Data Warehouse and Multidimensional Analysis for Admission Data

Vijay Dev
Faculty of Science
Dayalbagh Educational Institute
Dayalbagh, Agra

Preetvanti Singh
Faculty of Science
Dayalbagh Educational Institute
Dayalbagh, Agra

Address for Correspondence
Preetvanti Singh
Faculty of Science
Dayalbagh Educational Institute
Dayalbagh, Agra

Data Warehouse and Multidimensional Analysis for Admission Data

Abstract

The development of a data warehouse has become essential as real-life problems nowadays deal with multidimensional data, and data warehouses provide a trustworthy foundation for decision-making in this situation. Thus, the goal of this research study is to present the development stages of a data warehouse for managing the admission process in higher education institutions. Fact and dimensions tables are arranged using the snowflake schema for the logical arrangement of the multidimensional database. The paper also presents the use of the data warehouse to create online analytical processing (OLAP) cubes and perform OLAP analysis on multidimensional admission data. The admission data warehouse and the OLAP cubes are implemented in PostgreSQL. The developed system will provide the university administrators with various reports that can be used for further analysis.

Keywords: Data warehouse, multidimensional analysis, admission data, OLAP cubes, PostgreSQL.

1 Introduction

With the growing scope of modern-day business, it is required to pull massive amounts of multidimensional data from all aspects of the business. However, it is time-consuming to organize and glean insights from this data. A multidimensional data model is a method for arranging this type of data in the database with better structuring and organization of the contents in it. This data model is used in organizations to generate interactive reports that can be used for imperative decision-making, and for allowing users to grill analytical questions associated with the business trends.

A data warehouse allows analyzing multidimensional data easily as the data is pulled from all different sources within the business and is stored in a single data repository. Data warehouse, the core of business intelligence, delivers business reports to mine insights from multidimensional data and provides analytics tools for monitoring business performance. It provides a mechanism for assembling the contents in the database, building historical data over time, and understanding relationships and trends across the data. This enables businesses to stay competitive by enabling them to extract insights from their data, and support decision-making.

Decision-makers in the education domain continually hunt for new technologies to turn students' academic data into knowledge for making the right strategic decisions. The admission process in higher education is complex and thus, admission processing software is required to support a range of features that can enable administrators to manage the entire student journey, bring in efficiency, and help in scaling the admissions. A data warehouse can be very helpful in the admission process for the stakeholders as it can facilitate analyzing the trends to understand the preferences and behavior of the student and determine the adjustments to be made for dealing with these changing trends. Therefore, the main objective of this paper is to design an admission data warehouse for managing the admission process in the higher education environment. The developed system will be used for:

- providing a single data source to users for multidimensional analysis,
- producing easy-to-understand reports about key data trends based on user specifications,
- building predictive models to deal with changing trends, and
- identifying the business health based on current and historical data.

2. Background

A data warehouse is usually used to establish insights into the changing trends of market conditions, customer behavior and preferences, and company capabilities (Park & Kim, 2013). These have been efficiently used in different domains. Jo et al. (2022) developed machine learning-based prediction models to identify vision-threatening diabetic retinopathy in type 2 diabetes patients using medical data from a clinical data warehouse. Kaspar et al. (2023) explained the use of data warehouses in combining data from clinical and imaging systems of hospitals. A production picture archiving and communication system was integrated with a clinical data warehouse for querying the data from both domains using a single query. Liu (2021) created a model that used an agent-based method for extracting data, processing data, and querying data in a data warehouse system and performing real-time analysis. Fauzi, Pribadi & Alifi (2023) evaluated the usefulness of developing a spatial data warehouse to implement geographical data analysis and data visualization in the tourism sector using qualitative analysis. Singh & Dev (2023) overviewed the use of data warehouse and OLAP technologies to deal with multidimensional tourism data.

The use of the data warehouse in the education domain enables acquiring timely and accurate data related to its operations, managing this data efficiently, and analyzing this data to guide and improve its activities (Hamoud, Marwah, Alhilfi & Sabr (2021) and Zhang & You (2021)). Serasinghe, Jayakody, Dayananda & Asanka (2021) and Yu (2021) studied the use of developing a data warehouse to enable educationists to share useful information for decision-making among management. Bahaaudeen (2023) studied the impact of implementing a data warehouse in higher education institutions for efficient decision-making. Chen, Zhan & Tian (2023) developed a data warehouse for analyzing students' performance. The technology was used to integrate student performance data, enrolment data, and teaching plans and perform data analysis for excavating high-value information to optimize teaching strategies. Farhan, Youssef & Abdelhamid (2024) developed a model to enhance the processing of unstructured big data in a data warehouse. An extract–clean–load–transform layer was designed with an emphasis on text.

The representation of multidimensional data is better than traditional databases because these databases are multi-viewed, i.e., data can be analyzed from different perspectives. Multidimensional data models are workable on complex systems also. The development of a data warehouse has become essential as real-life problems nowadays deal with multidimensional data, and data warehouses provide a trustworthy foundation for decision-making in this situation. It delivers a richer business intelligence environment and enables efficient decision-making for business engineers to pull out all aspects of the business from the massive amounts of multidimensional data. A data warehouse in higher education institutions is used to collect and analyze data for improving the teaching and learning processes and for analyzing the performance of students. This system can be very useful in managing the admission process and performing multidimensional analysis to understand the preferences and behavior of the student and determine the adjustments to be made to deal with the changing trends. Moreover, the data warehouse provides a single data source to users for analysis and produces easy-to-understand reports about key data trends based on user specifications. Thus, an Admission Data Warehouse (ADW) is developed in this study to provide efficient multidimensional data analysis.

3. Methods

This section presents the methods to design the data warehouse and the OLAP cubes.

3.1 Data collection

The data for ADW is collected by taking the relevant data from the university database and other university sources. Data of 9 years (from 2015 to 2023) is considered to develop the system. The study takes the data of the applicants applying for B.Sc. from Dayalbagh Educational Institute (Deemed to be University), Agra, India. This data includes the demographic and academic data of applicants, papers opted by them for the entrance exam, and their program choices.

Part of the *University database* is used to create the admission data warehouse. The database is shown in Figure 1 as an entity-relationship diagram.

3.2 Designing the ADW

The steps involved in building ADW are (Kimball & Ross, 2011):

i. Identifying the Business process

To develop the ADW, the requirements were collected by personally interviewing the individual end users (members of the examination committee and administration of the Faculty of Science). The identified functional requirements of the system are listed below:

a. Pattern analysis of applicants applying to the university,
b. Pattern analysis of applicants appearing in the entrance examination, and
c. Trend analysis of the applicants applying in various courses.

Based on the identified functional requirements, a business process model is prepared and developed in the form of a business matrix as shown in Table 1. The columns of the matrix represent the common dimension used across the educational institute. The mark ‘X’ indicates the columns related to each row.

Table 1: Business Matrix

Business Processes ↓ / Dimensions →	Academic Date	Candidate	Program	Location	Faculty	Student
a. Pattern analysis of applicants applying to the university	X	X	X	X	X	
b. Pattern analysis of applicants appearing in the entrance examination	X	X	X	X	X	
c. Trend analysis of the applicants applying in various courses/stream	X		X	X	X	X

