**DP Executive Summary**

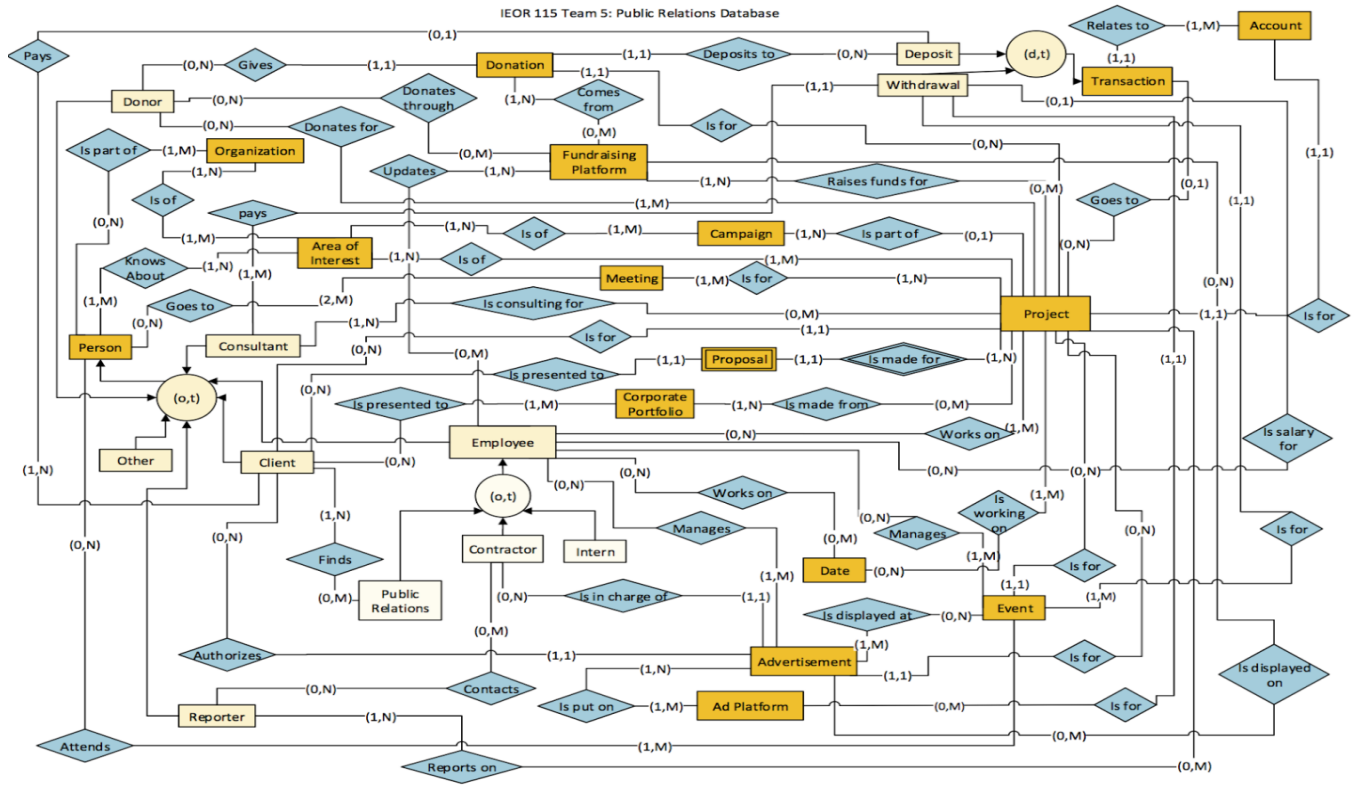
**Madera Group Public Relations Database Project**

Team 5: Nanavati Low, Chunchun Huang, Li Gu, Yikun Yang, Hing-Man Mak, Yueyilin Qi

## Client Description

Madera Group is a communications and development agency dedicated to helping clients reach and engage supporters and donors. It represents and advises global leaders as well as social entrepreneurs who are redefining solutions to tackle climate change, the local food revolution, education reform, new economic systems, green technology, and women's leadership. Madera Group's expertise includes big picture consultation for startups and social mission organizations. Their role is spans improvement in focused growth strategies, conference and special events planning, marketing, production and management, fundraising and development planning, donor research, and other online fundraising campaigns and social networking.

