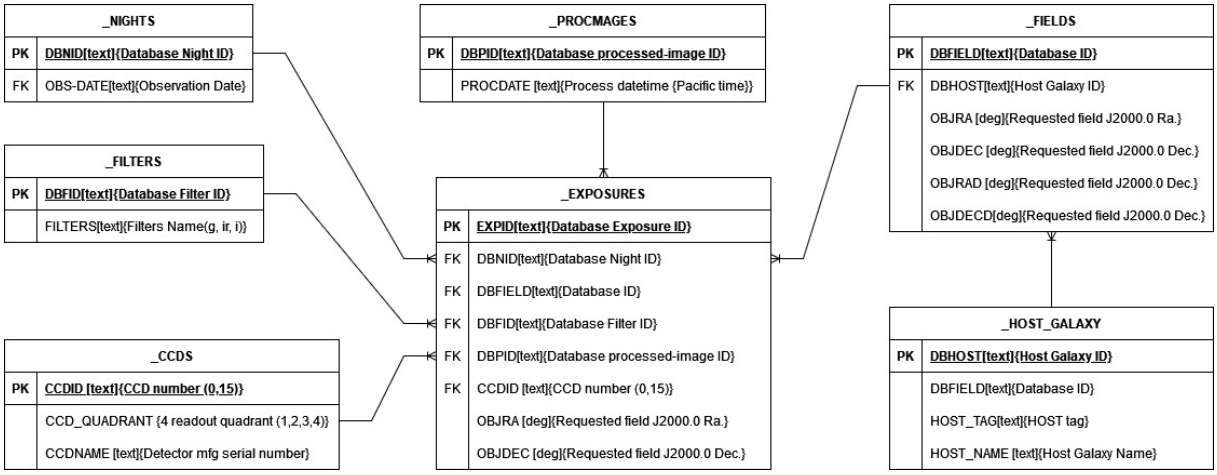

Figure 4.4: ER Model of the Database with Relations and Attributes [5]

- Nights: Nights contain the date or time of the images taken, with the unique index *nid* and alternative key—nightdate.
- Fields: Fields database table stores images according to the X and Y coordinate and assigned identification ID. In this table, *fieldid* is a unique index and *field* is an alternative key.
- Exposures: Exposure tables contain information from both night database table and fields database table. Exposure tables also contain detailed information on the CCD, such as CCD ID, exposure time, image type, etc. The *expid* is a unique index and *obsdate* is an alternative key.
- Procimages: Procimages contains processed-images metadata, i.e., image file names. The unique index is processed-images number *pid*.
- Filters: Filters includes a record for each camera filter that was used to acquire the exposures. The unique index is *fid*.
- CCDS: CCD (16 charge-coupled devices) contains the camera details numbered CCDID 1,...16. The unique index is *ccdid*.
- Host_galaxy: Host galaxy consists of the names of the galaxies.

4.3 Proposed System Overview

Domain experts in astronomy require query tools to retrieve data from astronomical bodies. Popular domain specialists in astronomy require demand query tools to retrieve data from a range of celestial bodies. The most popular method is for images and image-related information. Depending on these conditions, the user may have a range of requirements. To acquire information, users may wish to query a single object or a collection of objects.

Given the current state of data access in ZTF, this research focuses on developing an alternative top-down workflow web-based query management system for accessing ZTF data when searching for images and image-related information. Depending on these conditions, the user may have a range of requirements. To acquire information, users may wish to query a single object or a collection of objects. Current methods for astronomical domain-specific searches require the user to write complex scripts or formulate complex queries to acquire meaningful insights. Current methods for astronomical domain-specific searches require the user to construct elaborate scripts or formulate complex queries to acquire meaningful insights. Given the current status of ZTF data access, this research focuses on developing an alternative top-down workflow web-based query management system for ZTF data access.

4.3.1 Proposed System Architecture

The proposed system was a web-based information system. The HTML/CSS defines the layout based on the design of the user interface. Both JavaScript and PHP provide dynamic multi-stage table querying. The Figure 4.5 represents the architecture of the proposed system. Each file's header information was downloaded and stored on the local server (RDBMS), whereas the image file and all other associated data were stored on the IRSA cloud server. Consequently, the proposed query system supports the conversion of keys to addresses and the movement of image retrieval queries from the local server to the IRSA cloud server. In the proposed system, users can select objects and enter values based on their search criteria in an input box.

This system allows users to select and add predefined IDs or names. Users can associate

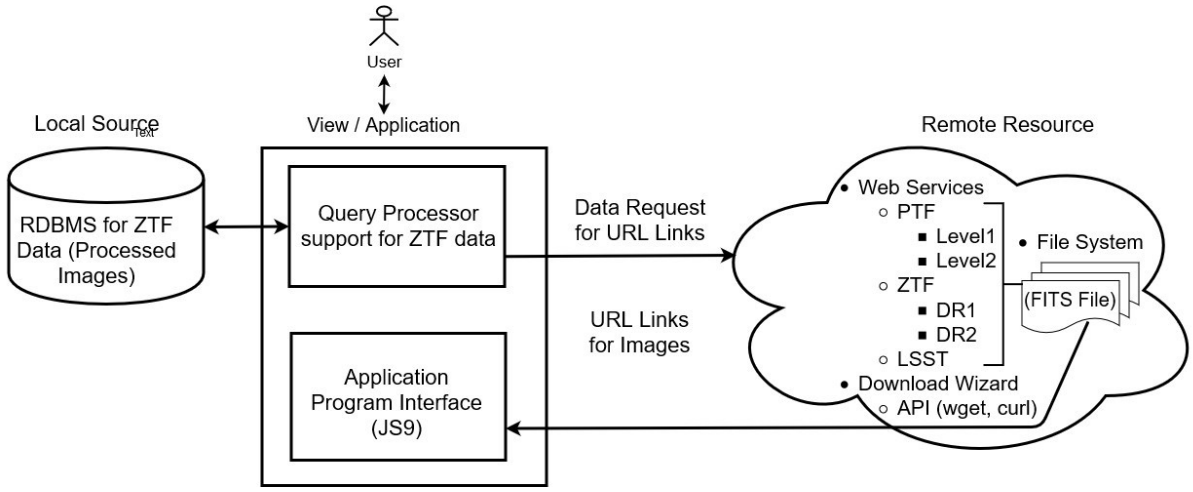

Figure 4.5: Proposed Web-Based Polystore System architecture [5]

numerous objects with a query based on its needs. Each time a user selects an object, a query is generated. Combining searches and linking several items makes the fundamental queries more complex. The term multi-stage querying refers to the process of adding queries in multiple stages [68]. The server then stores the query, and upon clicking the results button, the results are loaded into a table. The user can select the table content by clicking on the relevant result. The FITS image viewer JS9 is connected by selecting the desired result, and the images were displayed [67]. The interface for the query language implements a set-theoretic query language to resolve the relationships between objects of interest. It is based on SQL queries and is accessible to people with limited database programming experience.