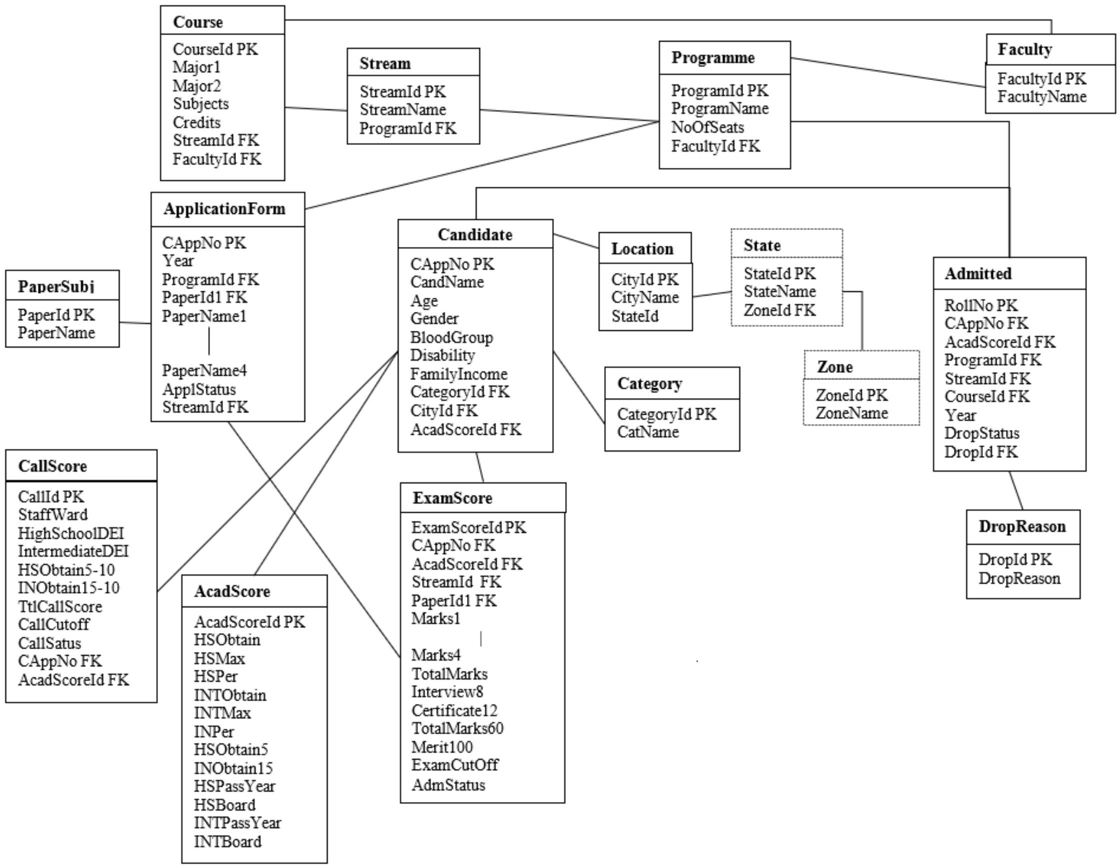

Figure 1. The entity-relationship diagram.

ii. Dimensional Modeling

Based on the above business processes, four fact tables are identified for the admission system that contains the foreign key column to allow joins with dimension tables and the measurement column for analysis. 8 dimension tables are also identified for the admission data warehouse. A dimension table stores the attributes that describe the data in the fact tables.

These dimension and fact tables that are arranged using the snowflake schema approach as shown in Figure 2. Snowflake schema is used as the dimensions had a long list of attributes thus these were split into sub-dimensions. Also, in this schema the dimension tables are normalized, thereby splitting the data into additional tables. The advantage of this is that redundancy is reduced, it requires small savings in storage space, and is easy to update and maintain.

iii. Declaring the Grain

This step is used to specify the atomic level granularity for each fact and dimension table. In this process, measures are identified along with the level of detail as shown in Table 2 and Table 3.

Table 2. Fact Table granularity.

Facts	**Grain**
FactLocation	1 row per applicant per zone per year (Applicants registered in a program year-wise)
FactLocationAdm	1 row per applicant admitted per zone per year (Applicant admitted to the program year-wise)
FactPerform	1 row per applicant per performance per year (Applicants registered in a program year-wise)
FactCourse	1 row per applicant admitted per course per year (Applicant admitted to the program year-wise)

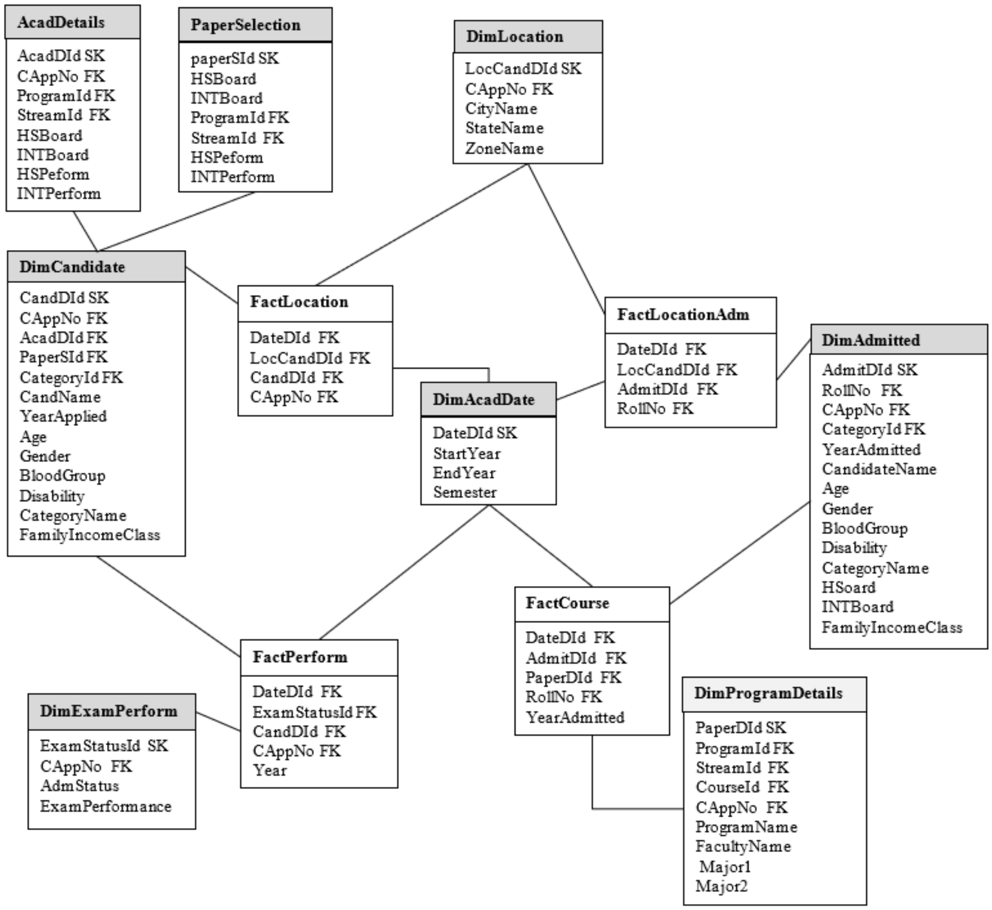

Figure 2. The admission data warehouse.

Table 3. Dimension Table granularity.

Dimensions	**Atomic Grain**	**Data Type**
Dimension Tables		
DimAcadDate	1 row per academic session	Academic session hierarchy
DimAdmitted	1 row per applicant admitted per year	Details of applicants admitted
DimCandidate	1 row per applicant per year	Applicant details
DimExamPerform	1 row per applicant performance per yr	Performance of applicants in written test
DimLocation	1 row per applicant location per year	Applicant location details
DimProgramDetails	1 row per program per year	Program hierarchy
Sub-Dimension Tables		
AcadDetails	1 row per applicant academic details per year	Applicant academic details
PaperSelection	1 row per applicant selected paper per yr	Applicant written test paper selection details

iv. The ETL Process

The extract, transform, and load (ETL) process for this study includes:

a. *Identifying the data sources:* The sources were identified by:
   - Listing every fact needed for analysis in fact tables,
   - Listing attributes for each dimension table,
   - Finding the appropriate source data item for each target data item,
   - Determining the default values, and
   - Searching the source data for the missing values. For example, searching for the location of applicants or their preferred written test paper options.

b. *Data Transformation:* The tasks involved in data transformation are:
   - Format revision: The data types and the lengths of individual data fields were revised. For example, the formats of addresses and emails were revised to suit the needs of the study.

- Decoding the fields: As the data items were taken from the application of the applicants, these were described by different field values. Thus the fields were coded for standardization. For example, codes female were mapped to ‘F’ and male to ‘M’ throughout.
- Splitting of fields: As an example, the address was stored in a single large text field. These were split into city, state, and zone.
- Key Restructuring: While extracting the data from the data source, the surrogate keys are formed for fact and dimension tables. For dimension tables, system-generated keys with no inherent meanings were generated as surrogate keys. The primary of the fact table is a concentrated key, that is the combination of surrogate keys of all participating dimension tables. For example, the dimension table *DimCandidate* has *CAppNo* as the primary key and *CandDId* as a surrogate key.

c. *Data Loading:* After extraction and transformation of data, the data is stored in the ADW repository by populating all the data warehouse tables for the first time (i.e., initial loading). The dimension tables are loaded first then the fact tables are loaded, and indexes on these tables are created. Then the ongoing changes are applied periodically.

v. *Defining the Metadata*

The example of metadata definition in the context of this study is given in Table 4.