## Simplified EER




## Relational Design (Schema)

1. **Organization** (OrganizationID, Name, Website, City)
2. **Person** (PID, FirstName, MI, LastName, HomePhone, CellPhone, OfficePhone, FaxNo, Address)
   - 2a. **Other** (PID[2])
   - 2b. **Employee** (PID[2], SSN, Wage)
     - 2ba. **Public Relations** (PID[2b])
     - 2bb. **Intern** (PID[2b], IsWorking, AvailableHourPerWeek)
     - 2bc. **Contractor** (PID[2b], Specialty, Category, CommissionFromProject)
   - 2c. **Reporter** (PID[2], OrganizationSource)
   - 2d. **Client** (PID[2], StartTime, EndTime)
   - 2e. **Donor** (PID[2], AddToMailingList)
   - 2f. **Consultant** (PID[2], SSN, JobDescription, Wage)
3. **Meeting** (MeetingID, Date, Time, Location, Remark, DateofResponse)
4. **AreaofInterest** (Area)
5. **Donation** (DonationID, ProjectID[12], Amount, Source, ConfirmationNumber, SpecialRequest, PID[2e], DepositedInto[16a], DonationDate, Time, ReasonOfDonation)
6. **FundraisingPlatform** (FID, Name, Type, Time, Amount, No.ofView, PaymentMethod)
7. **CorporatePortfolio** (PortfolioID, CorporateDescription)

8. **Advertisement (**AdID, Name, No.ofPrint, No.ofViews, No.LikeShare, personInCharge[2bc], ProjectID[12])
9. **AdPlatform (**AdPlatformID, Name, Type)
10. **Event (**EventID, Name, Description, Date, Address, City, State, Zip, ProjectID[12], Type)
11. **Proposal (**ProjectID[12], ProposalID, presentedTo[2d], , ProposalAttachment)
12. **Project (**ProjectID, Type(Profit, Non-Profit, or Pro-Bono), StartDate, EndDate, Description, Budget, TargetAmount, Client[2d], Campaign[14])
13. **Date (**Date)
14. **Campaign** (CampaignID)
15. **ProjectAccount (**AccountNumber, Amount, ProjectID[12])
16. **Transaction** (TransactionNumber, AccountNumber[15], Date, Time, Amount, ProjectID[12])
    16a. **Deposit (**TransactionNumber[16], PaidByClient[2d])
    16b. **Withdrawal (**TransactionNumber**[16]**, isSalaryForEmployee[2b], isSalaryForConsultant[2f], isForEvent[10], isForAdPlatform[9]**)**

