BERKELEY INSTITUTE
FOR DATA SCIENCE

Education and Training Database Project (Team 5)

Contributors:

Avie Sanchez, Andrew Ding, Christian Choi, Christopher Liu, Pui Fung Lam, Xiaohan Wang, and Yunshan Guo

I. Executive Summary

Client Description

Client: BIDS (Berkeley Institutes of Data Science)

Contact person: Anthony Suen (Moore Sloan Data Science Fellow) ,Vinitra (Researcher)

Emails: anthonysuen@berkeley.edu vinitra@berkeley.edu

URL: https://bids.berkeley.edu

Client Information:
BIDS is a central hub of research and education at UC Berkeley designed to facilitate and nurture data-intensive science. BIDS actively seeks and engages communities with different academic fields ranging from life, social and physical sciences in data science research. Some current projects BIDS is planning includes: Data Sciences for the 21st Century, BIDS Collaborative, and Data Science courses planning at UC-Berkeley.

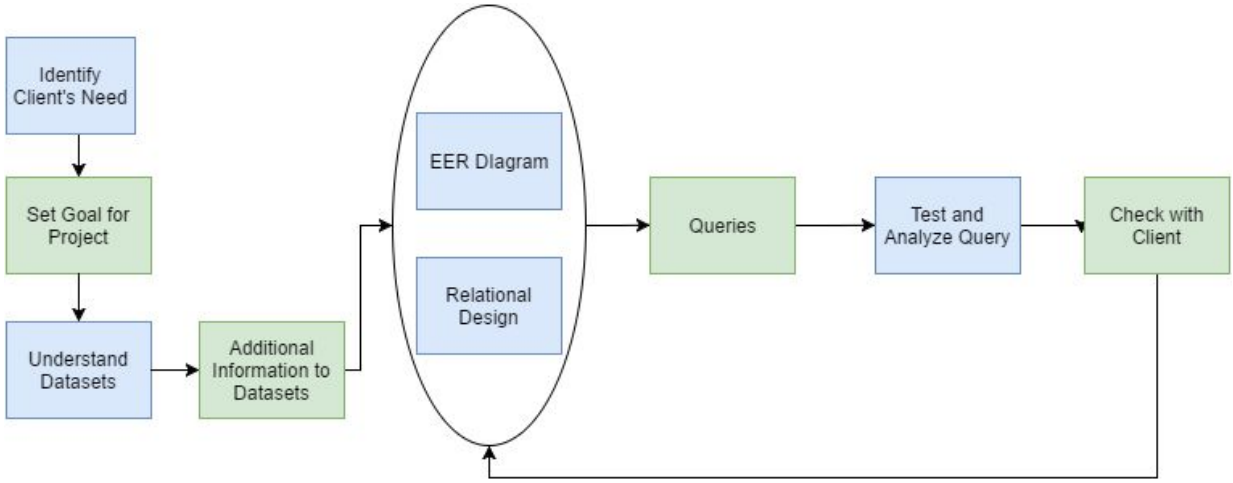
There are more than 30 projects going on at BIDS currently. While each of these projects has a different focus, we reached out to Anthony Suen, a project program analyst at BIDS who is in charge of Data Science Initiative Project. Anthony also co-founded the BIDS Collaborative, which helps both undergraduate and graduate students collaborate with nonprofits, government, and research institutes to solve challenging real-world issues. Therefore, the data sets he provided focus more on students, faculty members, research projects, and other related resources for people at Berkeley. Anthony is also working with some student groups to design new data science courses at UC-Berkeley.

Goals and Project Approach

When we were first given this project, we started with the intention of creating a database to solve our clients main need; however, as we moved further into the semester and gained more skills, we designed a database that exceeded the expectations of what our client needed to a fully integrated system that they could potentially use for all BIDS related purposes.