Table 4. Metadata definition for the data element student DOB in *DimCandidate* Table.

Metadata item	Example
Field length	8
Element type	Date / Datetime
Permitted values	Should not be earlier than the date that would result in the student being aged between 17 to 21.
Translations	The date in the form 20030109 (i.e., in the YYYYMMDD format) is translated to 09012003 in the DDMMYYYY format in the target system.
Source	Applicant Application Form
Target	May be used in data verification audit processes.
Meaning	The day, month, and year an individual was born.
Restrictions	Only users with access to individual applicant data are permitted to view the DOB.
Limitations	This item does not reflect the student's grade level.
Operations	The element is used to calculate the age of a Applicant.
Purpose/rationale	To determine the age of a Applicant.
Owner	Database Admin
Treatment	Birthdates entered using a different format are changed to DD/MM/YYYY.
History	Once entered, the element is not changed for an individual applicant.
Retention	After the applicant has exited the university.
Security/confidentiality	This is a confidential record.
Identity	Everyone has one birthdate on record.
Accuracy	Audited once after original entry.
Completeness/sparsity	All current session records loaded contain values for this field.
Value set	Records loaded contain values within the domain of permitted values.

3.3 Preparing Admission OLAP Cubes

The Admission OLAP cubes are developed by aggregating facts over dimensions of ADW. These data cubes are used to get the reports of applicants from different views like *#applicants according to their family income class, #applicants according to their caste*, *#applicants location-wise*, *#applicants according to their performance*, and *#applicants in various courses*. These reports will be used to perform multi-dimensional analysis.

4. The ADW

The ADW is developed to provide efficient decision-making capabilities in analyzing multidimensional admission data. The data warehouse emerges as a repository of multidimensional data that:

a. delivers a richer business intelligence environment,
b. enables efficient decision-making for business engineers to pull out all aspects of the business

from the massive amounts of multidimensional data, and

c. improves the timeliness and quality of information and helps managers to better understand the data.

The developed data warehouse will provide decision support in performing multidimensional trend analysis of applicants applying to DEI.

4.1 The framework of ADW

The framework of ADW is designed to get correct, valid, and in-time data that can be efficiently transformed into decision information. The components of the framework are (Figure 3):

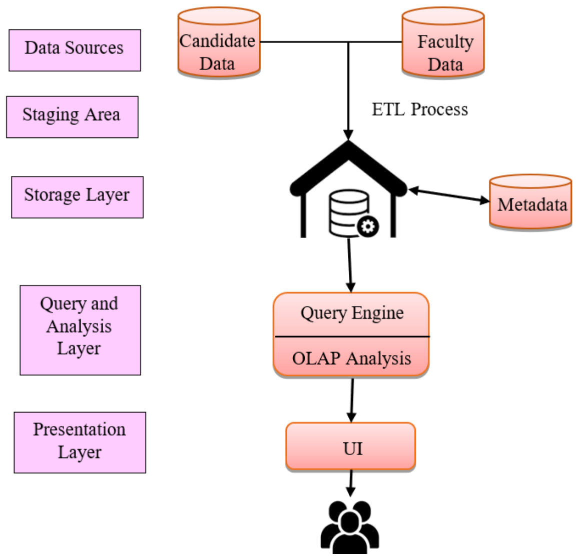

Figure 3. ADW Framework.

i. *Data Sources:* This component is the back end of the ADW system. Data is first processed in an online transaction processing environment and is then stored in the university database (see Figure 1).

ii. *Data Storage*: After performing the ETL process in the staging area, data is stored in ADW for future analysis in the form of dimension and fact tables that are informational and decision-support oriented. Metadata is created to understand and locate data in ADW.

iii. *Query and Analysis Layer*: This component consists of tools for analyzing and querying the data to provide educationists with direct and interactive access to data.

iv. *Presentation Layer*: The layer provides a business-friendly interface.

4.2 The Fact and Dimension Tables

Dimension Tables

The dimension tables include all the specific records about the admission process and are the main sources of information concerning the processes carried out during the admission process. The dimension tables contain surrogate keys that serve as primary keys of these tables. Dimension tables also contain the primary key of the database tables as foreign keys to establish the relation with the dimension tables. The dimension tables in this study are as follows:

1. DimAcadDate

The *DimAcadDate* dimension table serves all the dimension tables, as the admission process is time-spanned. The attributes of this table along with the explanation are given in Table 5.

Table 5. Structure of the *DimAcadDate* Table.

Attributes	Data Type	Description
DateDId	Integer	Surrogate key
StartYear	Integer	Session start year
EndYear	Integer	Session end year
Semester	Integer	Semester

2. DimCandidate

The *DimCandidate* dimension table stores the demographic data of applicants applying to DEI, particularly for science graduation in this study. This table is related to processes that include applicant data. The attributes of the table are shown in Table 6.

Table 6. Structure of the *DimCandidate* Table.

S.No.	**Attributes**	**Data Type**	**Description**
1.	CandDId	Serial	Surrogate Key
2.	AcadDId	Serial	Foreign key reference with *AcadDetails* table (see Figure 1)
3.	PaperSId	Serial	Foreign key reference with *PaperSelection* table (see Figure 1)
4.	CAppNo	Integer	Foreign key reference with *Candidate* table (see Figure 1)
5.	YearApplied	Integer	Year when the applicant applied for admission to DEI
6.	Age	Integer	Age of the applicant
7.	Gender	Char(1)	The gender of the applicant
8.	BloodGroup	Char(3)	The blood group of the applicant
9.	Disability	Char(1)	Whether the applicant is disabled or not
10.	CategoryId	Varchar(5)	Foreign key reference with *Category* table (see Figure 1)
11.	Categoryname	varchar(4)	Category of the applicant
12.	FamilyIncomeClass	varchar (20)	Family income class of the applicant

The family income class includes the economically weaker section (EWS), low-income group (LIG), middle-income group (MIG), and high-income group (HIG). The category includes general (GN), backward class (OBC), schedule caste (SC), and schedule tribes (ST).

The *DimCandidate* table branches out to *AcadDetails* and *PaperSelection* sub-dimension tables (Table 7). The table *AcadDetails* is a sub-dimension table of the *DimCandidate* table that stores the academic details of the applicant. The table *PaperSelection* is another sub-dimension table of the *DimCandidate* table that stores the program details for which the applicant has applied, as well as the details of the papers selected by him/her for the entrance examination.

Table 7. Structure of the sub-dimension tables.

Attributes	**Data Type**	**Description**
***AcadDetails* Table**		
AcadDId	Serial	Surrogate Key
CAppNo	Integer	Foreign key reference with *Candidate* table (see Figure 1)
HSBoard	Varchar(12)	High school board
INTBoard	Varchar(12)	Intermediate board
***PaperSelection* Table**		
PaperSId	Serial	Surrogate Key
CAppNo	Integer	Foreign key reference with *Candidate* table (see Figure 1)
ProgramName	Varchar (50)	Program in which the applicant has applied
StreamId	varchar (6)	Foreign key reference with *Stream* table (see Figure 1)
StreamName	Varchar (25)	Name of the Stream (Biology or Mathematics)
FacultyName	Varchar (35)	Name of the faculty
PaperId1	Varchar(5)	Foreign key reference with *PaperSubj* table (see Figure 1)
PaperName1	Varchar(30)	Title of Paper1, paper option for the written test chosen by the applicant
PaperId2	Varchar(5)	Foreign key reference with *PaperSubj* table (see Figure 1)
PaperName2	Varchar(30)	Title of Paper2, paper option for the written test chosen by the applicant
PaperId3	Varchar(5)	Foreign key reference with *PaperSubj* table (see Figure 1)
PaperName3	Varchar(30)	Title of Paper3, paper option for the written test chosen by the applicant
PaperId4	Varchar(5)	Foreign key reference with *PaperSubj* table (see Figure 1)
PaperName4	Varchar(30)	Title of Paper4, paper option for the written test chosen by the applicant

3. DimLocation

The location of applicants is stored in the *DimLocation* table. This information is used to define the location (city, state, and zone) concept hierarchy. The attributes of the table are as follows:

Table 8. Structure of the *DimLocation* Table.