4.3.2 Workflow Web-based Query Management System with Top-down approach

As depicted in Figure 4.6, the proposed web-based Information Requirements Elicitation (IRE) system has a workflow mechanism for user convenience and a query language that is simple to use. Using IRE, users can interact with a system to obtain objects of interest by generating queries. Users can choose objects via a graphical user interface (GUI) and assign a range of values to conduct this query within the IRE process. Whenever a value was assigned, and a search was conducted. The web application includes a visualizer, the

API named JS9 FITS viewer [67]. The JS9 image viewer API specifies the communication protocol between local and remote data storage. All ZTF data products are accessible via online (GUI-based) web tools and API services of NASA/IPAC Infrared Science Archive (IRSA), which can be accessed at https://irsa.irsa.caltech.edu/Missions/ztf.html. After transferring the data from the IRSA cloud to the web application via URL links, JS9 API helps visualize the requested images. Combining these searches and linking numerous objects makes the fundamental inquiry increasingly complex, as seen in Algorithm 2.

Algorithm 2: Query Workflow across the multiple data sources

```
i ← objects ;                                   // Nights, Fields, CCDS, ...
j ← attributes ;                                // nid, fid, ccdid, obj, ...
k ← attributes properties ; // nid=443,444.., fid=436, 836.., ccdid=0 16,
  ...
n ← number of objects
for ( i = 1; i <= n; i++ ) do
    if (k ϵ j_i) then
        R [ ] ← GeneratedImagesList
        satisfied ← TRUE
    else
        i ++ ;                                  // append with related object
    end
end
R [ ] ← GeneratedImagesList
r = 1
for each r in R [ ] do
    Select in R [r] ← SQL query is converted to server requests to obtain images
     from the IRSA cloud
end
```

4.3.3 Query Processor

The proposed web application includes an integrated query processor for mapping queries, converting data, and transferring them to a local database from a distant data store. Between the query processor, the web application, and the underlying database management system, SQL is utilized for communication. The Local RDBMS processes SQL query statements generated by the user's interaction with the system. The SQL query is then joined to the Processed Images (ProcImages) database, which includes serial numbers for each image. The query processor then compares the SQL query with the unique serial

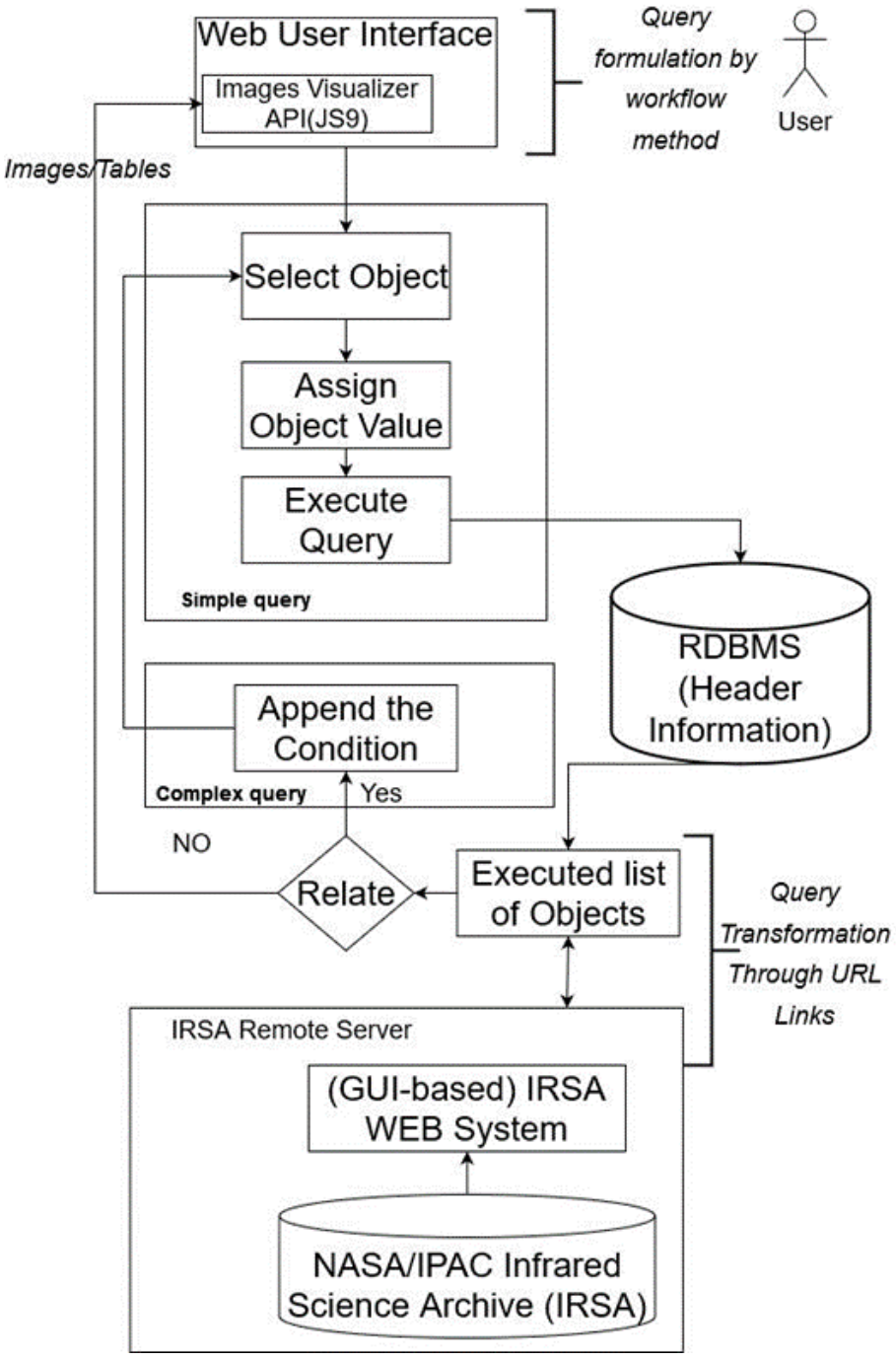

Figure 4.6: Workflow across the multiple data store

numbers of the images in the IRSA cloud, which are then translated into image URLs. The SQL query is then transformed to server requests to retrieve images from the IRSA data cloud, and matched images with the same unique serial number are displayed in the visualizer of the web application. In addition, the query system permits users to link and mix objects to construct a multi-object search. A query is generated each time a user picks an object. Combining searches and linking several items makes the fundamental queries more complex.