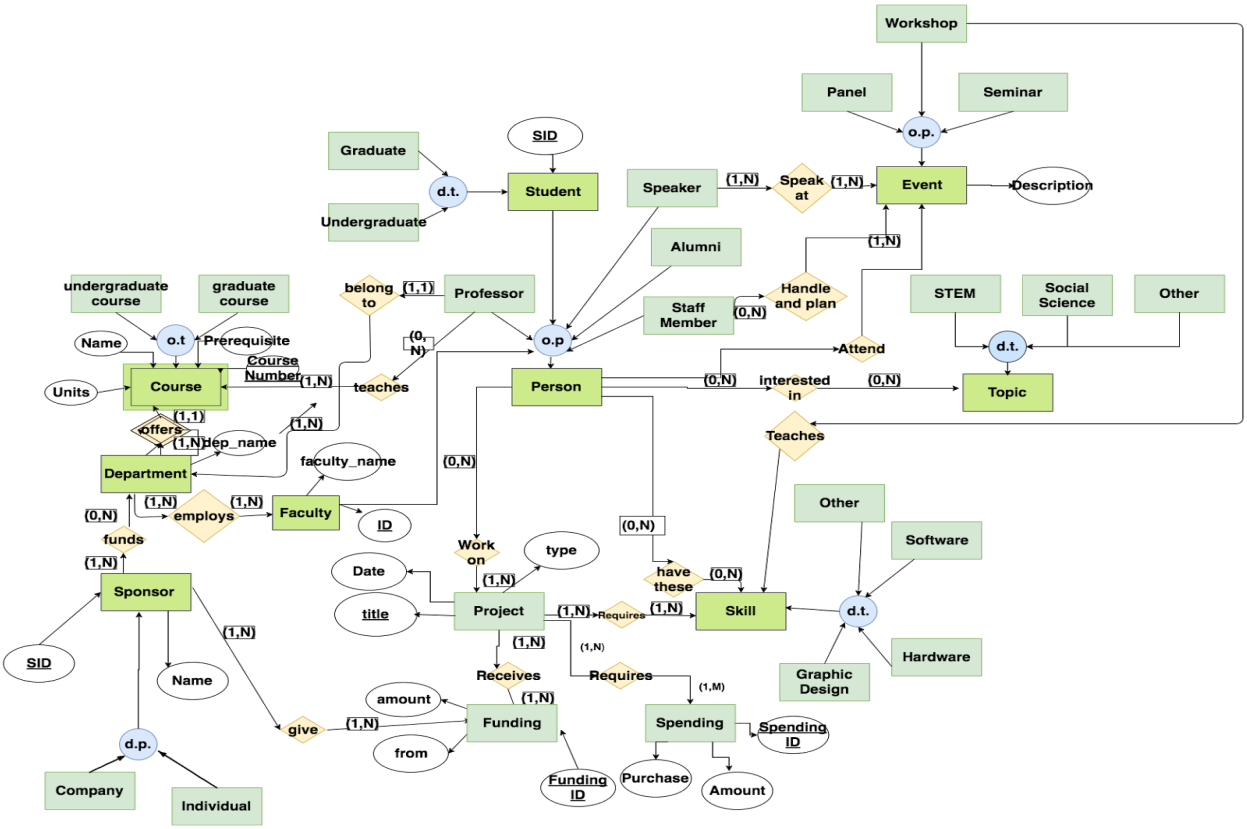
The problem is that our clients at BIDS have the data stored up in multiple .csv files and they have an issue with mapping or drawing connections from these files. They want us to design and to build a relational database that would make it easier/faster to reconcile data, so with the database, they hope to run SQL queries in order to perform data analysis. The datasets provided by BIDS include: Undergraduate courses, Graduate courses, Research grant history, and faculty research interests. Some of the questions they are interested in involve which department gets more funding, what are the differences in percentage of funding among

different departments, and which departments show interests in Data Science.



With the initial problem in mind, we we set a goal to create a database based on the four datasets we had. Yet, as we gained feedback and moved forward with the project, we expanded our project to a more expansive database.We expanded on our datasets and added additional information to the datasets by creating more entities than what we had. Therefore, our focus in the project changed into designing a database for BIDS that not only contains all of the data we have, but also other entities that may be useful for BIDS in future. Our EER diagram started with very little entities, and by the end of the semester, it became more of a complicated web of connected entities representative of a fully working database.

The EER design is centered around the superclass “person” as BIDS work with a whole variety of people assisting in its initiative. The design is people-centric and that entail subclasses with student, professor, speaker, staff member, faculty and etc with a unique relationship showing how they interact/contribute to BIDS. For example, sponsor would fund a specific department as that department employs faculty. These relationships would create a network of interactions that would form the framework for our queries. Below is a final EER diagram depicting all the relations and entities:



(Final EER Diagram)

1. Person(ID, Lname, Fname, Birth_Date, university, email_address, phone_number)
   - A. Student(ID[1], ...)
     - a. Graduate(ID[1], ...)
     - b. Undergraduate(ID[1], ...)
   - B. Speaker(ID[1], Event_Type, ...)
   - C. Alumni(ID[1], dep_name, Employer ...)
   - D. Staff_Member(ID[1], Event, dep_name, ...)
   - E. Professor(ID[1], dep_name, ...)
   - F. Faculty(ID[1], Faculty_name, dep_name[2])
2. Department(dep_name, dep_email, dep_address)
3. Course(Course_Num, dep_name[2], Name, Prerequisite, Units, ID[1E])
   - A. Undergraduate_Course(Course_Num[3], ...)
   - B. Graduate_Course(Course_Num[3], ...)
4. Sponsor(Sponsor_ID, dep_Name[2])
   - A. Company(Sponsor_ID[4], name, ...)
   - B. Individual(Sponsor_ID[4], Lname, Fname, ...)
5. Funding(Fund_ID, Amount, Date, Sponsor_ID[4], P_ID[6])
6. Spending(Spending_ID, ID[1], P_ID[7] Amount, Purchase)
7. Project(P_ID,Title, Date, Topic, PID[1])
8. Skill(S_Name)
   - A. Hardware(S_Name[7])
   - B. Software(S_Name[7])
   - C. Design(S_Name[7])
   - D. Other(S_Name[7])