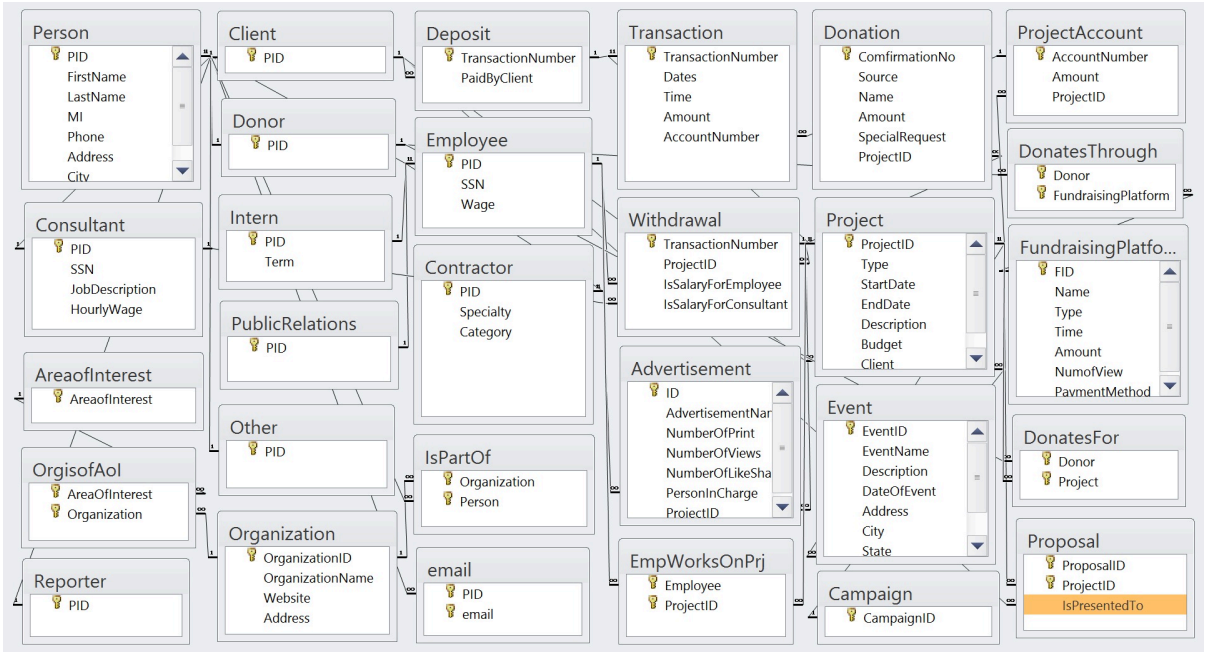
**N:M Relations**

17. **OrgIsofAoI** (AreaofInterest[4], Organization[1])
18. **IsPartOf** (Organization[1], Person[2])
19. **DonatesThrough** (Donor[2e], FundraisingPlatform[6])
20. **DonatesFor** (Donor[2e], Project[12])
21. **EmpWorksOnPrj** (PID[2b], ProjectID[12])
22. **EmpWorksOnDate** (PID[2b], Date[13],StartTime, EndTime)
23. **EmpManagesAd** (PID[2b], AdID[8])
24. **CPIsMadeFrom** (PortfolioID[7], ProjectID[12])
25. **CPIsPresentedTo** (ID[2d], PortfolioID[7])
26. **IsConsultingFor** (PID[2f], ProjectID[12])
27. **CampaignIsofAoI** (CampaignID[14],AreaOfInterest[4])
28. **MeetingIsForProject** (MeetingID[3],ProjectID[12])
29. **MeetingParticipant** (PID[2],MeetingID[3])
30. **IsPutOn** (AdPlatformID[9],AdID[8], StartDate, EndDate)
31. **ContactorContactsReporter** (PID[2bc], PID[2c], Day, Time, Description)
32. **EventManagedBy** (PID[2b], EventID[10])
33. **FundraisingPlatformRaisesForProject** (FID[6], ProjectID[12])
34. **DonationDonatedtoFundraisingPlatform** (Source[5], ConfirmationNumber[5], FID[6])
35. **EmployeeUpdatesFundraisingPlatform** (PID[2b], FID[6])
36. **ReporterReportsOnProjects** (PID[2c], Projects[12])
37. **PersonAttendsEvent** (PID[2,] EventID[10])
38. **AdvertisementIsDisplayedAtEvent** (AdID[8], EventID[10])
39. **AdvertisementIsDisplayedOnFundraisingPlatform** (AdID[8], FID[6])
40. **PersonIsInterestOfAOI** (PID[2], Area[4])
41. **ProjectIsOfAoI** (ProjectID[12], Area[4])
    42. **PRFindsClient** (PID[2ba], Client[2d])
    43. **ProjectIsWorkingOnDate** (ProjectID[12], Date[13])

**Multivalued Attribute(s)**

44. **Email** (PID[2], Email)
45. **InternSkills** (PID[2bb], Skill)

**Implemented Relational Schema (Table View)**



**Queries Description**

**Query 1: Trip Optimization by Maximizing Potential Donation**

*Determine which donors to visit on one business trip and maximize expected donation.*

Create a table for each donor's donation history with attributes donor ID, amount, area of interest, and time of donation. In Access, allow Madera Group to input surrounding ZIP code of visitation as a parameter to narrow down the number of potential donors.

Export the table of potential donors to visit in CSV format and import into R. Use R to find the potential donation amount by assuming there's an exponential decay after each donation with a rate depending on the amount of the donation. Find these rates using $f(t) = P_o e^{-rt}$ with $P_o$ = the donation amount, and for $t$ = time between this and next donation, set f(t) <<1. Do not consider donors with large $f(t)$ from the last donation(i.e.(f(t) >>1 where $t$ = time between last donation and project end date). Otherwise, use least squares regression to predict the donation amount from a plot of donation amount vs. time between donations using past data on the donor. Use the top n ranked potential donations (n is the number of donors to visit, specified by Madera Group) in a knapsack formulation written as strings in R and exported to CPLEX as a .mod file to solve.

**SQL Code 1**

```
SELECT d.PID, dt.Amount, aoi.Area,
dt.DonationDate, p.ZIP
FROM donor AS d, person AS p,
personisinterestofaoi AS aoi,
donation AS dt
WHERE (donation.PID)=d.PID) AND
d.PID)=p.PID) AND (aoi.PID)=p.PID)
AND ((aoi.Area)='Environmental'))
AND Donation.DonationDate >=
#12/4/2011#;
```

**Query 2: Project Assignment Allocation**

*Find the best assignment of project tasks for each intern by minimizing the total cost of salary paid to interns.*

Each project includes several assignments that need to be completed based on the skillset, availability, and wage of possible interns. Assume that each intern is able to finish the assignments they are allocated, that each assignment requires only one specific skill, and that each intern works only one shift on each day that they work. Use SQL to extract the required information and export the table in CSV format. Set up a linear programming model and solve in AMPL with CPLEX.

