**CSE 4/560 Data Models and Query Language**

**Project Report**

**Title: Provision for Analysis and Tracking of Educational data by Federal Authority**

**By N S Sai Teja Jampani(njampani@buffalo.edu)**

**& Anjan Krishna K(anjankri@buffalo.edu)**

1. **PROBLEM STATEMENT**

The Government provides various initiatives to the students in the country. On top of that, the federal authority also needs to regularize and monitor the working of various educational institutions. Secular norms like ethnic representation, Gender, etc. should be tracked and ensured for legitimacy. However, all this data will need to be converted to database system from mere forms for the government to get deeper insights and consider future decision making. For the very same purpose, we will be designing a database that will have university details, student details including aid given, degree completion time, ethnic background, etc., faculty details and other supporting information.

**Why Database over Excel?**

A structured Database will provide robust and secure storage for large volumes of data. It removes redundancies and implicit column dependencies. It will enhance visualization capabilities and remove any unwanted columns. It will also provide access to all the employees under Authorization to access and take decisions within their scope. With aforesaid reasons it is convincing to choose a database with normalized schema over traditional excel to our problem.

**User and Administrator**

This database that we design will be used by the state educational officers to get insights and make decisions. The administrator will be the federal Educational Department which will collect and maintain the database.

1. **Database Design:**

The parent Dataset has been obtained from Data World Website which contains the institution list and other details. It is available in the following link.

<https://data.world/databeats/college-completion/workspace/file?filename=cc_institution_details.csv>

This has been used primarily to design the database schema. The updated version of the schema after normalizing to BCNF is given below.

**Chart

Description automatically generated**

We can see that there is no transitive dependency of non-prime attributes in any of the tables and also for every functional dependency X - > Y, X is the super key of the table for each table. Thus, we can conclude that our schema is normalized to 3NF and BCNF.

The table details for each table are as follows.

**University**

*University\_id* (Integer): Represents the University ID to uniquely determine each University. It is a primary key and not null.

*Name* (Text): Represents the name of the University and is not null.

*Control* (Text): Specifies the control of the institution if it is private, public, etc. and is not null.

*Website* (Text): Website for the institution and is not null.

*Zipcode* (Integer): Zip code of the university, foreign key referencing table Address\_zip and not null.

**Address\_zip**

*Zipcode* (Integer): Zip code number to uniquely determine each zip code, primary key and not null.

*City* (Text): City to which the zip code belongs and is not null.

*State* (Text): State to which the zip code belongs and is not null.

**Ethnicity**

*Ethnic\_id* (Integer): Ethnicity number to uniquely determine each ethnicity, primary key and not null.

*Ethnicity* (Text): Ethnicity corresponding to ethnic number and not null.

**StudentDetails**

*Student\_id* (Integer): Student ID to uniquely determine each student. It is a primary key and not null.

*Student\_Name* (Text): Name of the student and it is not null.

*Gender* (Text): Gender of the student and not null.

*Sat\_Score* (Integer): Sat score of the student and not null.

*Aid\_Value* (Integer): Aid given by the Government to the student and not null.

*Degree\_Completion*

*\_time* (Integer): Time taken by the student to complete the degree and not null.

*Ethnic\_id* (Integer): Ethnic ID of the student and foreign key referencing table Ethnicity. It is not null.

*University\_id* (Integer): University to which the student belongs, foreign key to table University and not null.

**FacultyDetails**

*Faculty\_id* (Integer): Faculty number which uniquely determines the faculty, part of composite primary key and not null.

*University\_id* (Integer): University to which the faculty belongs, part of composite primary key and not null.

*Name* (Text): Name of the faculty and not null.

*Gender* (Text): Gender of the faculty and not null.

*Ethnic\_id* (Integer): Ethnic ID of the faculty, foreign key referencing table Ethnicity and not null.

University table data has been obtained from parent dataset, data for Address\_zip has been obtained from zip code dataset available on internet along with python simulation to generate city and state address for each zip code. Data for Ethnicity has been obtained through general inference of various ethnicities in the country and for StudentDetails and FacultyDetails, a python simulation has been used using faker library. The python files will be attached in the submission zip folder.

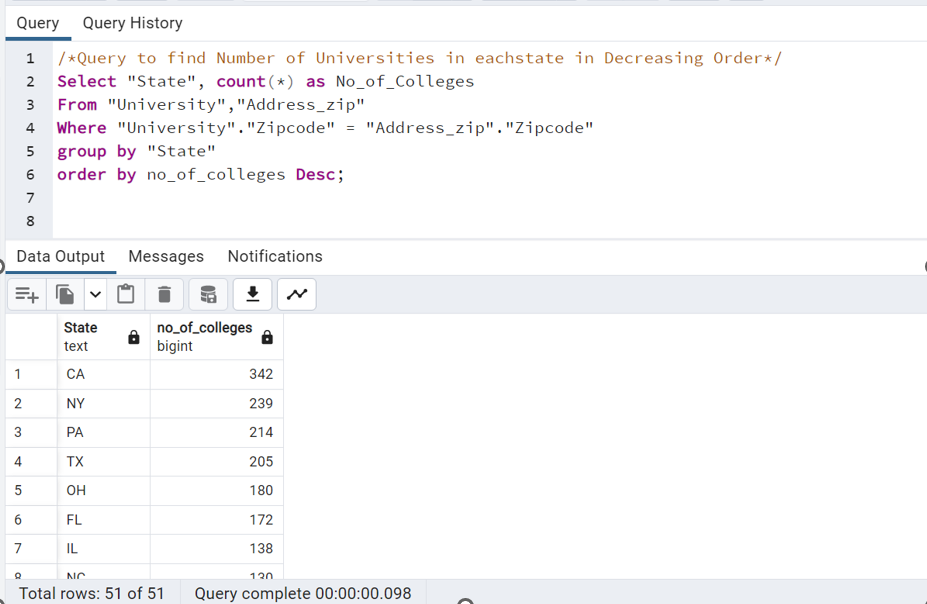
1. **Queries**

The following queries have been implemented after creating the database. We have described the scenario for which each query is used.