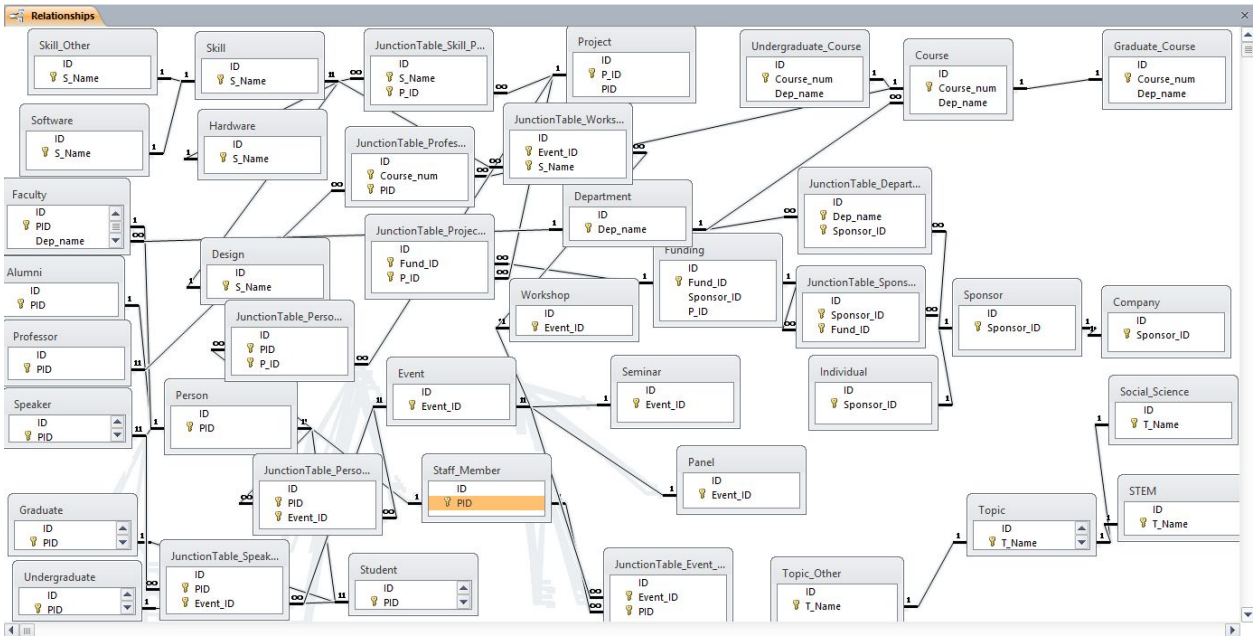
9. Event(Event_ID, Description, Type)
   A. Panel(Event_ID[9], ...)
   B. Workshop(Event_ID[9], ...)
   C. Seminar(Event_ID[9], ...)
10. Topic(T_Name)
   A. STEM(T_Name[9], ...)
   B. Social_Science(T_Name[9], ...)
   C. Other(T_Name[9], ...)

1. ProjectSkill(P_ID[7], S_Name[8], Level of Proficiency, Necessity)
2. TopicofInterest (Person_ID[1], t_Name[9], Reason, Intensity)
3. SkillPerson(Person_ID[1], S_Name[8], Method Acquired)
4. WorkshopSkill(EventID[8B], S_Name[7], Level of Proficiency)
5. AttendEvent(Person_ID[1], EventID[9], Date, Location, Satisfaction)
6. EventStaff(SID[10], EventID[9], Salary, Satisfaction)
7. ProjectPerson(P_ID[7], Person_ID[1], Qualification)
8. ProfessorCourse(P_ID[7], Course_num[3], Satisfaction, Rating)
9. ProjectSpending(Spending_ID[6], P_ID[7])
10. SpeakerEvent(Person_ID[1], EventID[9],Date, Location)

(Final Relational Design, schema)

Another factor we had to consider while creating the EER diagrams and Relational Design were the tables and attributes needed for our queries. Many of the Relational schemas were lacking in attributes, so in order to accommodate the queries we came up with, we added extra attributes and entities when necessary.

After revising the EER diagram and Relational Design many times, we started designing the database on MS Access. Getting the connections to work required a lot of meticulous and precise oversight over each table and connection. There were many points where creating the actual database made us stop to look at the accuracy of the diagram and schema because there were many spots where we missed junction tables for M:N relationships, and other such problems.



(Screenshot of Relationship View)

Conclusively, with a lot of detailed revision, we created a database that satisfactorily met our client's initial needs and went further expanded into a database for all of BIDS purposes.

II. Query Analysis

Queries

Query 1: Forecasting Skill Demand (Events Forecasting)

Description:
Over the last few years, certain skills have become more highly desirable while some skills have become obsolete. For example, increasingly, more and more people are interested in machine learning and coding skills versus almost 10 years ago. For our client, we decided to create a query to forecast what skills will be more desirable in order to accommodate and plan what classes or events they should offer in an upcoming semester/year. In order to create this query, we are going to use SQL to arrange the table we want to analyze; after aggregating the data, we are going to use R to create a time series prediction of where the projection of demand of skills will go. Depending on the data, we can model the data using ARIMA models.

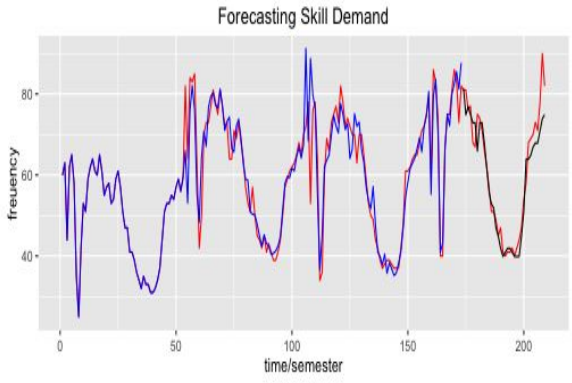
Benefits:
Since there are many varied events hold by BIDS all the time, forecasting the skill demand will provide BIDS with some insights on the general interest of the audience. Based on the analysis result, BIDS can better plan their future events meeting the expectation of the audience, thus enhancing the attendance. As BIDS is also working on designing some new courses of data science at UC-Berkeley, the skill demand analysis will give them insights in potential new classes that students will be interested in.