**SQL Code 2**

**InternSkillsets**

```
SELECT s.InternPID, s.Skill
FROM InternSkills s, Intern i
WHERE i.IsWorking AND i.PID =
s.InternPID;
```

**InternAvailability**

```
SELECT PID, AvailableHourPerWeek
FROM Intern
WHERE IsWorking;
```

**HourlyWage**

```
SELECT i.PID, e.Wage
FROM Employee AS e, Intern AS i
WHERE i.IsWorking AND i.PID =
e.PID;
```

**Assignment**

```
SELECT Assignment, HourDemand
FROM Assignment
WHERE ProjectID = XXX;
```

**Decision Variable**

$X_{ij}$ = the number of hours that intern $i$ is assigned to work on assignment $j$

**Parameters**

$\text{Skill}_{ij} = \begin{cases} 1 \; if \; intern \; i \; has \; skill \; j \; to \; accomplish \; assignment \; j \\ 0 \; otherwise \end{cases}$

$\text{HourlyWage}_i$ = the hourly wage of intern i

$\text{Availability}_i$ = the number of hours that intern i is available to work for a certain week

$\text{Demand}_j$ = the number of hours that is needed to finish assignment j

**Objective Function**

$\min \sum_{i=1}^{m} HourlyWage_i * \left(\sum_{j=1}^{n} X_{ij}\right)$

**Constraints**

$\sum_{j=1}^{n} X_{ij} \leq Availability_i, \forall i = 1, \dots, m$

$\sum_{i=1}^{m} X_{ij} \geq Demand_j, j = 1, \dots, n$

$X_{ij} \leq Demand_j * Skill_{ij}, \forall i = 1, \dots, m, j = 1, \dots, n$

$X_{ij} \geq 0, \forall i = 1, \dots, m, j = 1, \dots, n$

**Query 3: Non-linear Least Squared Regression to Sort Donations by Event**

*Find total donation amount and ROI of each event for a project.*

Use SQL to extract donation frequency by day for a given project and plot a graph of donation frequency versus time in R. Fit each specific event to a Rayleigh distribution using nonlinear least squares regression for all events, assuming the donation frequency peaks after an event and decreases over time. Assume the effect of each event is proportional; subtract the graph of all previous events to find the non-confounded frequency versus time graph for each event. This graph can then be used to find the total donations per event. Divide this by the cost of the event to find out the return on investment for Madera Group's future reference.

**SQL Code 3**

```
CREATE VIEW ad(DateofAd, ProjectID, AdvertisementName, EstTotalAmountRaised) as
SELECT a.DateofAd, a.ProjectID, a.AdvertisementName, a.EstTotalAmountRaised
FROM Advertisement AS a, Employee AS e, Contractor AS c
WHERE a.PersonInCharge = c.PID AND e.PID = c.PID AND c.PID = 8;

CREATE VIEW commission(ProjectID, TotalCommission) as
SELECT d.ProjectID, sum(d.Amount*c.Commission)
FROM donation AS d, ContractorReceivesCommissionFromProject AS c, Advertisement AS a
WHERE c.PID = 8 AND a.PersonInCharge = c.PID AND a.ProjectID = d.ProjectID
GROUP BY d.ProjectID;
```

**Query 4: Return on Investment and Forecasting for Contractors**

*Which contractor should Madera Group hire for a project?*

Calculate ROI for each contractor (those who help hold events, make videos, etc.) per project using

$$ROI = \frac{Salary}{Fundraising\ amount\ raised}$$

(taken from a result of a query similar to Query 3 but only using data of events related to one contractor). Plot ROI vs. time over all project finish dates for one contractor and use R and non-linear least squares regression to fit this to a logarithmic curve of the form a+b*log(x) by assuming that a contractor will improve his/her ROI with more experience, but with a decreasing increase over time. This query can also compare the ROIs of different contractors on the same task by plotting all curves on one graph. Ultimately this can be used to predict which contractor to use and how much Madera Group can expect to pay them by ranking the predicted ROIs at a particular time.

**SQL Code 4**

**Total Donations**

```
CREATE VIEW [Query 4] AS
SELECT d.ProjectID, Sum(d.Amount) AS TotalAmount
FROM donation AS d
GROUP BY d.ProjectID;
```

**Ads with contractorX in charge**

```
CREATE VIEW [Query4a] AS
SELECT a.DateofAd, a.projectID, a.AdvertisementName, a.EstTotalAmountRaised
FROM Advertisement AS a, Employee AS e, Contractor AS c
WHERE a.PersonInCharge = c.PID AND e.PID = c.PID AND c.PID = 8;
```

Total commission of projects for contractorX

```
CREATE VIEW [Query4b] AS
SELECT Query4.ProjectID, (Query4.TotalAmount*c.Commission) AS
```

**TotalCommission**

```
FROM Query4, ContractorReceivesCommissionFromProject AS c, Advertisement AS a
WHERE (((c.PID)=8) AND ((a.PersonInCharge)=[c].[PID]) AND ((a.ProjectID)=[Query4].[ProjectID])) AND c.ProjectID = Query4.ProjectID;
```

**Combined**

```
SELECT DISTINCT a.DateofAd, a.projectID, a.AdvertisementName
AS AdType, a.EstTotalAmountRaised, b.TotalCommission,
(a.EstTotalAmountRaised/b.TotalCommission) AS ROI
FROM Query4a AS a, Query4b AS b
WHERE a.ProjectID = b.ProjectID;
```

**Query 5: Portfolio Optimization**

*Which events should be held considering how events affect one another?*

Determine which event(s) to hold while minimizing the risk associated with holding a certain set of events together given a target fundraising amount and advertising budget. We try to adopt the optimal portfolio model from E120.