- Query 1: Find the image information from a field where the user selects example records from the object list (e.g fields, nights, exposures, procimages, ccd, etc.)
  SQL for Query 1:

```
select distinct on (A."DBFIELD") A.* from "_FIELDS" A\\
```

  Image SQL for Query 1:

```
select A.*,B.* from "_Exposures" A, "_PROCIMAGES" B,
(select distinct on (A."DBFIELD") A.* from "_FIELDS" A)
C where A."DBFIELD" = C."DBFIELD" and A."DBRID"= B."DBRID"
order by B."DBPID" offset 0 limit 10
```

- Query 2: Find the information of image from a certain place field and exposures.
  SQL for Query 2:

```
select distinct on (A."DBEXPID") A.* from "_exposures" A,
(select distinct on (A."DBFIELD") A.* from "_FIELDS" A)
B where A."DBFIELD"=B."DBFIELD"
```

  Image SQL for Query 2:

```
select A.*,B.* from "_PROCIMAGES" B,
(select distinct on (A."DBRID") A.* from "_exposures" A,
(select distinct on (A."DBFIELD") A.* from "_FIELDS" A)
```

```
B where A."DBFIELD"=B."DBFIELD" ) A where A."DBEXPID"= B."DBEXPID"
order by B."DBPID" offset 0 limit 10;
```

- Query 3: Find the information of image where field, night exposure is exactly the same in the tables.

  SQL for Query 3:

```
select distinct on (A."DBNID") A.* from "_NIGHTS" A,
(select distinct on (A."DBEXPID") A.* from "_EXPOSURES" A,
(select distinct on (A."DBFIELD") A.* from "_FIELDS" A)
B where A."DBFIELD"=B."DBFIELD" ) B where A."DBNID"=B."DBNID" ;
```

  Image SQL for Query 3:

```
select A.*,B.* from "_EXPOSURES" A, "_PROCIMAGES" B,
(select distinct on (A."DBNID") A.* from "_NIGHTS" A,
(select distinct on (A."DBEXPID") A.* from "_EXPOSURES" A,
(select distinct on (A."DBFIELD") A.* from "_FIELDS" A)
B where A."DBFIELD"=B."DBFIELD" ) B where A."DBNID"=B."DBNID" )
C where A."DBNID" = C."DBNID" and A."DBEXPID"= B."DBEXPID"
order by B."DBPID" offset 0 limit 10;
```

The proposed system can execute simple queries, such as Queries 1 and Queries 2 and sophisticated questions such as Query 3. Given the current status of ZTF data access, current research can concentrate on establishing an alternative top-down workflow web-based query management system for ZTF data access [5].

4.3.4 Querying in Time-domain Astronomy

Domain experts in astronomy require query tools to access various types of data on several astronomical entities. The most common searches were for images and image-related information. To acquire information, users may wish to query a single object or

a collection of objects. Depending on the circumstances, the user can request various requests. The user may also include specific logical operators (and, or) between multiple objects to refine the query further. Other known techniques for accessing PTF data require users to build complicated programs to execute complex queries.

The proposed system permits the formulation of inquiries using an interactive system in which users may select objects and enter their associated IDs and predefined object names. The user can relate numerous objects according to query criteria. Each time a user picks an object when querying, the query is generated. Combining searches and linking several items makes the fundamental queries more complex. The term multi-stage querying refers to the process of adding queries in multiple phases. The query is then saved on the server, and the information is returned to the table when the results button is clicked. The system connects to the FITS image reader and enables image visualization by selecting an item from a database [5].

- Query 1: Find the information of image where Fields ID is 00836 and CCDID is 11.
  SQL for Query 1:

  ```
  SELECT *FROM exposures WHERE ccdid = 16 OR field = 00836;
  ```

- Query 2: Find the images information from a certain place where the Field ID is 00424 and ccd id is 11.
  SQL for Query 2:

  ```
  SELECT field, ccdid, qid, rcid, pid FROM exposures where
  ccdid = 11 and field = 424;
  ```

- Query 3: Find the information of image where field and CCDID is exactly the same in the tables “CCD”, ”FIELD” and “EXPOSURES”.
  SQL for Query 3:

  ```
  SELECT a.fid, b.ccdid FROM exposures AS a, ccds AS b WHERE
  a.fid = b.ccdid;
  ```

4.4 Summary

There are numerous types of astronomy data and the amount of data available in repositories, such as PTF, iPTF, ZTF, and LSST. We used ZTF repository data to create databases and unify them in a common query language. Thus, we proposed a workflow web-based Polystore system architecture based on a top-down approach instead of a bottom-up approach system that prioritizes language translation. The query processor unifies queries from various databases into a common relational query language by processing the queries in the proposed system. This system is capable of querying multiple objects using the workflow method, because a query is generated via GUI interaction. The results are presented in tables and images. When an object is queried, the system generates two SQLs, one for the graphical user interface and the other for the FITS image viewer (visualizer) for linked open data in the ZTF. The image SQL connects to a remote image database, which is then converted into server URL requests to retrieve images and display them in the visualizer via the API. In addition, current top-down and bottom-up approach-based systems, and the latest approach, Polystores, are addressed. We compared the Polystore system to existing systems and compared the features of the proposed system with those of Polystore systems [5] [4].

Chapter 5

RESULTS AND DISCUSSION

5.1 Comparison with Existing Systems

The massive amount and variety of data available in the astronomical domain presents a formidable management challenge. Relational databases are utilized by most current astronomical solutions. Moreover, most solutions require users to wite complex programs to collect valuable insight. This study proposes a solution to the problems associate with big data and the lack of a suitable query tool for astronomy.

- Provide an optimal multi-database architecture to manage heterogeneous data in astronomy, as discussed in Section 4.3.1 System Overview.
- Provide a query language that will federate the information, transform, and effectively migrate data within the underlying data stores, as discussed in Section 4.3.3 query processor.
- Manage heterogeneous data via a fully automated workflow based query management system as discussed in Section 4.3.2.
- Minimize the local execution time in the data stores, by pushing down select operations in the data store sub queries and exploiting the bind join by query rewriting.
- Minimize global execution time by operator ordering.
- Minimize communication cost and network traffic by reducing data transfers between nodes.

Additionally, the solution should preserve the data integrity compared to the original source of the ZTF data. We consider implementing the data set of Data Release 1. Multiple criteria, including data support, query support functions, architecture flexibility, and users, can be used to evaluate the query management system. Query management systems focus on providing a uniform/single query language from multiple heterogeneous data sources with varying similarities and differences.

An evaluation of the previous work is presented in Table 5.1. We compared the IRSA ZTF Images GUI and Datawnt0 GUI to the GUI of our proposed systems. Unlike IRSA, Datawnt0 lacks data support for reference images and catalog files, despite containing features such as multi-object search and supporting relate and join operations. The proposed system has few parameters in common with IRSA, such as providing data support for reference images and catalog files, and a filtering option in the result table. However, the proposed system has additional parameters, such as QBE support, and can be used by novice users (who lack SQL knowledge), unlike the IRSA and Datawnt0.

5.2 Comparison with others Polystore Systems

Data archives have dealt with various data models and storage engines in their native formats over the past several decades. The data sets were utilized without being converted into a standard data model. Recent advancements in Polystore systems adhere to a bottom-up methodology in which incoming data from the source environment is the basis for information processing. As shown in Figure 5.1, the bottom-up approach to creating Polystore emphasizes language translation as the primary task.