SQL:

```
(SELECT p.P_ID, p.MONTH(date) as
month, p.YEAR(date) as year,
ps.S_name, count(ps.S_name)
FROM Project p, ProjectSkill as ps
WHERE p.P_ID = ps.P_ID
GROUP BY  p.P_ID, month, year
ps.S_name) a

(SELECT e.Event_ID, e.Description,
w.S_Name, count(w.S_Name)
FROM Event e, WorkshopSkill w
WHERE e.Event_ID = w.EventID
GROUP BY e.Event_ID, e.Description,
w.S_Name )b

SELECT a.P_ID, a.S_Name,
a.count(ps.S_name),
b.count(w.S_name)
FROM a, b
WHERE a.S_name = w.S_name
GROUP BY  a.P_ID, a.S_Name
```



Analysis:
By using SQL to retrieve the data of total demand of each skill regarding each month, we have a clear timeline as well as the corresponding demand. Therefore, we model the data using ARIMA model and forecast the future demand with autoregression. The graph above is the ARIMA plot in R. In the graph, the blue line is the original data, while the red line is the fitting line by using ARIMA model. The black line, however, is our desired line which is the future forecast based on the model we've trained. From the graph above, we can see that the model fits well with the original data. Therefore the black forecasting line should be reliable for us.

Query 2: Sponsors and Donors

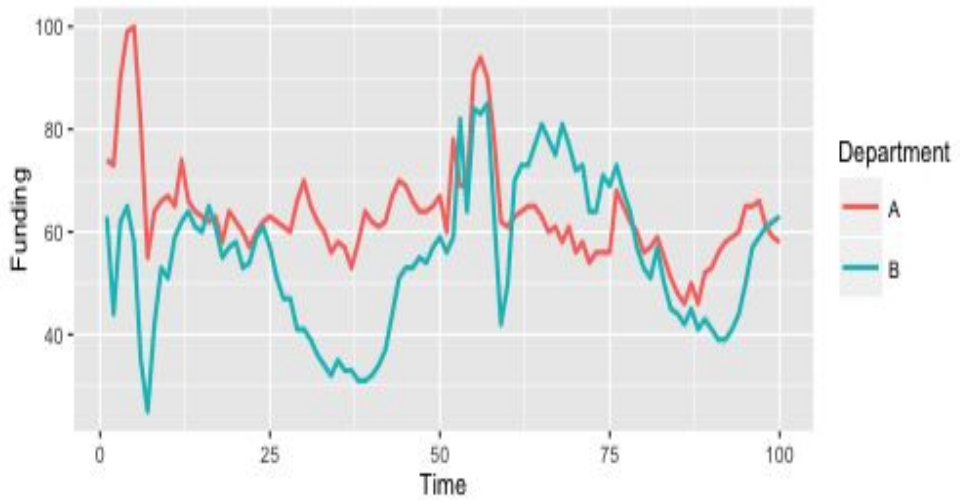
Description:
There are many alumni/ companies recorded in our database that have not donated to Berkeley yet. We wanted to create a query that looks into potential individuals or companies in our current database for fundings. To narrow it down to a list of potential donors, we are going to search the list of alumni and see which alumni who haven't donated. Also, to get more companies to donate, we also decided to see where their employers are working. Additionally, we wanted to see which department needs more funding so we can ask/look for more companies who would most likely donate to that department. Depending on the industry type, we wanted to create an additional query to determine which departments these donors should give their money.

Benefits:
Instead of sending emails to all people and encourage them to donate money, we can target specific one by having a list of alumni,who would potentially interested and become our sponsors, introducing more details about current projects, will help us increase the chance to get more fundings.
Virtualizing data of fundings received by each department help to observe general trends and present a clear comparison of fundings between different departments. If fundings from some department shrinks and might need more money, combing with potential donor list, we can figure out an effective way to obtain money and plan for future.

Analysis:
In SQL, we first inner join tables of Alumni and donors by name to find those who are both alumni and donor. Second step is to go back to Alumni list and subtract the names we got from last step, and then the remaining part are potential donor list created from Alumni table.

Alumni_ID	Lname	Fname
11	Colford	John
12	Needell	Barbara
20	Alber	Thomas
14	Carmichael	Christopher
13	Midgley	James

Given the data tables of fundings, we reload it through R and convert to dataframe for future analysis. We aggregate data to make it into more tight format. Using built in package, we plot the fundings received by different departments versus time and various color represents each department. Here, we also can treat these data as a time series, by fitting some reasonable model, predictions could be made based on that, which helps us planning ahead. Andwe also notice that popular majors/departments generally receive more money so that they have enough resource to development better. However, some small developments might have less research opportunities due to limited fundings they have. Thus, next step we suggest is to enough donors to pay more attention on less popular departments.



SQL:

```
(SELECT Person_ID
FROM Alumni
WHERE Person_ID NOT IN
    (SELECT a.Person_ID
       FROM Alumni as a,
       Individual as i
       WHERE a.Person_ID =
i.Person_ID)) a

SELECT a.Person_ID, c.name
FROM a, Individual as i, Company as
c
WHERE a.Person_ID = i.Person_ID
AND c.Sponsor_ID = i.Sponsor_ID

SELECT f.Sponsor_ID, f.Fund_ID,
f.amount, p.dep_name
FROM Funding as f, Project as p
WHERE f.P_ID = p.P_ID
```

Relational Algebra:

(For finding companies)

R1 <- π_{name}(Company)

R2<- $\pi_{Employer}$(Alumni)

R<- R_2-R_1

*Department funding - Analysis (in R) of the fundings each department receives and how they change throughout the years

Query 3: Team Assignment (Matching Problem)

Description:
Whenever working on a research project, it is always good to know what is the optimal team and who are the best people for the job. This is a problem we thought would come up when BIDS fellows are deciding to start a new project. In order to do this, we brainstormed that we need to determine the best team size, by looking at all the projects and see how many people were involved, number of fellows and project assistants, and compare it to the length of time it took to finish the project. Accounting that some projects will take longer than others due to the necessity of time in the project, we decided to separate each project by department/type of project such as Biology, Social Science, etc.
Also, in order to optimize who would be the best people to get involved in the project, we decided to turn this into a matching problem. We decided we needed to pair the necessary skills needed for a project to the most qualified person who has the right skills.