**General Queries Joins, Group by, order by and Subqueries**.

**Requirement 1**: Government wish to know the number of colleges in each state to allocate budget.

We write a query to find Number of Universities in each state in Decreasing Order. Query and output is attached below.



**Requirement 2:** Government wish to keep a check on the number of Public Universities

in the country and wants to know the present number of public Universities in the country.

The Query and result is attached below.

Graphical user interface, text, application, email

Description automatically generated

**Requirement 3:** Government wish to know the total aid going to each ethnic group to see if there is bias towards any specific ethnic group**.**

Query and result is attached below.

Graphical user interface, application

Description automatically generated

**Requirement 4:** Government wants to know the Ethnic representation of faculty for each University to see if guidelines are followed.

Query and result is attached below.

**Graphical user interface, text

Description automatically generated with medium confidence**

**Requirement 5:** Government need to know the number of colleges present in Buffalo

to make it an educational corridor if the number of colleges is greater than 5.

The query and result is attached below.

**Graphical user interface, text, application

Description automatically generated**

**Requirement 6:** Government wants to know the male to female ratio of faculty for

Amridge University, University ID = 3 after complaints of poor gender ratio.

Graphical user interface, text, application

Description automatically generated

**Queries for Query Analysis and usage of Explain Tool.**

**Requirement 7:** Government wishes to know the student details who score above average in sat to analyze the trend in their degree completion.

We query Students who have sat score greater than 1500 and less than 1600. We shall check the query time before indexing.

**Table

Description automatically generated**

Explain tool Gives us the following time taken and other details with respect to the execution of the query.

Diagram, application

Description automatically generated with medium confidence **Graphical user interface, application

Description automatically generated**

Now, an index has been created for the Sat\_Score column of StudentDetails table and the explain tool details are as follows.

**Graphical user interface, text, application

Description automatically generated** Graphical user interface

Description automatically generated with low confidence

**Chart, treemap chart

Description automatically generated**

We can observe that after indexing, the query time has been improved drastically.

**Requirement 8:** Government needs to send inputs to universities doing poorly in terms of degree completion time and therefore wants to find the Universities that have average degree completion time greater mean degree completion time of all the students.

The Query and results are given below**.**

**Graphical user interface, text, application

Description automatically generated**

We have observed that the query is not optimized as seen from the explain tool attached below.

**Diagram

Description automatically generated**

**Table

Description automatically generated**

We now optimize the query by introducing index on Degree\_Completion\_time column and the results obtained are shown below.

Graphical user interface, text

Description automatically generated

**Diagram, box and whisker chart

Description automatically generated with medium confidence**

**Table

Description automatically generated**

We can see a drastic reduction in query time after indexing.

**Requirement 9:** Government wish to analyze the students who receive aid greater than 18000$ and less than 20000$ and degree completion is greater than 4 years but less than 6 years to see if the aid is really helping them.

The query and result is given below.

A picture containing table

Description automatically generated

We now use the Explain tool to find if the results can be optimized. The details are attached below.

**Application

Description automatically generated with medium confidence**

**Table

Description automatically generated with medium confidence**

We run the query after creating index for Aid\_Value and Degree\_Completion\_time and use the explain tool to check for updated query analysis. The results are as follows.

**A picture containing box and whisker chart

Description automatically generated**

**Treemap chart

Description automatically generated with medium confidence**

Thus, We can see a drastic change in query Execution time after Indexing.

**Queries for Insert, Delete and Update**

**Requirement 10:** New college is established at Englewood. The government needs the college to be inserted in the University Table.

Graphical user interface, text, application, email

Description automatically generated

**Graphical user interface, application

Description automatically generated**

**Requirement 11:** Faculty Cindy Smith has retired, Government wants to remove her entry from the FacultyDetails table.

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Requirement 12:** Jenifer Carr, Student ID =1 received a federal aid of 12000$. The department wants an update to her details.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application, table

Description automatically generated

**Requirement 13:** New University SUNY Englewood website has crashed due to a lot of traffic while receiving applications. A new website has been created and the department needs an update to the database.



**Graphical user interface, text, application, email

Description automatically generated**

**Table

Description automatically generated**

1. **Bonus Task- Visualize Query on a Website:**

We have enabled the localhost server/website to display all the entries from the University table. A provision is also given to the administrator with authorized access to add or delete entries in University and StudentDetails tables.

We have used python-based web development framework Django for doing this part.

We have first integrated our postgres database with Django backend and visualization of the query on the front end is done via HTML. The screenshots for the same are attached below.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, website

Description automatically generated

Graphical user interface, application, email

Description automatically generated

1. **REFERENCES**
2. <https://data.world/databeats/college-completion/workspace/file?filename=cc_institution_details.csv>
3. <https://beginnersbook.com/2015/05/normalization-in-dbms/>
4. <https://www.w3schools.com/>
5. <https://djangoadventures.com/>