The features of the proposed system were compared to those of BigDAWG and CloudMdsSQL, which are existing systems. The result of this evaluation are presented in Table 5.2. Instead of using islands/shims or mediators/wrappers to manage heterogeneity and multiple data stores, the proposed system used APIs to process queries. The proposed system uses native API calls, such as BigDAWG or CloudMdsSQL, to achieve autonomy, rather than wrappers. The proposed system specifies no data repositories because it aims to achieve ultimate transparency. In contrast to existing systems, which require the speci-

Table 5.1: Evaluation based on Existing Work

Evaluation framework	**IRSA ZTF Images GUI**	**Datawnt0 GUI (Past work)**	**Proposed systems GUI**
Data support	• Reference images and catalog files • Epochal science images and catalog files	• Epochal science images and catalog files	• Reference images and catalog files • Epochal science images and catalog files
Query Support Function	• Single Object Search • Multi-object Search user upload manually predefined table • Querying by Fields, CCDs, and Galaxies name • No Relate and Join Function • Filtering option present in the result table • No query by Example Function	• Single Object Search • Multi-object Search • Query by Fields, CCDs, Nights • Support Relate and Join Function • Filtering option is not present in the result table • No Query by Example Function	• Single Object Search • Multi-object Search • Query by Fields, CCDs, Nights, Galaxies name • Support Relate and Join Function • Filtering option present in the result table • Support Query by Example Function
Architecture Flexibility	• Local: Catalogs and images • Remote: Multiple directories with images and logs files	• Local: Catalogs with index • Remote: Single directory for images accessible	• Local: Catalogs with index • Remote: Multiple directory for images accessible
Users	• Expert SQL users	• Amateur SQL users	• Novice users

fication of data stores or information islands, the proposed method automatically switches to a table representation and hides it. Although the schema can be manually updated and a standard and QBE SQL define the workflow, the proposed system lacks schema flexibility. Active query transformation with user-defined data migration is proposed to enhance the BigDAWG's data transformation and active data migration capabilities. CloudMdsSQL does not support these features [69].

5.2.1 Existing Bottom-up design Polystore Systems

Language translation is regarded as the most important task in a bottom-up approach for creating a Polystore. The existing data sets contain vast amount of data from numerous sources. L1, L2, and L3 are examples of languages that can be used to access these diverse data sets, as shown in Figure 5.1. These might or might not have employed SQL. If a large amount of data come from various sources and environments that support different languages and schemes, the bottom-up method is not recommended. Consequently, the data and information in these systems create database connectivity and compatibility issues [4].

Using a bottom-up methodology, HYBRID.POLY analytical Polystore was developed to manage this type of data. HYBRID.POLY is a platform for storing, analyzing, and gaining access to many heterogeneous data sets. The in-memory storage engine supports many data models, and the query interface of HYBRID.POLY accepts queries written in a hybrid language, which, as a superset of SQL, can generate complex analytical queries on non-relational data (JSON, XML, media files) and relational data. The hybrid optimizer optimizes queries and can process large queries with numerous nodes. The HYBRID.POLY query processing engine includes a query parser, a query compiler, and a query optimizer. HYBRID.POLY has a single combined data storage capable of storing any data, which improves its performance by decreasing query processing and communication costs [70].

According to [71], BigDAWG has multiple data sets in the medical domain MIMIC-II, which is an advantage of tightly coupled Polystore. MIMIC-II contains numerous types

Table 5.2: Evaluation based on features of Polystore systems

Evaluation framework	**BigDAWG**	**CloudMdsQL**	**Proposed system**
Heterogeneity	• Multiple Data Stores with Multiple Query Interfaces • Query Processed through Islands and Shim Operators	• Multiple Data Stores with Single Query Interface • Query Processed through Mediators/Wrappers	• Multiple Data Stores with Single Query Interface • Query Processed through APIs
Autonomy	• Use of Shims, where catalog information updated automatically • Multi-object Search user upload manually predefined table • Native API calls	• Use of Wrappers where catalog updated automatically and manually • Native API calls	• Use of APIs where catalog updated automatically and manually • Native API calls
Transparency	• Specify information islands with SCOPE operators and hide transformation detail with CAST operators	• Specify data stores, data types and automatic transformation into table representation	• No need to specify data store, automatic transformation into table representation which is hidden
Flexibility	• No schema flexibility • Query interfaces of fixed islands, not readily extensible	• No schema flexibility • Subquerying and user defined MFR functions	• No schema flexibility (Can be updated manually) • User defined workflow with standard and QBE SQL
Optimality	• Query Rewriting, data transformation and active data migration	• No active transformation and migration	• Active query transformation with user defined data migration

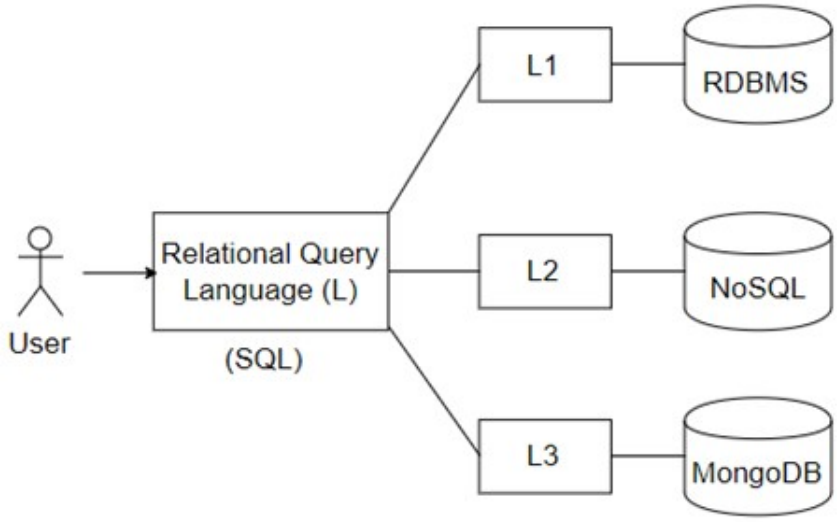

Figure 5.1: Bottom-up design Polystore System

of data, including patient metadata, physician and nurse notes, and lab results. Consequently, this system must support various data types, including standard SQL analytics, complex analytics, text search, and real-time monitoring of data stored in SciDB, Postgres, S-Store, and Apache Accumulo. These databases migrate data via casts, whereas shim translates queries from the island to the database [71].