Attributes	**Data Type**	**Description**
LocCandDId	Serial	Surrogate Key
CAppNo	Integer	Foreign key reference with *Candidate* table (see Figure 1)

Attributes	Data Type	Description
CityName	Varchar(20)	The city where the applicant lives
StateName	Varchar (20)	Their state
ZoneName	Varchar(50)	Zone wise description

4. *DimProgramDetails*

The *DimProgramDetails* table provides data for the programs offered by the faculties and the courses offered in each program. The courses offered in the BSc program are Physics-Mathematics Group (C04), Computer Science-Mathematics Group (C05), Mathematics-Chemistry Group (C06), and Mathematics-Economics Group (C07) for Mathematics stream students. For Biology stream students the courses are Zoology-Chemistry Group (C08), Zoology-Botany Group (C09), Botany-Chemistry Group (C10), Applied Botany Group (C11), and BSc Agriculture (C12). Such information can be useful in analyzing the most preferred and least preferred courses.Attributes with the description are shown in Table 9.

Table 9. Structure of the *DimProgramDetails* Table.

Attributes	Data Type	Description
PaperDId	Serial	Surrogate Key
ProgramId	varchar(5)	Foreign key reference with *Program* table (see Figure 1)
ProgramName	Varchar(50)	Description of Program like BSc, MSc
StreamId	Varchar(6)	Foreign key reference with *Stream* table (see Figure 1)
CourseId	Varchar(4)	Foreign key reference with *Course* table (see Figure 1)
Major1	varchar (20)	Major subject 1
Major2	varchar (15),	Major subject 2
FacultyName	Varchar(35)	Name of the Faculty
CAppNo	Integer	Foreign key reference with *Candidate* table (see Figure 1)

5. *DimExamPerform*

The data about the performance of applicants in written test and their admission status is stored in this table. This data is used to analyze the entrance exam performance of the applicants. The field *ExamPerformance* takes the values *outstanding*, *very good*, *average*, *poor*, and *very poor* depending upon their high school, intermediate, and written test performance.

Table 10. Structure of the *DimExamPerform* Table.

Attributes	Data Type	Description
ExamStatusSId	Serial	Surrogate Key
CAppNo	Integer	Foreign key reference with *Candidate* table (see Figure 1)
AdmStatus	Varchar (10)	Admission status of the applicant
ExamPerformance	Varchar (30)	Includes the performance of applicants in written test

6. *DimAdmitted*

The *DimAdmitted* dimension table stores the demographic and academic data of applicants admitted to DEI for science graduation. The attributes of the table are shown in Table 12.

Table 12. Structure of the *DimAdmitedd* Table.

Attributes	Data Type	Description
AdmitDId	Serial	Surrogate Key
RollNo	Integer	Foreign key reference with *Candidate* table (see Figure 1)
CAppNo	Integer	Foreign key reference with *Admitted* table (see Figure 1)
YearAdmitted	Integer	The year when the applicant was admitted to DEI
Age	Integer	Age of the applicant
Gender	Char(1)	The gender of the applicant
BloodGroup	Char(3)	The blood group of the applicant
Disability	Char(1)	Whether the admitted applicant is disabled or not
CategoryId	Varchar(5)	Foreign key reference with *Category* table (see Figure 1)
Categoryname	varchar(4)	Category of the applicant
HSBoard	Varchar(12)	High school board
INTBoard	Varchar(12)	Intermediate board
FamilyIncomeClass	varchar (20)	Family income class of the admitted applicant

The implementation of the above dimension tables is done in PostgreSQL as shown below.

```
create table DimLocation (
       LocCandDId serial primary key,
       CAppno integer,
       CityName Varchar(20),
       StateName Varchar(20),
       ZoneName Varchar(50) );

insert into DimLocation(CAppno, CityName, StateName, ZoneName)
select c.CAppno, l.CityName, st.StateName, z.ZoneName
from candidate c, location l, state st, zone z
where c.cityid =l.cityid
       and l.stateid = st.stateid
       and st.zoneid=z.zoneid;
```

(a) Query to create the dimension table *DimLocation*

```
create table DimAcadDate
(DateDId integer primary key,
       StartYear integer,
       EndYear Integer,
       Semester integer );
```

(b) Query to create the dimension table *DimAcadDate*

```
create table AcadDetails(
       AcadDId serial primary key,
       HSBoard Varchar(12),
       INTBoard Varchar(12),
       ProgramId varchar(4),
       StreamId varchar(6),
       HSperform Varchar(30),
       INTPerform Varchar(30) );

insert into AcadDetails(HSBoard, INTBoard, ProgramId, StreamId,
                           HSPerform, INTPerform)
select ac.HSBoard, ac.INTBoard, st.ProgramId, st.StreamId,
       case
              when ac.hsper >=90 then 'Outstand'
              when ac.hsper between 75 and 89.99 then 'VeryGood'
              When ac.hsper between 60 and 74.99 then 'Average'
              When ac.hsper between 50 and 59.99 then 'Poor'
              else 'VeryPoor'
end as HSperform,
              case
              when ac.intper >=90 then 'Outstand'
              when ac.intper between 75 and 89.99 then 'VeryGood'
              When ac.intper between 60 and 74.99 then 'Average'
              When ac.intper between 50 and 59.99 then 'Poor'
              else 'VeryPoor'
end as intperform
       from candidate c, AcadScore ac, applicationform a, stream st,
programme p
       where c.cappno=a.cappno
       and c.AcadScoreId = ac.AcadscoreID
       and a.streamid=st.streamid
       and st.programid=p.programid
```

(c) Query to create the sub-dimension table *AcadDetails*

```
create table PaperSelection(
      PaperSId serial primary key,
      CAppNo integer,
      ProgramName varchar (50),
      StreamId varchar (6),
      Streamname varchar(25),
      FacultyName varchar(35),
      PaperId1 Varchar(5),
      PaperName1 Varchar(50),
      PaperId2 Varchar(5),
      PaperName2 Varchar(50),
      PaperId3 Varchar(5),
      PaperName3 Varchar(50),
      PaperId4 Varchar(5),
      PaperName4 Varchar(50),
      Foreign Key (CAppNo) references Candidate (CappNo),
      Foreign key (PaperId1) References PaperSubj (PaperId),
      Foreign key (PaperId2) References PaperSubj (PaperId),
      Foreign key (PaperId3) References PaperSubj (PaperId),
      Foreign key (PaperId4) References PaperSubj (PaperId) );

insert into PaperSelection(CAppNo, ProgramName, StreamId, Streamname, FacultyName,
                        PaperId1, PaperName1, PaperId2, PaperName2, PaperId3,
                        PaperName3, PaperId4, PaperName4)
select c.CAppNo, p.programname, st.streamId, st.streamname, f.facultyname,
      a.PaperId1, a.PaperName1, a.PaperId2, a.PaperName2, a.PaperId3,
      a.PaperName3, a.PaperId4, a.PaperName4
from candidate c, programme p, stream st, ApplicationForm a, faculty f
where (c.cAppno= a.cAppNo
      AND p.ProgramId = a.ProgramId
      and p.facultyid = f.facultyid
      and a.streamid=st.streamid);
```

(d) Query to create the sub-dimension table *PaperSelection*

```
create table DimCandidate(
      CandDId serial primary key,
      AcadDId serial,
      PaperSId serial,
      CAppno integer,
      CandName varchar (50),
      YearApplied integer,
      Age Integer,
      Gender Char(1),
      BloodGroup Char(3),
      Disability Char(1),
      CategoryId Varchar(5) References Category(CategoryId),
      Categoryname varchar(4),
      FamilyIncomeClass varchar (20),
      Foreign key (AcadDId) References acaddetails(AcadDId),
      Foreign key (PaperSId) References paperselection(paperSId) );

insert into DimCandidate (AcadDId, PaperSId, CAppNo, CandName, YearApplied, Age,
                          Gender,BloodGroup, Disability, CategoryId,
                          Categoryname, FamilyIncomeClass)
select acd.AcadDId, pas.PaperSId, c.cappno, c.candname, a.year, c.age, c.Gender,
      c.BloodGroup, c.Disability, ct.CategoryId, ct.Catname,
      case
            when c.FamilyIncome >=120000 then 'HIG'
            when c.FamilyIncome between 50000 and 120000 then 'MIG'
            When c.FamilyIncome between 25000 and 50000 then 'LIG'
            else 'EWS'
end as FamilyIncomeClass
from candidate c, acaddetails acd, paperselection pas, applicationform a, category
ct
      where c.cappno = acd.cappno
      and c.cappno = pas.cappno
      and c.cappno=a.cappno
      and c.categoryid = ct.categoryid
```