Benefits:

There are a lot of benefits to optimizing the matching involved with team members and projects. Research has shown that having too many members on a team can actually hinder productivity, so one does not want too many members working on the same project. On the other hand, another goal is to make sure as many projects as possible can be successfully completed, so projects shouldn't be understaffed either.
In addition to the quantities, more projects will be complete successfully if team members with the right skill sets are paired with the right projects, as this will lead to people being able to use their skills correctly and optimally. Of course there are a lot of qualitative factors involved in picking the right team members as well, but this provides a base guideline.

Analysis:

SQL:

```
(DECLARE @rank int;
SET @rank = (SELECT COUNT(p.t_name) from p)
SELECT p.t_name, @rank
FROM project p
GROUP BY p.t_name) a

SELECT p.P_ID, person.P_ID, a.t_name, a.@rank
FROM Project as p, Person as person, Topic as t, a
WHERE p.Person_ID = person.Person_ID
AND t.T_Name = p.Topic
AND p.Topic = a.t_name
HAVING t.T_Name in
    (SELECT p.Topic
      FROM Project as p)
```

```
GROUP BY p.P_ID
```

To find the top ‘n’ skills:

```
SELECT top n @rank FROM a;
```

We made an LP where all projects require skills, all skills are weighted based off the rank and we want to maximize the overall productivity.

$X_{k,j}$ = if student k works on project j (binary)
$Y_{i,j}$ = if student k has skill i (binary)
$W_{i,j}$ = weight of skill i on project j.
C_k = Student project capacity.
Max $\Sigma_{Xk,j}$, $\forall Y_{i,j} \in Y$ $W_{i,j} * Y_{i,k} * X_{k,j}$
s.t.
$X_{k,j} \in \{0, 1\}$, $\forall X_{k,j} \in X$ Binary
Constraint
$Y_{i,j} \in \{0, 1\}$, $\forall$ $Y_{i,j\,j} \in Y$ Binary
Constraint
$\Sigma_j X_{k,j} \leq C_k$ Capacity Constraint
$\Sigma_k X_{k,j*} W_{i,j*} Y_{i,j}$ $\forall$ $W_{i,j} \in W \geq W_{i,j}$ $\forall$ $i \in I$ Fulfillment Constraint

AMPL Code:

```
param n;
param m;
param l;
param b{i in 1..m};
param s{i in 1..j};

var c{k in 1..l}; #capacity
var x{k in 1..l, j in 1..m} binary; #student k working on project j
var y{i in 1..n, k in 1..l} binary; #student vs. skill
var w{i in 1..n, j in 1..m}; #weights

maximize productivity: sum{k in 1..l, j in 1..m} (sum {y in y[i,k]}) w[i,j]*y[i,k]*x[k,j];

subject to capacity {j in 1..m}: x{k in 1..l, j in 1..m} <= c{k in 1..l};
subject to fulfillmentone {k in 1..l, j in 1..m}:
sum{k in 1..l, j in 1..m} (sum {y in y[i,k]}) w[i,j]*y[i,k]*x[k,j] >= w[i,j];
subject to fulfillmenttwo {i in 1..n, k in 1..l}:
sum{k in 1..l, j in 1..m} (sum {y in y[i,k]}) w[i,j]*y[i,k]*x[k,j] >= w[i,j];
```

Sample Result for two students:

```
x[1,1] = 0
x[1,2] = 1
x[1,3] = 0
x[1,4] = 0
x[1,5] = 0
x[1,6] = 0
x[1,7] = 1
x[1,8] = 0
x[1,9] = 0
x[1,10] = 0
x[1,11] = 0
x[2,1] = 0
x[2,2] = 0
x[2,3] = 0
x[2,4] = 0
x[2,5] = 1
x[2,6] = 0
x[2,7] = 0
x[2,8] = 0
x[2,9] = 0
x[2,10] = 0
x[2,11] = 0
```

Query 4: Professor Spending Log

Description:
One of the things that our clients has asked us to do is find a way to keep track of how the BIDS fellows are spending the money they are given from sponsors for their projects. This would include spending on equipment, project assistants and etc. After keeping a log, he wants us to analyze and observe what professors were spending on most or least. What is the average spending? And how much of the spending is left after the project ends? Based on this need, we have created an entity called “Spending”, which contains the spending information for each professor. While the original requirement is to create a log, we have seen the potential benefits in this spending log with more in-depth analysis on it. Therefore, we designed this query to retrieve the data and then analyse it using R.