Let $y_i$ be 1 if advertisement/event i is taken, 0 otherwise;

$R_i$ = expected rate of return for each event, calculated based on the past data for events of same type and area of interest [(return – cost)/cost]

ρ = (target fundraising amount)/budget

$B_i$= event's cost over budget

$$min \sum_j \sum_i B_i * B_j * COV(R_i * y_i, R_j * y_i)$$

$$\text{st } \sum_i R_i * y_i \geq \rho \text{ and } \sum_i B_i * y_i \leq 1 \text{ for } y_i \in \{0,1\}$$

We can find the covariance matrix using all past events' rate of return grouped by type (e.g, a Youtube video and a banquet) for the area of interest of the current project from the results in Query 3 and using R. By minimizing the weighted covariance, the risk of spending on a certain event while meeting expected fundraising amount is minimized.

**SQL Code 5**

```
SELECT e.Type,w.isForEvent
FROM Withdrawal as w, Event as e, ProjectIsOfAoI as pa, Transaction as t
WHERE pa.Area='XXX' AND w.TransactionNumber = t.TransationNumber AND t.ProjectID = pa.ProjectID AND e.ProjectID = pa.ProjectID
GROUP BY e.Type
UNION
SELECT ad.Type,w.isForAdPlatform
FROM Withdrawal as w, IsPutOn as ad, Advertisement as a, ProjectIsOfAoI as pa, Transaction as t
WHERE pa.Area='XXX' AND w.TransactionNumber = t.TransationNumber AND t.ProjectID = pa.ProjectID AND a.ProjectID = pa.ProjectID AND a.AdID = ad.AdID
GROUP BY ad.Type;
```

## Client Introduction

Founded over 10 years ago, Madera Group is a private consulting company that fills the gaps between marketing, brand awareness, and strategy growth for non-profit organizations. They focus on methods to engage the public through optimizing outreach strategies and leveraging social media to carry their clients' messages to broad audiences. Additionally, Madera actively mines and maintains supporter and donor groups across all channels through coordinated technology platforms in order to help broadcast their clients' message and mission.

## Previous Approach and Goals

Throughout the semester, the project took several pivotal turns in order to adjust and tailor our database specifically for our client. Namely, the EER diagram and consequently database design either added or further broke down the donation, project, people, area of interest, and transaction entities.

In order to encompass all the different services offered by Madera Group, our previous iterations shifted focus from advertising to include specific entities such as Donation, Project, and Person. While target marketing is an important aspect of the company, we wanted to have a more flexible framework for corporate expansion in the future. Furthermore, documenting donation attributes such as source, date, time, etc. allows us to further mine and retrieve interesting query results such as determining fundraising event selection considering the amount raised and correlation to other events.

Another idea we decided to incorporate into our database design was forecasting and predicting user data. In order to analyze this data, we added the entity Area of Interest. This allows us to track and group projects as well as distinct roles of people affiliated with Madera's organization operations. Translated into query form, this entity allows us to determine which contractor to hire for a certain project and the areas of interest Madera Group makes the most profit for make future investments. Without this entity, we originally had area of interest as an attribute inside the project entity but soon noticed the limitations it created in generating relationships to other entities and use in queries.

Our original entity to keep track of money flows in a project was called Deposit/Cash Flows, which was only related to the Client entity. This was designed because our main focus was on managing employees and advertisements. However, there are more relationships that should be connected to the money flow, such as donations, employee salary, and project expenses. Thus the ProjectAccount and Transaction entities were created to store cash flow information for a specific project.

In order to keep better track of employees' contribution to a project, we divided employees by their responsibilities (eg. public relations, contractor, intern) instead of working hours (full-time, part-time, intern) as we originally had. That approach would not be able to standardize a metric to evaluate categorized roles. For example, contractors can be hired either part-time or full-time by Madera Group. Furthermore, one of our queries shows the need of this differentiation to compare hired contractors with one another.

## Query Description

### Query 1: Trip Optimization by Maximizing Potential Donation

*Justification*

When planning a business trip for either a member of Madera Group or a client for the purpose of donor visits, the traveler needs to know which donors are expected to donate the largest amount and plan accordingly to fully utilize the trip and obtain the maximum potential donation.

*Implementation*

Determine which donors to visit on one business trip to maximize potential donations by using exponential decay to estimate potential donation amount and linear programming.

*1. SQL*

First in Access, allow Madera Group to input the surrounding ZIP Code of visitation as a parameter to narrow the number of donors involved.