(e) Query to create the dimension table *DimCandidate*

```
create table DimExamPerform(
      ExamStatusSId serial primary key,
      CAppNo integer,
      AdmStatus Varchar (10),
      ExamPerformance varchar(30),
      Foreign Key (CAppNo) references Candidate (CappNo)   );

insert into DimExamPerform(CAppNo, AdmStatus, ExamPerformance)
select c.CAppNo,  es.AdmStatus,
      case
            when es.Merit100 >=90 then 'Outstand'
            when es.Merit100 between 75 and 89.99 then 'VeryGood'
            When es.Merit100 between 60 and 74.99 then 'Average'
            When es.Merit100 between 50 and 59.99 then 'Poor'
            else 'VeryPoor'
end as ExamPerformance
from candidate c, ExamScore es
where c.cAppno= es.cAppNo;
```

(f) Query to create the dimension table *DimExamPerform*

```
create table DimProgramDetails(
       PaperDId serial primary key,
       ProgramId varchar(5),
       ProgramName varchar (50),
       Streamid varchar(6),
       COURSEID VARCHAR (4),
       Major1 varchar (20),
       Major2 varchar (15),
       FacultyName Varchar(35),
       Cappno integer );

insert into DimProgramDetails(programid, ProgramName, Streamid, CourseId,
                              Major1, Major2, FacultyName, CAppno)
select p.programid, p.programname, st.streamid, D.COURSEID, d.major1,
       d.major2, f.facultyname, ad.cappno
from programme p, faculty f, stream st, courses d, admitted ad
where p.facultyId = f.facultyId
       and p.programid=ad.programid
       and ad.streamid=st.streamid
       and ad.courseid=d.courseid
```

(g) Query to create the dimension table *DimprogramDetails*
Figure 4. Creating dimension tables for ADW

Fact Tables

The fact tables refer to the key processes followed during the admission process. ADW design consists of 4 fact tables: *FactLocation, FactLocationAdm, FactPerform*, and *FactCourse*. The fact table entities are the integer identifications of the dimension tables.

1. FactLocation

The *FactLocation* fact table helps in identifying the locations (zones) from where the applicants are coming. This table uses the dimension tables *AcadDate*, *DimCandidate*, and *DimLocation* with the keys of the table *FactLocation* as *DateDId*, *CandDId*, *LocCandDId*, and *CAppNo*. *CandDId* represents demographic information and *LocCandDId* identifies the zones of the applicants. To have the session information about applicants, the attribute *DateDId* is used. This fact table allows the analysis of the locations of applicants.

2. FactLocationAdm

The *FactLocationAdm* fact table helps in identifying the locations (zones) of the applicants admitted to DEI. The table uses the dimension tables *AcadDate*, *DimAdmitted*, and *DimLocation* with the keys of the table *FactLocation* as *DateDId*, *AdmitDId*, *LocCandDId*, and *RollNo*. *AdmitDId* represents the personal information of the applicants admitted and *LocCandDId* identifies the zones. This fact table allows the analysis of the locations of applicants admitted to DEI.

3. FactExamPerform

The *FactExamPerform* fact table makes it possible to store the records of applicant performance. The table uses the dimension tables *AcadDate*, *DimAdmitted*, and *DimExamStatus*. The foreign keys of the fact table are *DateDId*, *AdmitDId*, *ExamStatusId*, and *CAppNo*. *ExamStatusId* enables identifying the performance class of the applicants. The table allows for analyzing the time-based academic performance of applicants.

4. FactCourse

After being admitted to a program (science graduation in this study), the *FactCourse* fact table is used to store courses (major1 and major2) chosen by the applicants. The table uses the dimension tables *AcadDate*, *DimAdmitted*, and *DimProgramDetails* with the keys of the fact table as *DateDId*, *AdmitDId*, *PaperDId*, and *RollNo*. *PaperDId* identifies the courses chosen by the applicants. This fact table allows analysis of applicants course-wise for a session.

The implementation of the fact tables is also done in PostgreSQL as shown below.

```
create table factLocation
       (datedid serial,
        canddid serial,
        loccanddid serial,
        Cappno integer
       )

insert into factlocation(datedid, canddid, loccanddid, cappno)
       select dad.datedid, dc.canddid, dl.loccanddid, dc.cappno
       from dimacaddate dad, dimcandidate dc, dimlocation dl
where dc.cappno=dl.cappno
and    dc.yearapplied=dad.startyear and dad.semester = 1;
```

(a) Query for *FactLocation*

```
create table factLocationadm
       (datedid serial,
        Admitdid serial,
        loccanddid serial,
        RollNo integer
       )

insert into factlocationadm(datedid, canddid, loccanddid, rollno)
       select dad.datedid, da.admitdid, dl.loccanddid, da.rollno
       from dimacaddate dad, dimadmitted da, dimlocation dl
where da.cappno=dl.cappno
and    da.yearadmitted=dad.startyear and dad.semester = 1;
```

(b) Query for *FactLocationAdm*

```
create table factPerform
       (datedid serial,
        examstatusid serial,
        CappNo integer,
        Year integer
       )


insert into factPerform(datedid, examstatusid, CAppno, year)
select dad.datedid, des.examstatussid, dc.CAppno, dc.yearapplied
       from dimacaddate dad, dimcandidate dc, dimexamperform des
where dc.cappno=des.cappno
       and    dc.yearapplied=dad.startyear and dad.semester = 1;
```

(c) Query for *FactPerform*

```
create table factCourse
       (datedid serial,
        admitdid serial,
        paperdid serial,
        RollNo integer,
        YearAdmitted integer
       )

insert into factcourse(datedid, admitdid, paperdid, rollno, yearadmitted)
select dad.datedid, da.admitdid, dpd.paperdid, da.rollno, da.yearadmitted
       from dimacaddate dad, dimadmitted da, dimprogramdetails dpd
where da.rollno=dpd.rollno
and    da.yearadmitted=dad.startyear and dad.semester = 1;
```

(d) Query for *FactCourse*

Figure 5. Creating fact tables of ADW

5. Multidimensional trend analysis of the admission data

To perform multidimensional analysis, the admission data cubes are created in PostgreSQL. Various data cubes generated are as follows:

a. Gender data cubes

The *gender* data cube extracts the number of female and male applicants applying to DEI for the BSc program. The purpose of developing this cube is to get the applicants based on their family income class and the category (caste).

The *AdmGender* data cube extracts the number of female and male applicants admitted to DEI for the BSc program. This information can be useful in designing strategies for EWS and LIG students. It will also be useful for university administration to optimally allocate university resources.

```
select dc.yearapplied, dc.categoryname, dc.gender, dc.familyincomeclass,
        count(*) AS CandCount  into OLAPGender
from factlocation fc, dimacaddate dad, dimcandidate dc
where fc.datedid = dad.datedid
       and fc.cappno=dc.cappno
group by dc.yearapplied, dc.categoryname, dc.gender, dc.familyincomeclass
order by dc.yearapplied, dc.gender, dc.familyincomeclass
```

Figure 6. *Gender* data cube.

```
select da.yearadmitted, da.categoryname, da.gender, da.familyincomeclass,
        count(*) AS CandCount  into OLAPGenderAdm
from factlocationadm fca, dimacaddate dad, dimadmitted da
where fca.datedid = dad.datedid
       and fca.rollno=da.rollno
group by da.yearadmitted, da.categoryname, da.gender, da.familyincomeclass
order by da.yearadmitted, da.gender, da.familyincomeclass
```

Figure 7. *AdmGender* data cube.

b. Location data cubes

The *Location* data cube extracts the number of female and male applicants based on their locations and family income.