5.2.2 Top-down design for Web Polystore Systems

As shown in Figure 5.2, the workflows query over the database engines is organized using the relational query language in the top-down approach. The IRE Workflow has been demonstrated to query various databases using the data retrieval API. The function of the API is to retrieve data for input key values. This is the most basic form of the component workflow for the databases and data sets being accessed. Many types of querying exist, such as QBE (Query by Example) Workflow or set-theoretic toolkits, which serve as informative toolkits for users in the astronomical domain by saving time when writing complex queries. The idea is to unify queries across multiple data models to manage data more efficiently. The polystore database system serves as the foundation for this workflow. Compared with the bottom-up approach, the top-down approach is superior because most components can be customized for simplicity. All information is

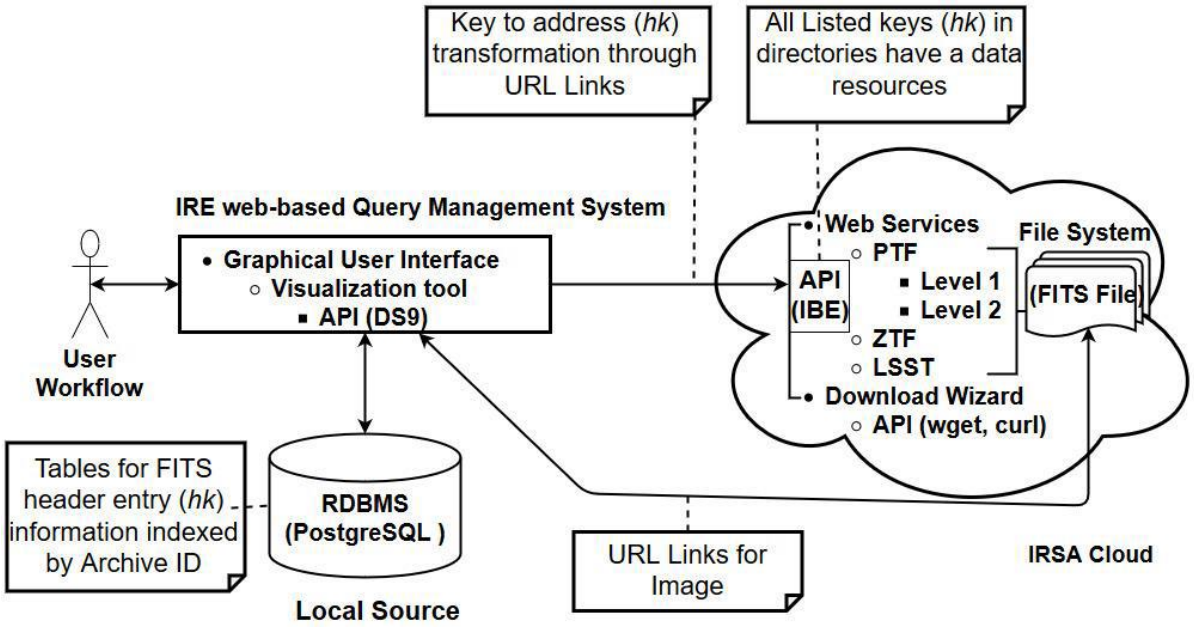

Figure 5.2: Web Polystore System top-down design approach

distributed, allowing it to be queried easily [4].

PostgreSQL was used to query the IRSA server data sets containing all PTF data information. The query is unified on the IRSA server and the local server in the proposed system. It contains cloud data directories (for raw PTF images, ProcImages, and other data) and PTF data information (image header information). Managing all the data on servers as databases with language translation would have been a time-consuming task that would have reduced efficiency and performance. Consequently, the proposed query system focuses on efficiently managing heterogeneous data distributed across multiple storage engines.

The main advantage of the proposed system is that it can perform recursive queries without no data loss. RDBMS headers can support both QBE workflow and set-theoretic toolkit. The QBE workflow employs visual tables from which the user can select conditions, enter commands, or select objects. Other operations such as inserting, updating, and deleting are also possible. This workflow converts the action of a user into a set of SQL-like commands.

A set-theoretic workflow will be a useful tool for astronomers because it can perform all set-theoretic operations such as union, intersection, and joins. This is more like a relational

algebra toolkit, simplifying the work of dealing with heterogeneous data. Consequently, this top-down approach can support a calculator-like tool in which sequences can be saved and used to generate results. The proposed system can be easily scaled by integrating various distributed databases. The query processor unifies queries from different databases into a common relational query language, thus making multiple heterogeneous data sets easily accessible through design uniformity. These are easily accessible owing to the growing number of new user workflows. Consequently, query conversion was not required in this approach.

5.3 Experimental Setup

We chose 20 queries for the experiment and compared the current system with the IRSA web-based system to assess the state of querying and analysis of large-scale astronomical data. The evaluation confirms that the current query system can handle more queries and may be useful for novice users unfamiliar with the query languages. Query by positions, query by observation date and time, query by host galaxies name, query by camera details, and query by example features are the most popular queries.

- Query by position: Uses galactic coordinates to specify the exact position to map the exact fields of the galactic plane. Find all objects in a certain galactic position.
- Query by observational date and time: Uses built-in calendar input function (OBJ-DATE) details and Night details, which include the date (DD:MM:YYYY) and time (HH:MM:SS) per observed astronomical body. Find all objects within a certain time period.
- Query by host galaxies: Uses a target search for catalogs of nearby galaxies Find all the objects related to the specific galaxies.
- Query by camera details: Uses 16ccds cameras as per the different object filters used by ZTF, namely, zg, zr, and zi for exposure acquired in g,R, I, respectively,

and bi, dk for bias and dark images, respectively. Find all objects from the camera filters. Find all objects from the camera name.

5.4 Query Comparison Analysis

ZTF DR1 data were used and analyzed to evaluate the workflow-based query system. This data set includes images, metadata containing image header information, and relationships. The metadata and relations were saved in the local Postgres database, as stated in Section. We downloaded the data, created a schema, and created 20 queries for the performance evaluation, as shown below.

1. Find all the images where Fields ID = 841;
2. Find all the images where Exposures ID = 44316126;
3. Find all images from fields where OBSJD = 2458197.6612616;
4. Find all the images where Night ID = 443;
5. Find all the images where Host galaxies where HOSTTAG = m81;
6. Find all images by observation date between 2018-04-01 and 2018-04-30;
7. Find all the images where Filters = 2;
8. Find all the images with R-band filters = zr;
9. Find all the images with CCD ID = 16;
10. Find all the images with Night ID = 443 and Fields ID = 809;
11. Find all the images with Night ID = 443 and CCD ID = 5;
12. Find all the images with Night ID = 443 and filtercode = zg;
13. Find all the images with Field ID = 841 and Exposure ID = 44316126;
14. Find all the images with Field ID = 809 and filtercode zg;

15. Find all the images where Exposures ID = 44316126 and Filters ID = 2;
16. Find all the images from date 2018-04-01 and 2018-04-30 and Field ID = 841 and CCD ID = 5 with R-band filters;
17. Find all the images with Night ID = 443 and Fields ID = 809 and CCD ID = 12 with g-band filter;
18. Find all the images from the Fields table or exposures tables;
19. Find all the Science Exposures images where Host galaxies name = m81;
20. Find all the References Images and Science Images where Host Galaxies name = ngc 13.