For each potential donor, create a table of donation history with attributes donor(PID), amount, area of interest, time of donation and location(ZIP).

***SQL Code***

```
SELECT d.PID, donation.Amount, aoi.Area, donation.DonationDate, p.ZIP
FROM donor AS d, person AS p, personisinterestofaoi AS aoi, donation
WHERE ((([donation].[PID])=[d].[PID]) AND (([d].[PID])=[p].[PID])
 AND ((aoi.PID)=[p].[PID]) AND ((aoi.Area)='Environmental'))
AND Donation.DonationDate >= #12/4/2011#;
```

Query1

| PID | Amount | Area | DonationDat | ZIP |
|---|---|---|---|---|
| 15 | $580.00 | Environmental | 1/22/2012 | 94709 |
| 15 | $400.00 | Environmental | 9/10/2012 | 94709 |
| 15 | $495.00 | Environmental | 2/27/2013 | 94709 |
| 15 | $220.00 | Environmental | 8/10/2013 | 94709 |

**Figure 1.1 SQL resulting table for one hypothetical donor**

*2. R*

We first export the table in CSV format for every potential donor and import it into R to find who are the potential donors and their expected donation amount (used later as parameters for a linear program.) We assume that there is an exponential decay after each donation made with a rate depending on the amount of the donation. Find these rates using $f(t) = P_o e^{-rt}$ for each donation per donor with $P_o$ = the donation amount, and for $t$ = time between this and the next donation. Set f(t) <<1 at the time of the next donation. For the last donation, let the rate of decay, r, be the average of the previous rates of decay. We discard donors from consideration who have large $f(t)$ values using the $f(t)$ equation for their last donation (i.e.(f(t) >>1 where $t$ = time between last donation and the project end date), meaning not sufficient time has passed since the donor made the last donation and thus does not have potential to donate during the time frame of the project. See Figure 1.2 This graph has black lines indicating the exponential decay of the donations. If the project end date is at the green vertical line, for example, we discard this donor since $f(t)$ is not sufficiently small yet. If project end date is, say, at the blue line, we keep the donor since the intersection of the blue line and the most recent decay is very small (<<1).

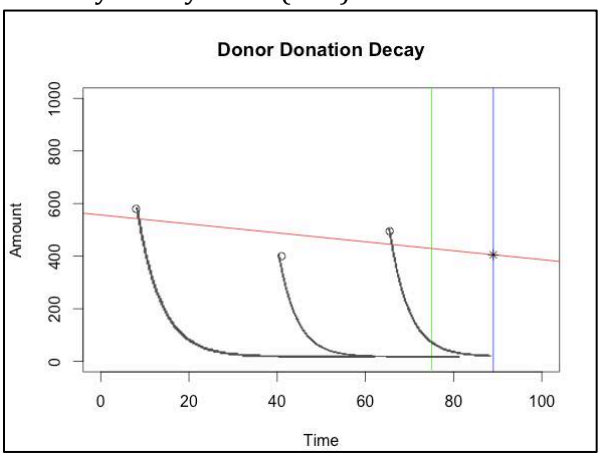



**Figure 1.2 Predicting expected donation amount for the potential donors in R**

If the donor has potential to donate, we use a linear regression to predict the donation amount from a regression on a plot of donation amount vs. time between donations using past data on the donor. In this example, the red line stands for the linear regression and it intersects the blue

line (which represents the end time of the project) at 405, indicating the expected donation amount is $405 for this donor.

*3. AMPL with CPLEX*

After acquiring the expected donation amount for all potential donors, we use the donors with top n ranked potential donations (n is the number of donors to visit, specified by Madera Group) in a knapsack integer-programming problem. The object function is to maximize the total donation amount subject to a time constraint as follows:

Let
$x_i = 1$ if donor $i$ is visited, 0 otherwise
$D_i =$ the expected donation amount of donor $i$
$T_i =$ the time required to visit donor $i$
$A =$ the total time allotted for donor visitation on trip
$\max \sum_i x_i * D_i$
s.t. $\sum_i x_i * T_i \leq A$ and $x_i \in \{0,1\}$

We formulate the integer-program and data generated from R into .mod and .dat files and enter them into AMPL to solve using CPLEX.