The *AdmLocation* data cube extracts the number of female and male applicants admitted to DEI based on their locations and family income. The information will be useful in analyzing the location of applicants to provide proper accommodation facilities for the applicants admitted to the program. This information can also help in determining the popularity of the educational institutes.

```
select dc.yearapplied, dc.gender, dc.familyincomeclass,
       dl.zonename, count(*) AS CandCount  into OLAPLocation
from factlocation fc, dimacaddate dad, dimcandidate dc, dimlocation dl
where fc.datedid = dad.datedid
       and fc.cappno=dc.cappno
       and dl.cappno=dc.cappno
group by dc.yearapplied, dc.gender, dc.familyincomeclass, dl.zonename
order by dc.yearapplied, dc.familyincomeclass
```

Figure 8. *Location* data cube.

```
select da.yearadmitted, da.gender, da.familyincomeclass, dl.zonename,
         count(*) AS CandCount  into OLAPLocationAdm
from factlocationadm fca, dimacaddate dad, dimadmitted da, dimlocation dl
where fca.datedid = dad.datedid
       and fca.cappno=da.cappno
       and dl.cappno=da.cappno
       group by da.yearadmitted, da.gender, da.familyincomeclass, dl.zonename
       order by da.yearadmitted, da.familyincomeclass
```

Figure 9. *AdmLocation* data cube

c. Performance data cube

The *Performance* data cube extracts the number of male and female applicants who were found eligible to give entrance exam of the BSc program with their performance in secondary school, senior secondary school, and the written test of BSc. The information will be used to analyze the academic performance of applicants based on their family income class.

```
select fp.year, dc.gender, dc.familyincomeclass, des.examperformance, count(*)
as candcount into olapperform
from factperform fp, dimcandidate dc, dimexamperform des, dimacaddate dad
where fp.year = dad.startyear
      and fp.cappno=dc.cappno
            and des.examstatussid=fp.examstatusid
      and fp.datedid=dad.datedid
group by fp.year, dc.gender, dc.familyincomeclass, des.examperformance
order by fp.year
```

Figure 10. Performance data cube.

d. Course data cube

This data cube extracts the number of male and female applicants admitted to the BSc program with their course preferences.

```
Select fco.yearadmitted, da.gender, da.familyincomeclass, dpd.courseid,
       count(*) as CandCount into OLAPCourse
from factcourse fco, dimacaddate dad, dimadmitted da, dimprogramdetails dpd
where fco.datedid = dad.datedid
      and fco.yearadmitted=da.yearadmitted
      and fco.rollno=da.rollno
      and da.cappno=dpd.cappno
group by fco.yearadmitted, da.gender, da.familyincomeclass, dpd.courseid
order by fco.yearadmitted
```

Figure 11. Course data cube.

OLAP Operations, drill down, and slice operations, are used next to perform the multidimensional trend analysis. For example, the following query displays the female applicants from the gender data cube based on their family income class year-wise using the slice operation.

```
select yearapplied, familyincomeclass, sum(candcount) as females
from olapgender
where gender ='F'
group by yearapplied, categoryname, familyincomeclass
order by yearapplied, categoryname, familyincomeclass
```

Figure 12. Slice operation using gender data cube.

6. Result and Discussion

The ADW deals with the data related to applicants applying and getting admission to a higher educational institute. It provides:

1. Applicant information, i.e., the demographic data, contact information, and academic details of applicants applying at a higher education institution.
2. Entrance Exam information, i.e., data related to entrance examinations like exam dates, exam scores, attendance details, and merit lists.
3. Faculty information including the data of its department, faculty members, and the programs offered.
4. Course information, i.e., data on the courses offered by the faculty, including course names, descriptions, prerequisites, and credits.
5. Applicant Enrollment information, i.e., data of applicants admitted to a higher education institution.

The design of the developed ADW is shown in Figure 2 (see Section 3). The following is a list of solutions offered by the designed warehouse:

a. Applicant-level facts: The ADW can answer questions at the most transactional level about the applicants registering in DEI and getting admission. For example, statistical analysis is performed to get the demographic information of applicants and a program-wise list of applicants.
b. Program-level facts: The ADW provides summarized program-level statistics like courses preferred by applicants in a program in a semester.
c. Performance-based facts: These provide the academic performance details such as test

scores, and high school and intermediate percentages.

Admission data for BSc in the faculty of Science, DEI was taken to implement the ADW. The data consisted of 12357 records for the period of 9 years (2015 to 2023). The multidimensional trend analysis of the data reveals the following facts.

Among the total of 12357 records, number of male applicants was 5995 and of female applicants was 6362. An increasing trend was observed in the number of applicants applying for the BSc program as illustrated in Figure 13. Number of female applicants applying in DEI was more than the number of male applicants. The graph also predicts the number of applicants applying to DEI for the next 4 years.

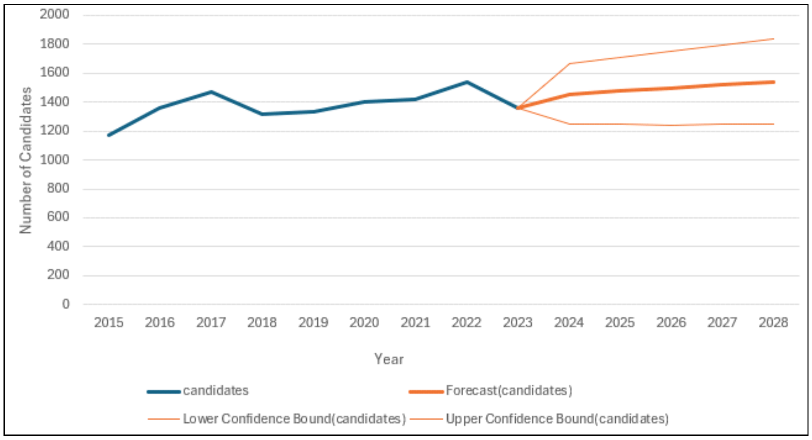

Figure 13: Number of applicants

The slice operation was performed on the gender data cube to analyze the data caste-wise. Most of the applicants who applied for these courses were of the general category (46.4% of 12357). 0.8% of applicants belonged to the ST category, 16.27% were SC applicants, and the remaining 36.4% were from the backward class.

Figure 14 shows the family income distribution of the applicants. It is observed that most of the applicants were from the MIG class (50.66% of 12357). 23.65 % of applicants were from the LIG class, 8.4% were from the EWS class and the remaining 17.29% were from the HIG class.

A total of 2416 applicants (out of 12357) were admitted to the BSc program in the past 9 years. Of these 2416 applicants, 1193 were female applicants and 1223 were males.

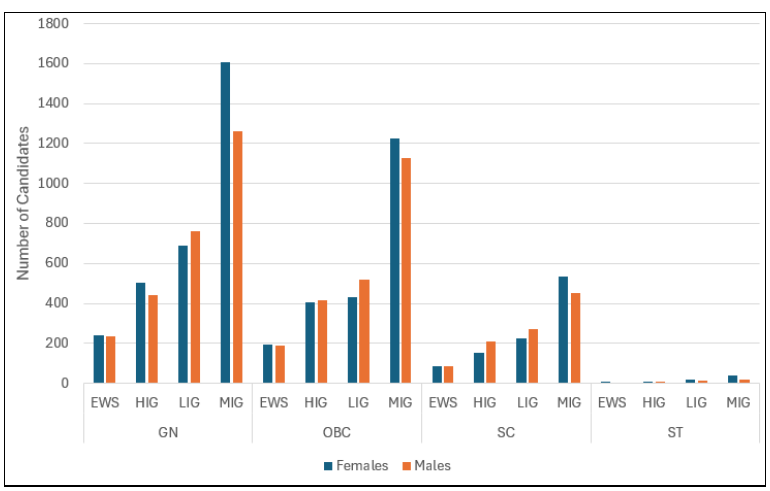

Figure 14: Number of applicants according to their family income class

The family income distribution of applicants admitted to DEI is shown in Figure 15. According to this, 7.63% of 1193 females and 9.89% of 1223 males from EWS class, and 21.4%

females and 33.9% males from LIG class qualified the exam and were admitted to the BSc pro//gram. Female applicants were more in MIG (51.38%) and HIG (19.6%) income classes.

An increasing trend was observed for females of the EWS class and both males and females of the LIG and MIG classes. The reason for this may be the affordable fee structure at DEI. However, a decreasing trend was observed in the HIG class (Figure 16).