Benefits:
By calculating the percentage of funding usage for each professor and the balance of their projects, we can provide BIDS with insights on how efficient professors spend their money and what type of projects have a higher demand of funding. This analysis is very useful for professors who are planning for a new project and want to know how much funding they may need to apply for. By updating the spending log regularly, BIDS may be able to provide professors with some updated information such as when they will run out of their funding, or

when their balance will be negative. In this way, professor can automate and systematically calculate budget.

```
SQL:
SELECT p.P_ID, p.Person_ID,
s.Spending_ID, SUM(s.Amount) as
Total, AVG(s.Amount) as AVG,
s.Purchase, (f.Amount - s.Amount) as
Difference
FROM Project as p, Spending as s,
Funding as f
WHERE p.P_ID = f.P_Id
AND p.P_ID = s.P_ID
GROUP BY
p.P_ID, p.Person_ID, s.Spending_ID,
s.Purchase;
```

Relational Algebra to calculate balance of Project 2:

R_1<- $\pi_{Amount}(\Theta_{P\_ID=2}(Funding))$

R_2<- $\pi_{Amount}(\Theta_{P\_ID=2}(Spending))$

R<-R_1-R_2

Analysis:

Percentage Of Money Spent Over Fund For Each Professor:

	PID	eff
1	000018-008	6.769892e+00
2	000266-013	-1.197704e+00
3	000268-018	1.390835e+03
4	000294-018	1.645693e+02
5	000316-012	-1.387696e+00
6	000317-011	1.436432e+02
7	000350-010	-1.000000e+00
8	000350-016	1.409550e+02
9	000369-012	-3.097794e+00
10	000392-014	-1.000000e+00
11	000419-016	-2.047598e+00
12	000420-015	-1.295179e+01
13	000425-017	-1.512302e+00
14	000439-019	1.633400e+02
15	000454-011	-1.000000e+00
16	000454-016	1.085291e+02
17	000463-017	-1.701585e+00
18	000506-015	-3.475109e+00
19	000615-010	-1.000000e+00
20	000622-014	-6.173756e+00
21	000624-013	-1.000000e+00
22	000651-011	-1.000000e+00
23	000651-017	6.858082e+01
24	000660-015	-6.827609e+00
25	000661-012	-1.000000e+00
26	000661-018	1.913233e+02
27	000663-013	-7.118480e+00
28	000678-012	-4.865813e+01
29	002121-002	-1.000000e+00
30	003561-002	-1.000000e+00
31	007143-002	-1.901195e-01
32	009466-002	-1.000000e+00
33	009489-008	-1.242006e+00
34	009685-010	-1.726475e+00
35	010168-007	-1.515874e+00
36	011429-002	7.439891e+01
37	011551-010	-3.561208e+00

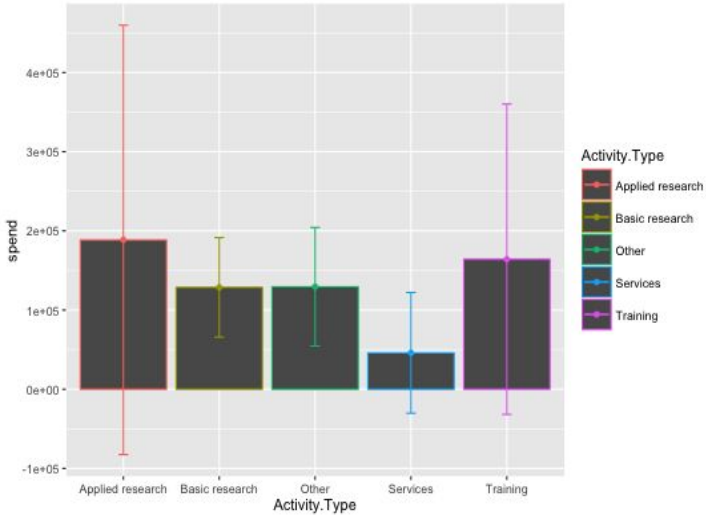
The output above shows the percentage of money that professors have spent out of their funding. Notice that there are some negative numbers. That means the professor has spent more that their funding. In that case, BIDS can send emails to them asking if they need additional information of some extra fundings that are available.

Summary of percentage:

```
> summary(newda$eff)
    Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
-48.6600   -1.0000   -0.6099   13.9800    2.2540 1423.0000        53
```

The summary above shows that the average of percentage is about 14. The summary shows the median and mean have a difference of 15, which shows that the distribution is not normal. This may due to the incompleteness of our information.

Error Bars Of Funding Regarding Research Type:



By using ggplot2, we have the error bar plot of different research types. From the plot above, we can see that Applied Research has the highest average of spend, while Service has the lowest. However, Applied Research also has the biggest SD since the error bar is the longest. Therefore, the funding demands of Applied Research vary most.

Query 5: Student Event Recommendation Filter

Description:
For people who attended/interested in a specific topic, this query would recommend certain upcoming events to these people who may be interested in attending more events. For example, if a student is interested in events about “software” then we can have the following query. It would also allow us to predict demands of events based on the specific topic.

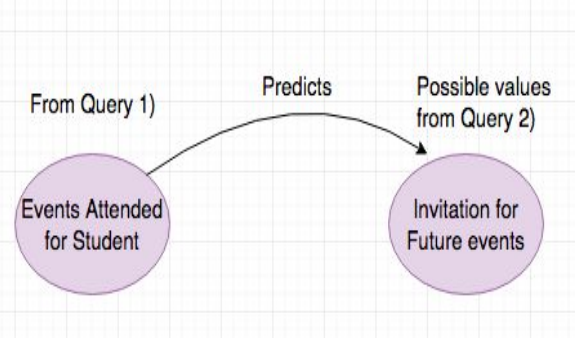
Benefits:
This query is specifically designed to help students find out which events they would be interested in. In doing so, we optimize benefits of events by increasing attendance and interest level.