We expect to generate AMPL outputs with 0-1 binary variables indicating whether Madera Group should visit a certain donor during this trip or not as well as the total expected donation. The following figures show a sample result with total expected donation of $7,600 coming from three potential donors (labeled as X[1], X[3], and X[5] in the AMPL output) on their 3-day trip.

```
set l;

param ExpectedDonation{i in l};
param TimeSpent{i in l};
param TotalAval;

var X{i in l} binary;

maximize totalExpectedAmount:
        sum{i in l} ExpectedDonation[i]*X[i];

subject to

Constraint1: sum{i in l}X[i]*TimeSpent[i] <= TotalAval;
```

**Figure 1.3 AMPL Code**

```
ampl: option solver "/Users/apple/Desktop/AMPL160/cplex";
ampl: model "/Users/apple/Downloads/Query1.mod";
ampl: data "/Users/apple/Downloads/query1.dat";
ampl: solve;
CPLEX 12.6.0.1: optimal integer solution; objective 7600
1 MIP simplex iterations
0 branch-and-bound nodes
ampl: display _varname, _var.val;
: _varname _var.val    :=
1   'X[1]'     1
2   'X[2]'     0
3   'X[3]'     1
4   'X[4]'     0
5   'X[5]'     1
6   'X[6]'     0
;
```

**Figure 1.4 AMPL Result**

**Query 2: Project Assignment Allocation**

*Justification*

We'd like to come up with the best match overall between intern skillset and company demands while minimizing the cost of the total salary paid to interns. This intern allocation method is designed to ensure all tasks are efficiently covered and that interns are only assigned tasks that they are able to do.

*Implementation*

The assumptions in this query are that each intern is able to finish the assignments that they are allocated, each assignment only requires one specific skill, multiple interns can work on the same assignment and one intern can work on multiple assignments, each intern works only one shift on each day that they work. The second assumption is based on one limitation of this query: we cannot assign tasks that involve multiple skills; this will be covered further in the Future Work section.

*1. SQL*

In order to incorporate all factors that will influence the distribution of assignments to interns, four tables with information about intern experience and assignment demands were extracted using SQL code and an example of the results in Access:

**InternSkillsets**

```
SELECT s.InternPID, s.Skill
FROM InternSkills s, Intern i
WHERE i.IsWorking AND i.PID = s.InternPID;
```

**InternAvailability**

```
SELECT PID, AvailableHourPerWeek
FROM Intern
WHERE IsWorking;
```

**HourlyWage**

```
SELECT i.PID, e.Wage
FROM Employee AS e, Intern AS i
WHERE i.IsWorking AND i.PID = e.PID;
```

**Assignment**

```
SELECT Assignment, HourDemand
FROM Assignment
WHERE ProjectID = XXX;
```

Query2a

| InternPID | Skill |
|---|---|
| 6 | Writing |
| 6 | Graphic Design |
| 6 | Microsoft Office |
| 6 | Finance |
| 6 | Data Analyzing |
| 7 | Writing |
| 7 | Social Networking |
| 7 | Research |
| 7 | Graphic Design |
| 7 | Leadership |
| 7 | Data Analyzing |
| 10 | Microsoft Office |
| 10 | Social Networking |
| 10 | Access |
| 10 | Graphic Design |
| 10 | Finance |
| 10 | Leadership |
| 10 | Data Analyzing |
| 11 | Writing |
| 11 | Microsoft Office |
| 11 | Social Networking |
| 11 | Research |
| 11 | Access |
| 11 | Finance |
| 11 | Leadership |
| 11 | Data Analyzing |

Query2b

| PID | AvailableHou |
|---|---|
| 6 | 11 |
| 7 | 15 |
| 10 | 12 |
| 11 | 18 |

Query2c

| PID | Wage |
|---|---|
| 6 | $12.00 |
| 7 | $13.00 |
| 10 | $12.00 |
| 11 | $15.00 |

Query2d

| ProjectID | Assignment | HourDemand |
|---|---|---|
| 92818 | Blogging | 4 |
| 92818 | Database Maint | 10 |
| 92818 | Event Promotio | 3 |
| 92818 | Donor Research | 5 |
| 92818 | Data Collection | 10 |
| 92818 | Web Page | 4 |
| 92818 | Accounting | 5 |
| 92818 | Event Planning | 6 |
| 92818 | Data Analysis | 10 |

Figure X. Query 2a is InternSkillsets, 2b is InternAvailability, 2c is HourlyWage, and 2d is Assignment

It is important to note that query results for 2a, 2b, and 2c are only for interns who are currently working since only those interns could work on any projects. These queries pull out which interns have which skills, the number of hours they are available, and their hourly wage. The last query is to find all current assignments that interns could work on and the hours needed for each.