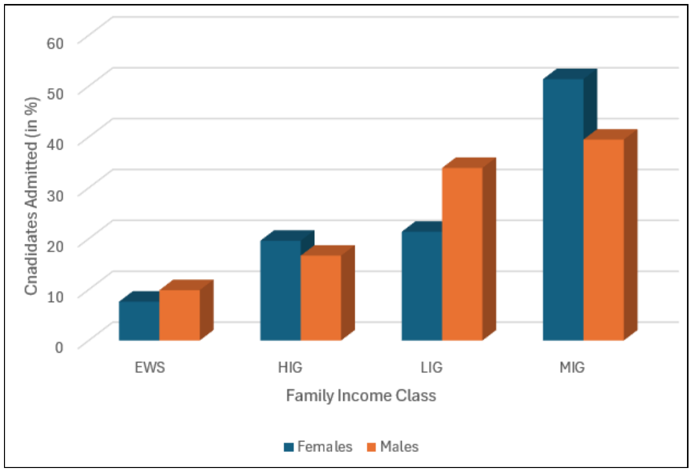

Figure 15: Family income class of applicants admitted

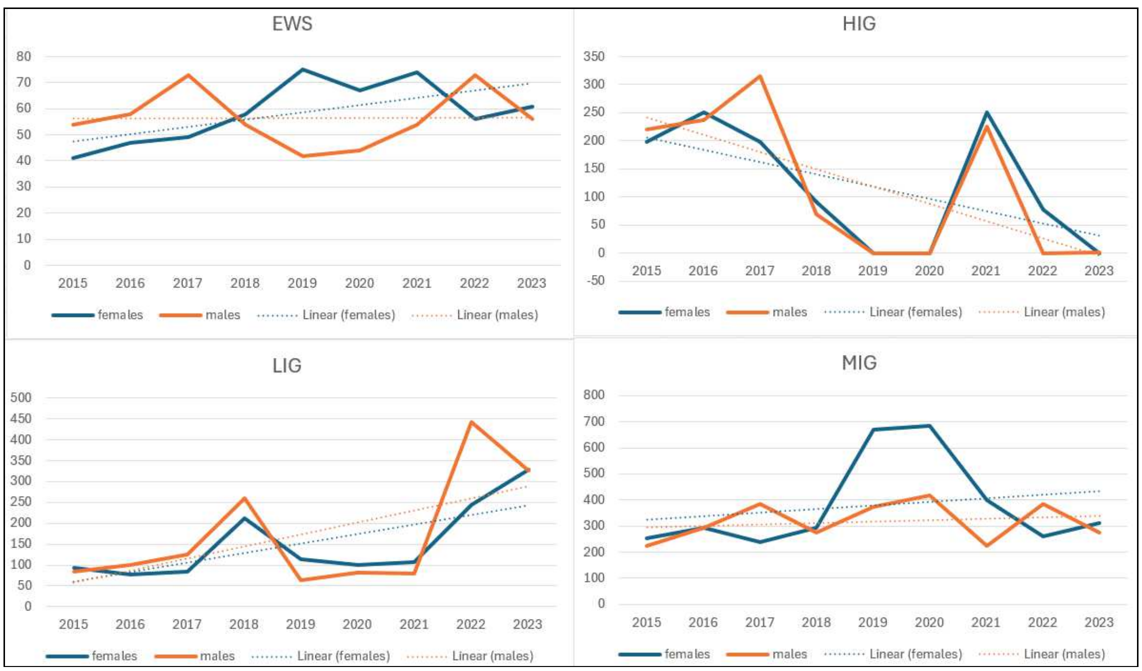

Figure 16: Family income class trends

Applicants from all over India apply in DEI as shown in Figure 17 and Figure 18. These are from Central Zone (Madhya Pradesh and Uttar Pradesh), Eastern Zone (Bihar, Jharkhand and West Bengal), North-Eastern States (Assam), North Zone (Delhi, Haryana, Punjab and Rajasthan), Southern Central (Telangana) and Southern Zone (Andhra Pradesh, Karnataka and Tamil Nadu). Most of the applicants are from Central Zone (90.77%).

Performing the drill-down operation on *AdmLocation* data cubes, it was observed that most of the applicants are from Uttar Pradesh (90.18%), 1.63% applicants are from Delhi, 1.75% are from Haryana and 28% are from Rajasthan. Only 3.64% of applicants are from other states. Drilling down further, it was seen that 23.86% of applicants were from Agra and 55.12% were from cities near Agra (this includes 16 cities within 80-100 km).

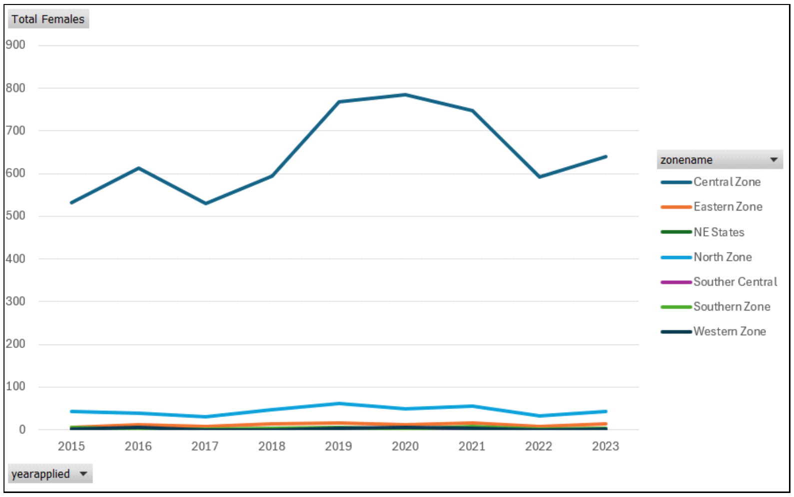

Figure 17. Location distribution of female applicants.

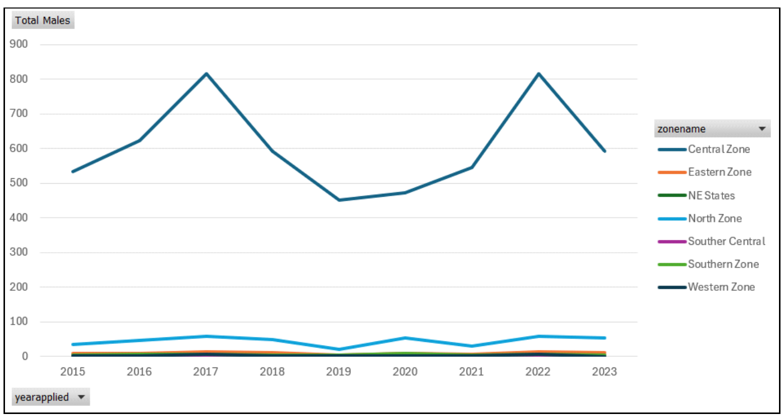

Figure 18. Location distribution of male applicants.

Out of 6362 female applicants and 5995 male applicants, 4139 female applicants and 3887 male applicants were called for the written test. The performance chart (Figure 19) shows that only 2 male applicants outperformed (>=90% marks) in the entrance exam. Out of 4139 females, 210 were good performs. Among these 8.1 % were from EWS class and 35.7% were from LIG class. Out of 3887 males, 264 were good performs. Among these 9.5% were from EWS class and 38.6% were from LIG class.Most of the applicants in all the years are average performers (marks secured between 60% to 74.9%). An increasing trend was observed in good and average performers (Figure 20). Thus, strategies may be designed to encourage the applicants in EWS and LIG family income groups.

Figure 21 displays the number of applicants in different courses by slicing the course data cube. An increasing trend was observed for the Physics-Mathematics group as well as the Computer Science-Mathematics group in male applicants. However, a decreasing trend was observed for BSc agriculture among the male applicants. All the other groups were not preferred by the males. For the female applicants, a decreasing trend was observed for the Physics-Mathematics group and Zoology-Chemistry group whereas an increasing trend was observed for BSc agriculture. All the other groups were not preferred by the females (Figures 22 and 23). The courses which were not preferred by the students may need a revision of the syllabus.

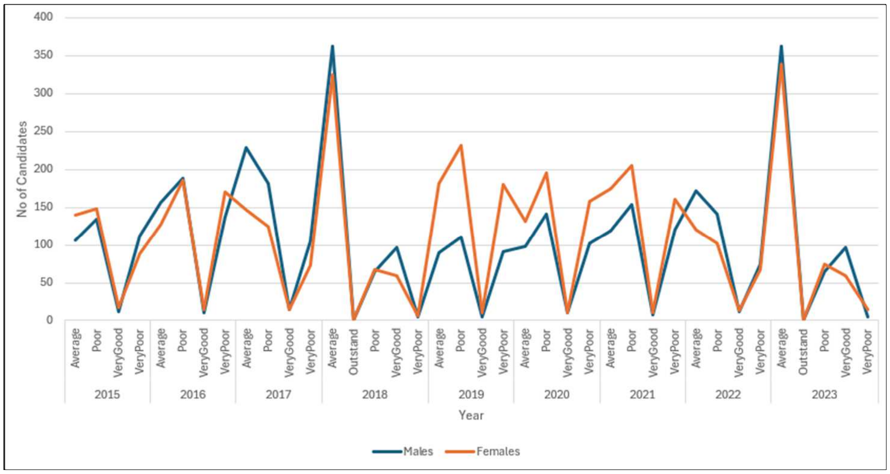

Figure 19. Performance of applicants

Figure 20. Performance trends.