Implementation Process:

1. From the queries, we get a list of events a student attended.
2. Factorize and combine all the corresponding vectors for matrix construction based on student’s attendance
3. Use K-means Clustering to classify each student to a future event based on nearest distance/interest

4. Send invitations to students based on which BIDS event/workshop the algorithm recommended

	2412	2634	2677	2795	...
Coding Workshop	4	0	5	3	
Graphics Design	0	5	0	0	
Data & Diversity	1	0	3	4	
Info Sessions	1	6	0	0	
...					



Sample Output:

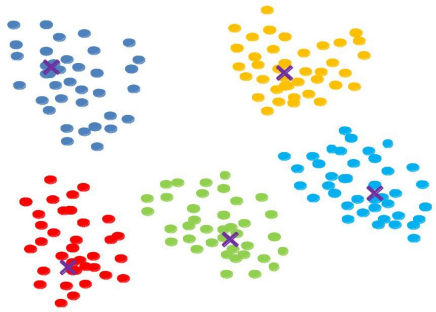


Fig. 13. Exemplary K-Means result

SQL :

```
Select E.Description, ae.EventID, COUNT(distinct ae.Person_ID)
From Event as E, AttendEvent as ae
Where E.EventID = ae.EventID
GROUP BY
E.Description, ae.EventID;

Select *
From Event as E
Where Description
LIKE '%software%';
```

Sample R script:

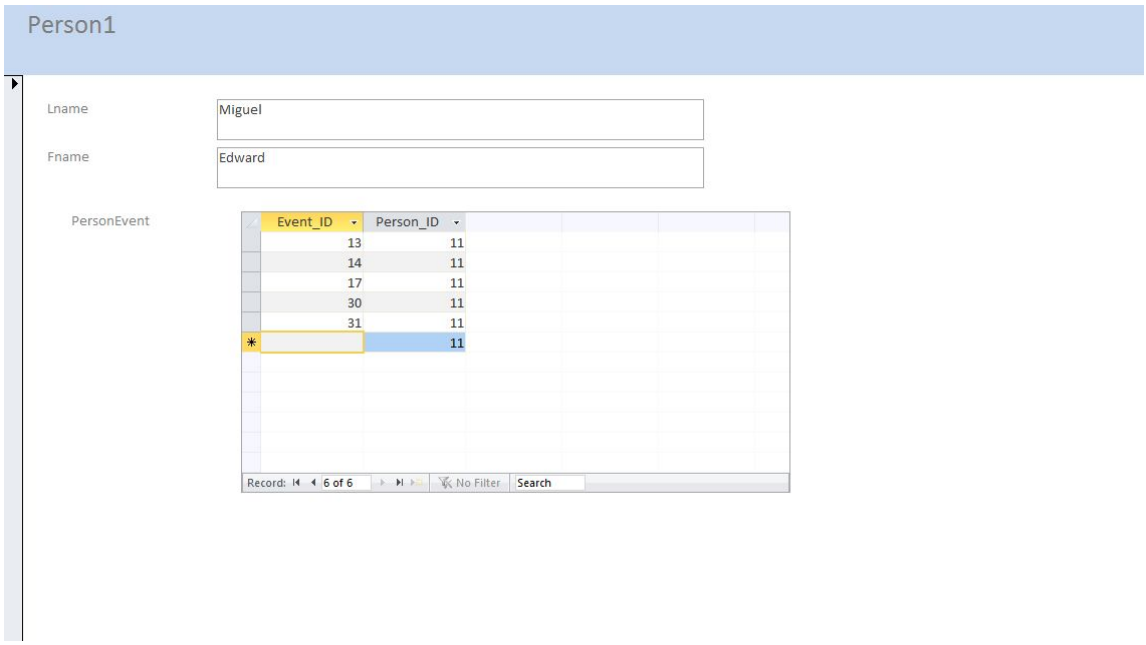
```
```{r}
library("cluster.datasets")
RawData <- read.table(....)
data_corpus <- Corpus(DataframeSource(data.frame(RawData[,2])))
tdm <- DocumentTermMatrix(data_corpus)
train <- as.matrix(tdm) %>%
  as.data.frame()

freq <- colSums(as.matrix(tdm))

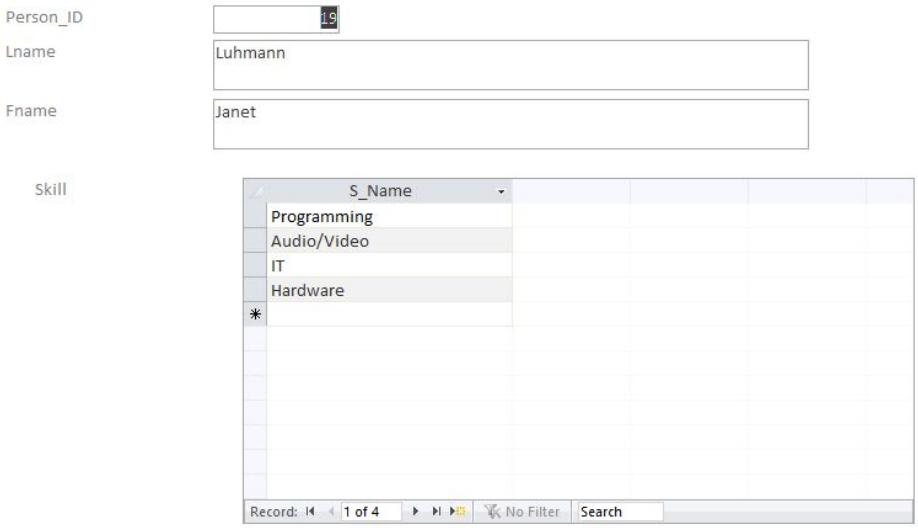
ktest <- train[,names(train) %in% names(freq)]
km.out <- kmeans(ktest, 5)
result <- km.out$cluster
```
```

III. Access Screen Shots

Form 1



This form is a data entry form that allows the database to enter in what events each person attends. This knowledge is very important for Query 5, which seeks to track event attendance in order to suggest future events to look out for.