To validate the queries, the proposed system and the source of the ZTF data (IRSA web system) were used. Queries may involve retrieving data from a single or multiple objects. We used all possible queries as examples, which were already predefined by the proposed system for users who were unfamiliar with query languages (for novice users). Figure 5.3 depicts a comparison of the query results.

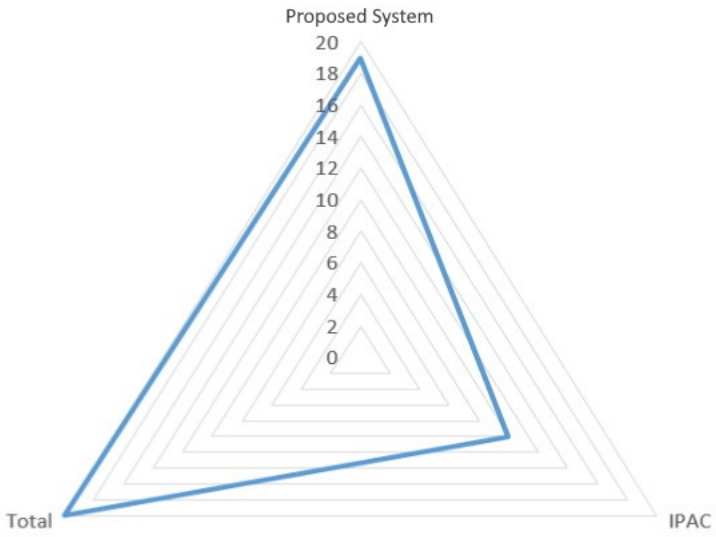

Figure 5.3: Query Comparison

Example of a Single Object Query (Q4): Find all changes from 2018-01-01 to 2018-12-31

```
Find all the images where Night ID = 443;
```

Process: The user selects the Nights object and provides the requested calendar information. When the search button was clicked, the results were tabulated. The user can view and/or download images of any search result.

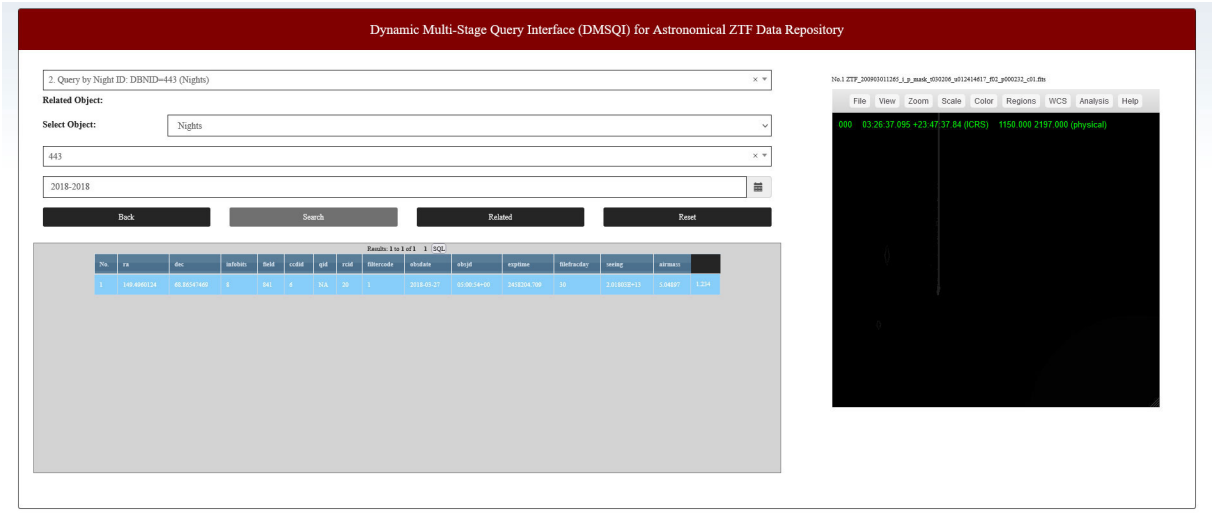

Figure 5.4: Workflow Web-based Polystore System of ZTF Archives

A polystore method for integrating massive heterogeneous PTF data is developed and demonstrated in Figure 5.4. A workflow-based query solution is described to assist users with simple database-like queries across IRSA services.

Twenty queries were evaluated using an example data set. A superior query performance was observed when compared with the existing system. A list of queries was distributed to colleagues and feedback was provided based on their observations. The feedback indicates that the proposed (or developed or whatever) system provided access to more queries. The interface was found to be easy to use. The feedback from user testing indicated that the usability of the system was enhanced because of search based on graphical data and metadata. The results table generated the metadata from the local database and the image data were then fetched from the cloud server by clicking a button.

Chapter 6

CONCLUSION AND FUTURE SCOPE

In the past few years, there has been a surge of interest in the database community for managing large amounts of unstructured data from disparate data stores. This problem has received special attention owing to the size of data, the rate at which data is added, and the emergence of new data types in various scientific data archives. Numerous heterogeneous data stores have developed as a consequence of the rise in big data. Although numerous models exist for integrating these data, it remains difficult to combine these enormous amounts of data into a single model. There is a demand for database management circles to manage large volumes of unstructured data originating from unrelated and unconnected sources.

Astronomy is also evolving into a science that is increasingly based on data processing, and involves a wide range of data. This information is retained in the domain-specific archives. Several astronomical studies have generated massive data archives. These archives were then made public as data repositories. These primarily consist of unstructured images and text, as well as data with certain structures, such as relations with key values. When archives are published as remote data repositories, it is difficult to organize the data in light of their increased diversity and to meet user information requests.

To address this issue, the a Polystore system was created to manage user workflows and visualize astronomical domain data using an integrated single query language. There are many different types of astronomy data, and the amount of data available in repositories such as PTF, iPTF, ZTF, and LSST. ZTF has linked open data available in FITS format,

and we created databases using data from ZTF repositories in order to unify them into a common query language. As a result, we proposed a workflow web-based Polystore system architecture that is top-down rather than bottom-up, with language translation as the primary task.

In the proposed system, the query processor unifies queries from different databases in a common relational query language. Because a query is generated through user interaction in the GUI, this system supports querying for multiple objects using the workflow method. The results are visualized in the form of tables and images. When an object is queried in the ZTF, the system generates two SQLs: one for the GUI and one for the FITS image viewer (visualizer). The image SQL is used to connect to a remote image database, which is then transformed into server URL requests to fetch images and display them in the visualizer via the API.

A method for managing a local data store and communicating with a remote cloud data store using a web-based query system is demonstrated. In addition, we addressed the current top-down and bottom-up approach-based systems, along with Polystores. We also evaluated the Polystore system against existing works and compared the features of the proposed system to those of Polystore systems.

In future work, we will attempt to test the performance of the proposed system and calculate the precision and recall. We also add the latest updated data to analyze the query requirements for managing multiple data stores.