7. Concluding Remarks

The focus of this paper is on the design and development of a data warehouse for the admission process to enable the multidimensional analysis of applicant data. Fact and Dimensions tables are arranged using the snowflake schema for the logical arrangement of the multidimensional database. The developed admission data warehouse is a data repository that integrates sparsely distributed admission data into a comprehensive analytical fashion for making effective decisions. This extensible data model allows for easy integration of new data types and a library of widgets for performing various analyses. The multidimensional trend analysis is presented using OLAP cubes. These data warehouse and data cubes are created in PostgreSQL.

The developed ADW can be useful for managing the admission process of any higher education institution by providing analyses of applicants from different views. It can also help in evaluating the quality of education and the performance of the institutions. The limitation of the developed data warehouse is that only relational DBMS is considered. However, in the future, an object-relational data warehouse can be developed.

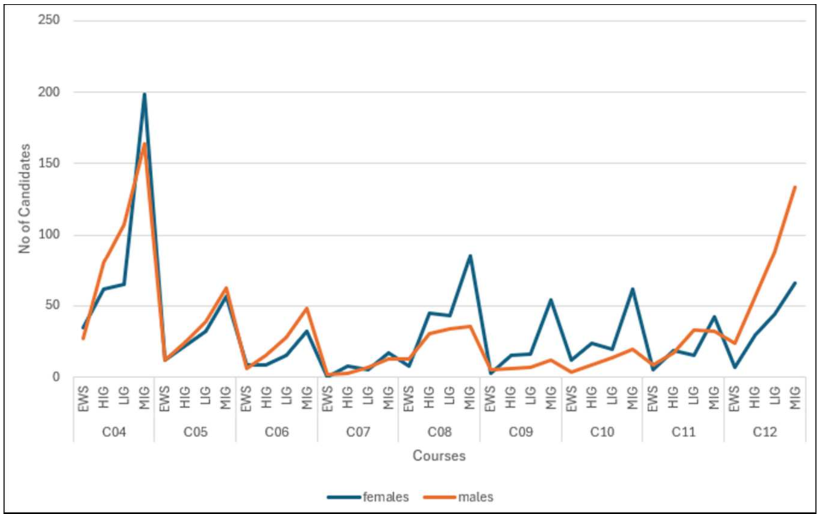

Figure 21. Applicants selecting different courses.

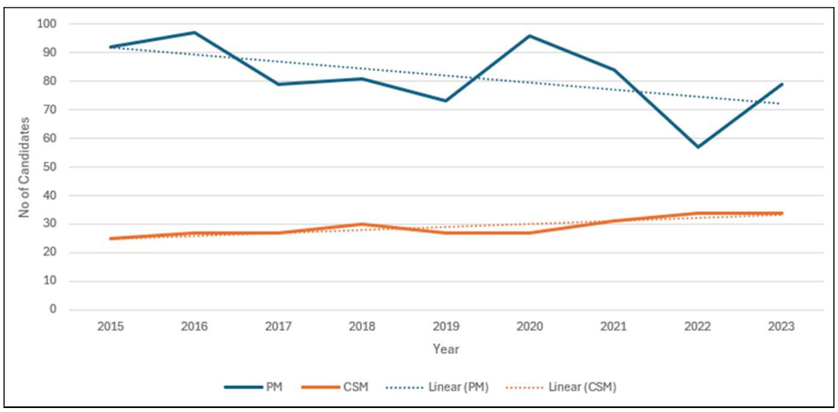

Figure 22. Preferences mathematics stream courses.

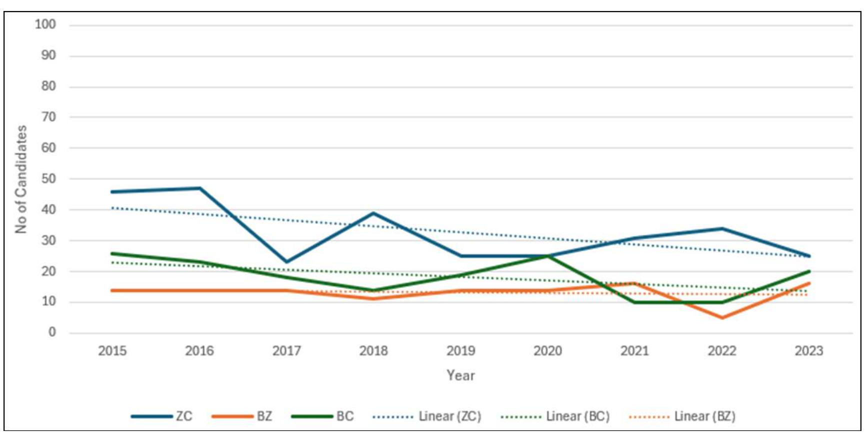

Figure 23. Preferences biology stream courses.

References

Bahaaudeen, A. (2023, March). The Impact of the Data Warehouse on Decision Making Quality and Speed in Higher Education. In *2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD)* (pp. 1-10). IEEE.

Chen, J., Zhan, J., & Tian, F. (2023, September). Research on the Construction of a Data Warehouse Model for College Student Performance. In *International Conference of Pioneering Computer Scientists, Engineers and Educators* (pp. 408-419). Singapore: Springer Nature Singapore.

Farhan, M. S., Youssef, A., & Abdelhamid, L. (2024). A Model for Enhancing Unstructured Big Data Warehouse Execution Time. *Big Data and Cognitive Computing*, *8*(2), 17.

Fauzi, C., Pribadi, D. S., & Alifi, M. R. (2023). Spatial data warehouse: an analysis in tourism sector of west java province. *International Journal of Basic and Applied Science*, *11*(4), 161-171.

Hamoud, A. K., Marwah, K. H., Alhilfi, Z., & Sabr, R. H. (2021). Implementing data-driven decision support system based on independent educational data mart. *International Journal of Electrical and Computer Engineering*, *11*(6), 5301.

Jo, K., Chang, D. J., Min, J. W., Yoo, Y. S., Lyu, B., Kwon, J. W., & Baek, J. (2022). Long-term prediction models for vision-threatening diabetic retinopathy using medical features from data warehouse. *Scientific Reports*, *12*(1), 8476.

Kaspar, M., Liman, L., Morbach, C., Dietrich, G., Seidlmayer, L. K., Puppe, F., & Störk, S. (2023). Querying a Clinical Data Warehouse for Combinations of Clinical and Imaging Data. *Journal of Digital Imaging*, *36*(2), 715-724.

Kimball, R., & Ross, M. (2011). *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons..

Liu, L. (2021, January). Visualized Analysis of Tourism Big Data based on Real-Time Analysis and Complexity Measurement. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)* (pp. 497-500). IEEE.

Park, T., & Kim, H. (2013). A data warehouse-based decision support system for sewer infrastructure management. *Automation in Construction*, *30*, 37-49.

Serasinghe, C. U., Jayakody, D. C. R., Dayananda, K. T. M. N., & Asanka, D. (2021, April). Design and Implementation of Data Warehouse for a Higher Educational Institute in Sri Lanka. In *2021 6th International Conference for Convergence in Technology (I2CT)* (pp. 1-5). IEEE.

Singh, P., & Dev, V. (2023). Data Warehouse with OLAP Technology for the Tourism Industry. In *Encyclopedia of Data Science and Machine Learning* (pp. 191-211). IGI Global.

Yu, X. (2021). The Application of Data Warehouse in Teaching Management in Colleges and Universities. In *Journal of Physics: Conference Series* (Vol. 1738, No. 1, p. 012090). IOP Publishing.

Zhang, S., & You, L. (2021, September). Research on Higher Education Intelligent Decision System Based on Data Mining. In *2021 4th International Conference on Information Systems and Computer Aided Education* (pp. 2863-2871).