This form performs similarly to the one above, allowing the database to keep track of skills that a person has, as it is important for the database to know this for Query 3 for team matching.

Report 1

Skill

S_Name	Person_ID	Lname	Fname
Audio/Video			
	19	Luhmann	Janet
	24	Burgmann	Roland
Documentation			
	20	Mathies	Richard
	23	Candida	Richard
Hardware			
	19	Luhmann	Janet
IT			
	23	Candida	Richard
	19	Luhmann	Janet
Languages			
	20	Mathies	Richard
	24	Burgmann	Roland
MS Office			
	25	Kammen	Daniel
Photoshop			
	29	Kolomensky	Yury
	30	Hawkins	Isabel
Programming			
	24	Burgmann	Roland
	19	Luhmann	Janet

Report 2

Attendance

Person_ID	Lname	Fname
11	Miguel	Edward
Event_ID	30	
Event_ID	17	
Event_ID	31	
Event_ID	14	
Event_ID	13	
15	Lee	Adrian
Event_ID	33	
Event_ID	2	
Event_ID	61	
16	Smoot	George
Event_ID	28	
17	Heathcock	Clayton
Event_ID	29	
18	Genzel	Reinhard
Event_ID	56	
Event_ID	78	
19	Luhmann	Janet
Event_ID	30	
20	Mathies	Richard
Event_ID	39	

IV. Functional Dependence

```
F={
Person_ID-> Lname, Fname, Birth_Date, University, Email_address,
Phone_number
Dep_Name-> Dep_Email, Dep_Address
Spending_ID, P_ID7 ->ID1, Amount
Amount ->Purchase
Spending_ID->Purchase
}
```

V. Normalization Analysis

1NF:

```
Course(Course_Num, dep_name2, Name, Prerequisites, Units, ID1E)
```

because Prerequisites can be multi valued.

To fix- Eliminate repeating groups:

```
Course(Course_Num, dep_name2, Name, Prerequisites, Units, ID1E)
→

Course_Info(Course_Num, dep_name2, Name, Units, ID1E)
+
Course_Prereq(Course_Num, Prerequisites)
```

2NF:

```
Spending(Spending_ID, P_ID7 ,ID1, Amount, Purchase)
```

Because Spending_ID is enough to know the Amount

To fix:

```
Spending(Spending_ID, P_ID7 ,ID1, Amount, Purchase) →

Spending_Info(Spending_ID, P_ID7 ,ID1, Purchase)
+
Spending_Amount(Spending_ID, Amount)
```

3NF:

```
Project(P_ID,Title, Date, Topic, Person_ID1, dep_name2)
```

**The attribute dep_name is determined by Person_ID ,which is determined by P_ID. In turn, dep_name is determined by P_ID transitively .*

To fix:

```
Project(P_ID,Title, Date, Topic, Person_ID1, dep_name2)  →

Project(P_ID, Title, Date, Topic, Person_ID1)
+
Proj_dep(Person_ID1, dep_name2)
```

BCNF:

```
Spending_Info(Spending_ID, P_ID7 ,ID1, Purchase)
Spending_Amount(Spending_ID, Amount)
```

Spending_ID is a unique identifier that would determine P_ID7 and ID1. We need to remove the partial dependencies that transitively dependent on the candidate key. Every determinant needs to be a candidate key.

```
Spending_Project(Spending_ID, P_ID7)
Spending_User(Spending_ID, ID1)
Spending_Info(Spending_ID, P_ID7 ,ID1, Purchase)
Spending_Amount(Spending_ID, Amount)
```

VI. Team Member Contribution

Team Member	Contribution
Avie Sanchez	**Query 1, Presentation Layout and Design**
Andrew Ding	**Query 4, Normalization Analysis**
Christian Choi	**EER, Relational Schema**
Christopher Liu	**Access Database, Relational Schema**
Pui Fung Lam	**Query 5, EER, Client Contact**
Xiaohan Wang	**Query 2, Relational Schema, R script**
Yunshan Guo	**Query 3, R script**

VII. Future Work

After the final presentation to both the professor and clients, we discussed future phase for this project. Some of these recommendations would serve as continuations/additional features for the database and they include:

1. Collaborating with BIDS to acquire necessary data- This would include reconciling pre-existing records or data mining to generate raw data into useful information.

2. Revising new changes- There might be different aspects that the design team have overlooked. Incorporated these changes into our database design would tailor to and benefit our client's need.
3. Implementing the queries to get our desire outputs- Some of the SQL queries have not been tested and in order to check if they are fully functional, the project team needs to implement and fine tune their queries.
4. Performing large-scale data analysis to find interesting patterns for reporting or improvement purposes- The purposes for our queries and analysis are to further assist BIDS in their operation efficiencies. Having an accurate prediction/inferences would allow them to continue or guide them in their future goals.