Bibliography

[1] S. M. Coetzee *et al.*, "An analysis of a data grid approach for spatial data infrastructures," Ph.D. dissertation, University of Pretoria, 2009.

[2] R. G. Patidar, S. Shrestha, and S. Bhalla, "Polystore data management systems for managing scientific data-sets in big data archives," in *International Conference on Big Data Analytics*. Springer, 2018, pp. 217–227.

[3] JavaTpoint, *Three schema Architecture*, 2021, https://www.javatpoint.com/dbms-three-schema-architecture.

[4] M. Poudel, R. P. Sarode, S. Shrestha, W. Chu, and S. Bhalla, "Development of a polystore data management system for an evolving big scientific data archive," in *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*. Springer, 2019, pp. 167–182.

[5] M. Poudel, R. P. Sarode, Y. Watanobe, M. Mozgovoy, and S. Bhalla, "Processing analytical queries over polystore system for a large astronomy data repository," *Applied Sciences*, vol. 12, no. 5, p. 2663, 2022.

[6] Ontotext, *Data Integration*, 2022, https://web.cs.wpi.edu/~cs561/s12/Lectures/IntegrationOLAP/DataIntegration.pdf.

[7] A. Sen and A. P. Sinha, "A comparison of data warehousing methodologies," *Communications of the ACM*, vol. 48, no. 3, pp. 79–84, 2005.

[8] Ontotext, *Data Warehouse*, https://www.comp.nus.edu.sg/ lingtw /cs4221/ dw.pdf.

[9] A. P. Sheth and J. A. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM Computing Surveys (CSUR)*, vol. 22, no. 3, pp. 183–236, 1990.

[10] W. M. van der Aalst, "Federated process mining: Exploiting event data across organizational boundaries," in *2021 IEEE International Conference on Smart Data Services (SMDS)*. IEEE, 2021, pp. 1–7.

[11] T. Risch and V. Josifovski, "Distributed data integration by object-oriented mediator servers," *Concurrency and computation: Practice and experience*, vol. 13, no. 11, pp. 933–953, 2001.

[12] R. J. Miller, "Open data integration," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 2130–2139, 2018.

[13] Y. Zhang and Y. Zhao, "Astronomy in the big data era," *Data Science Journal*, vol. 14, 2015.

[14] M. Stonebraker, *The Case for Polystores*, 2015, http://wp.sigmod.org/?p=1629.

[15] J. Duggan, A. J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner, S. Madden, D. Maier, T. Mattson, and S. Zdonik, "The bigdawg polystore system," *ACM Sigmod Record*, vol. 44, no. 2, pp. 11–16, 2015.

[16] E. Bellm, "The zwicky transient facility," in *The Third Hot-wiring the Transient Universe Workshop*, vol. 27, 2014.

[17] P. Valduriez, *An overview of Polystores*, 2021, https://slideplayer.com/slide/13365730/.

[18] G. V. Pereira, M. A. Macadar, and M. G. Testa, "Delivery of public value to ultiple stakeholders through open government data platforms," in *Electronic Government and Electronic Participation: Joint Proceedings of Ongoing Research, PhD Papers, Posters and Workshops of IFIP EGOV and EPart*, vol. 22, 2015, p. 91.

[19] Ontotext, *What is Open Data?*, 2022, https://opendatahandbook.org/guide/en/what-is-open-data/.

[20] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data: The story so far," in *Semantic services, interoperability and web applications: emerging concepts*. IGI global, 2011, pp. 205–227.

[21] B. McBride, "The resource description framework (rdf) and its vocabulary description language rdfs," in *Handbook on ontologies*. Springer, 2004, pp. 51–65.

[22] SAS, *Big Data*, 2022, https://www.sas.com/en_us/insights/big-data/what-is-big-data.html.

[23] T. Segal, *Big Data*, 2022, https://www.investopedia.com/terms/b/big-data.asp.

[24] Ontotext, *What Are Linked Data and Linked Open Data?*, 2022, https://www.ontotext.com/knowledgehub/fundamentals/linked-data-linked-open-data/.

[25] M. Beno, K. Figl, J. Umbrich, and A. Polleres, "Open data hopes and fears: determining the barriers of open data," in *2017 Conference for E-Democracy and Open Government (CeDEM)*. IEEE, 2017, pp. 69–81.

[26] Y. Zhang and Y. Zhao, *Data mining in astronomy*, 2008, https://spie.org/news/1283-data-mining-in-astronomy?SSO=1.

[27] K. Chathuranga, *Big Data in Astronomy*, 07 2018.

[28] A. Szalay and J. Gray, "Science in an exponential world," *Nature*, vol. 440, no. 7083, pp. 413–414, 2006.

[29] P. J. Quinn, D. G. Barnes, I. Csabai, C. Cui, F. Genova, B. Hanisch, A. Kembhavi, S. C. Kim, A. Lawrence, O. Malkov *et al.*, "The international virtual observatory alliance: recent technical developments and the road ahead," *Optimizing scientific return for astronomy through information technologies*, vol. 5493, pp. 137–145, 2004.

[30] N. M. Law, S. R. Kulkarni, R. G. Dekany, E. O. Ofek, R. M. Quimby, P. E. Nugent, J. Surace, C. C. Grillmair, J. S. Bloom, M. M. Kasliwal *et al.*, "The palomar transient factory: system overview, performance, and first results," *Publications of the Astronomical Society of the Pacific*, vol. 121, no. 886, p. 1395, 2009.

[31] S. Kulkarni, "The intermediate palomar transient factory (iptf) begins," *The Astronomer's Telegram*, vol. 4807, p. 1, 2013.

[32] K. D. Borne, "Scientific data mining in astronomy," in *Next Generation of Data Mining*. Chapman and Hall/CRC, 2008, pp. 115–138.

[33] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, "Knowledge discovery in databases: An overview," *AI magazine*, vol. 13, no. 3, pp. 57–57, 1992.

[34] N. M. B. . R. J. Brunner, *Data Mining and Machine Learning in Astronomy*, 2010, https://ned.ipac.caltech.edu/level5/March11/Ball/Ball2.html.

[35] M. Poudel, R. P. Sarode, Y. Watanobe, M. Mozgovoy, and S. Bhalla, "A survey of big data archives in time-domain astronomy," *Applied Sciences*, vol. 12, no. 12, p. 6202, 2022.

[36] StatCodes, *Online statistical software for astronomy and related physical sciences*, 2005. [Online]. Available: https://astrostatistics.psu.edu/statcodes/

[37] P. S. University, *Statistical Analysis for the Virtual Observatory*, 2022. [Online]. Available: http://astrostatistics.psu.edu:8080/vostat/

[38] Weka, *Weka 3: Machine Learning Software in Java*, 2010. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/index.html

[39] AstroWeka, *Data Mining in the Virtual Observatory*, 2010. [Online]. Available: http://astroweka.sourceforge.net/

[40] bsipocz, *astroML*, 2022. [Online]. Available: https://github.com/astroML/astroML

[41] DAME, *Data Mining Exploration*, 2014. [Online]. Available: http://dame2.na.astro.it/

[42] Auton, *The Auton Lab*, 2021. [Online]. Available: https://www.autonlab.org/

[43] R. A. Shaw, F. Hill, and D. J. Bell, "Astronomical data analysis software and systems xvi," *Astronomical Data Analysis Software and Systems XVI*, vol. 376, 2007.

[44] M. Wenger, F. Ochsenbein, D. Egret, P. Dubois, F. Bonnarel, S. Borde, F. Genova, G. Jasniewicz, S. Laloë, S. Lesteven *et al.*, "The simbad astronomical database-the cds reference database for astronomical objects," *Astronomy and Astrophysics Supplement Series*, vol. 143, no. 1, pp. 9–22, 2000.

[45] *SMOKA Science Archive*, 2022, https://smoka.nao.ac.jp/.

[46] R. R. Laher, J. Surace, C. J. Grillmair, E. O. Ofek, D. Levitan, B. Sesar, J. C. van Eyken, N. M. Law, G. Helou, N. Hamam *et al.*, "Ipac image processing and data archiving for the palomar transient factory," *Publications of the Astronomical Society of the Pacific*, vol. 126, no. 941, p. 674, 2014.

[47] M. J. Kurtz, G. Eichhorn, A. Accomazzi, C. S. Grant, S. S. Murray, and J. M. Watson, "The nasa astrophysics data system: Overview," *Astronomy and astrophysics supplement series*, vol. 143, no. 1, pp. 41–59, 2000.

[48] *Science Data Center for Astrophysics Planetary Sciences*, https://www.ipac.caltech.edu/.

[49] Y. Xia, X. Yu, M. Butrovich, A. Pavlo, and S. Devadas, "Litmus: Towards a practical database management system with verifiable acid properties and transaction correctness."

[50] R. Han, L. K. John, and J. Zhan, "Benchmarking big data systems: A review," *IEEE Transactions on Services Computing*, vol. 11, no. 3, pp. 580–597, 2017.

[51] M. Poudel, S. Shrestha, R. P. Sarode, W. Chu, and S. Bhalla, "Query languages for polystore databases for large scientific data archives," in *2019 9th International*

Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2019, pp. 185–190.

[52] Oracle, *Data Warehousing Concepts*, 1999, https://docs.oracle.com/cd/A84870_01/doc/server.816/a76994/concept.htm.

[53] M. Stonebraker and U. Çetintemel, "" one size fits all" an idea whose time has come and gone," in *Making Databases Work: the Pragmatic Wisdom of Michael Stonebraker*, 2018, pp. 441–462.

[54] P. Kranas, B. Kolev, O. Levchenko, E. Pacitti, P. Valduriez, R. Jiménez-Peris, and M. Patiño-Martinez, "Parallel query processing in a polystore," *Distributed and Parallel Databases*, vol. 39, no. 4, pp. 939–977, 2021.

[55] Caltech, *Zwicky Transient Facility - Mission Characteristics*, 2021, https://www.ztf.caltech.edu/. [Online]. Available: https://irsa.ipac.caltech.edu/Missions/ztf.html

[56] F. J. Masci, R. R. Laher, B. Rusholme, D. L. Shupe, S. Groom, J. Surace, E. Jackson, S. Monkewitz, R. Beck, D. Flynn *et al.*, "The zwicky transient facility: Data processing, products, and archive," *Publications of the Astronomical Society of the Pacific*, vol. 131, no. 995, p. 018003, 2018.

[57] Caltech, *The intermediate Palomar Transient Factory*, 2021. [Online]. Available: https://www.ptf.caltech.edu/page/about

[58] R. Dekany, R. M. Smith, R. Riddle, M. Feeney, M. Porter, D. Hale, J. Zolkower, J. Belicki, S. Kaye, J. Henning *et al.*, "The zwicky transient facility: Observing system," *Publications of the Astronomical Society of the Pacific*, vol. 132, no. 1009, p. 038001, 2020.

[59] R. M. Smith, R. G. Dekany, C. Bebek, E. Bellm, K. Bui, J. Cromer, P. Gardner, M. Hoff, S. Kaye, S. Kulkarni *et al.*, "The zwicky transient facility observing system," in *Ground-based and Airborne Instrumentation for Astronomy V*, vol. 9147. SPIE, 2014, pp. 2294–2306.

[60] S. Schmidt, A. Malz, J. Soo, I. Almosallam, M. Brescia, S. Cavuoti, J. Cohen-Tanugi, A. Connolly, J. DeRose, P. Freeman *et al.*, "Evaluation of probabilistic photometric redshift estimation approaches for the rubin observatory legacy survey of space and time (lsst)," *Monthly Notices of the Royal Astronomical Society*, vol. 499, no. 2, pp. 1587–1606, 2020.

[61] Caltech, *NASA/IPAC INFRARED SCIENCE ARCHIVE*, 2021, https://irsa.ipac.caltech.edu/frontpage/.

[62] ——, *Zwicky Transient Facility - Public Data Release 2*, 2021. [Online]. Available: https://www.ztf.caltech.edu/news/public-data-release-2

[63] D. C. Wells and E. W. Greisen, "Fits-a flexible image transport system," in *Image processing in astronomy*, 1979, p. 445.

[64] Caltech, *ZTF API Queries*, 2021. [Online]. Available: https://irsa.ipac.caltech.edu/docs/program_interface/ztf_api.html#accesscontrol

[65] ——, *ZTF Graphical Interface*, 2022. [Online]. Available: https://irsa.ipac.caltech.edu/applications/ztf/?__action=layout.showDropDown&

[66] J. Samos, F. Saltor, J. Sistac, and A. Bardes, "Database architecture for data warehousing: an evolutionary approach," in *International Conference on Database and Expert Systems Applications*. Springer, 1998, pp. 746–756.

[67] JS9, *JS9: astronomical image display everywheret*, 2021, https://js9.si.edu/.

[68] A. Madaan and S. Bhalla, "Domain specific multistage query language for medical document repositories," *Proceedings of the VLDB Endowment*, vol. 6, no. 12, pp. 1410–1415, 2013.

[69] M. Poudel, S. Shrestha, W. Yilang, C. Wanming, and S. Bhalla, "Polystore database systems for managing large scientific dataset archives," in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2018, pp. 1–6.

[70] M. Podkorytov, D. Soderman, and M. Gubanov, "Hybrid. poly: An interactive large-scale in-memory analytical polystore," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 43–50.

[71] V. Gadepally, K. OBrien, A. Dziedzic, A. Elmore, J. Kepner, S. Madden, T. Mattson, J. Rogers, Z. She, and M. Stonebraker, "Version 0.1 of the bigdawg polystore system," *arXiv preprint arXiv:1707.00721*, 2017.