With all the data pulled out as tables using SQL and saved in CSV files, we can build a linear programming model to solve this task assignment problem. The next figure summarizes this program and is explained in detail below:

**Decision Variable**

$X_{ij}$ = the number of hours that intern i is assigned to work on assignment j

**Parameters**

$$Skill_{ij} = \begin{cases} 1 \ if \ intern \ i \ has \ skill \ j \ to \ accomplish \ assignment \ j \\ 0 \ otherwise \end{cases}$$

$HourlyWage_i$ = the hourly wage of intern i

$Availability_i$ = the number of hours that intern i is available to work for a certain week

$Demand_j$ = the number of hours that is needed to finish assignment j

**Objective Function**

Min $\sum_{i=1}^{m} HourlyWage_i * \left(\sum_{j=1}^{n} X_{ij}\right)$

**Constraints**

$\sum_{j=1}^{n} X_{ij} \leq Availability_i, \forall i = 1, \dots, m$

$\sum_{i=1}^{m} X_{ij} \geq Demand_j, j = 1, \dots, n$

$X_{ij} \leq Demand_j * Skill_{ij}, \forall i = 1, \dots, m, j = 1, \dots, n$

$X_{ij} \geq 0, \forall i = 1, \dots, m, j = 1, \dots, n$

**Figure 2.1 Linear Programming Model Formulation**

The decision variable is $X_{ij}$, denoting the number of hours that intern *i* is assigned to work on assignment *j*. There are 4 parameters, each representing the data stored in the 4 tables resulting from SQL queries. Based on the purpose of this query, the objective function is to minimize the total salary paid to interns, which can be evaluated as the sum of the hourly wage of an intern times the number of hours they work on each task they are assigned.

The first constraint of this formulation is a limitation on the availability of each intern per week. The second constraint is to satisfy the number of hours that each task requires. The third constraint makes sure that an intern has the skill to accomplish the task they are assigned to by including the binary parameter $Skill_{ij}$.

After the formulation, we can transport all the data and the linear programming model into a .dat file and a .mod file that can be solved using CPLEX in AMPL. In this particular example, we have 4 interns and 9 assignments to be distributed.

```
set m;
set n;

param Skill{i in m, j in n};
param HourlyWage{i in m};
param Availability{i in m};
param Demand{j in n};

var X{i in m, j in n}>=0;

minimize SalaryPaid:
        sum{i in m} HourlyWage[i]*(sum{j in n} X[i,j]);

subject to
totalhours {i in m}: sum{j in n} X[i,j] <= Availability[i];
assignment {j in n}: sum{i in m} X[i,j] >= Demand[j];
intern1skill {i in m, j in n}: X[i,j] <= Demand[j]*Skill[i,j];
```

**Figure 2.2 Query2 AMPL formulation**

```
set m := 1 2 3 4;
set n := 1 2 3 4 5 6 7 8 9;

param Skill :  1   2   3   4   5   6   7   8   9:=
          1    1   1   0   0   0   1   1   0   1
          2    1   0   1   1   0   1   0   1   1
          3    0   1   1   0   1   1   1   1   1
          4    1   1   1   1   1   0   1   1   1;

param HourlyWage :=
1    12
2    13
3    12
4    15;

param Availability :=
1    11
2    15
3    12
4    18;

param Demand :=
1    4
2    7
3    3
4    5
5    10
6    4
7    5
8    6
9    10;
```

**Figure 2.3 Query2 .dat file for this particular example**

After inputting the model and data into AMPL, the final result is given as follows. The AMPL output shows which interns (labeled as columns 1 through 4) and the number of hours scheduled for each are assigned to which tasks (labeled as rows 1 to 9) . For example intern 2 works on assignment 1, 4, 6, and 8 for different number of hours and the model ensures that intern 2 has all the skills to accomplish these tasks. Also, assignment 1 (blogging) is allocated to intern 1, 2, and 4, ensuring that all interns 1, 2, and 4 have the skills to work on assignment 1. The optimal objective function value in this particular case is $711, the minimum total salary paid to all interns. In this case, query2 is economically useful for Madera Group because it can generate the most efficient and economic way to schedule assignments to interns.

```
ampl: option solver "/Users/apple/Desktop/AMPL160/cplex";
ampl: model "/Users/apple/Desktop/Query2/query2.mod";
ampl: data "/Users/apple/Desktop/Query2/query2.dat";
ampl: solve;
CPLEX 12.6.0.1: optimal solution; objective 711
16 dual simplex iterations (0 in phase I)
ampl: display X;
X [*,*] (tr)
:   1    2   3   4     :=
1    1   2   0    1
2    0   0   7    0
3    0   0   3    0
4    0   5   0    0
5    0   0   0   10
6    0   2   2    0
7    0   0   0    5
8    0   6   0    0
9   10   0   0    0
;

ampl: display SalaryPaid;
SalaryPaid = 711
```

**Figure 2.4 Query 2 AMPL Output**

**Query 3: Non-linear Least Squares Regression to Sort Donations by Event**

*Justification*

Madera Group can determine the effectiveness of each event by determining an estimated donation amount generated by each event and compare ROIs across events to find the most successful ones for future reference.

*Impementation*

*1. SQL*

Use SQL to one table with total number of donations made and total amount of these donations grouped by day, with the number of days that have passed since the start of the project; extract another table with the time each event happened, again, in the form of days since the start of the project and the respective cost and type of the event. Below is the SQL code and resulting tables :

***SQL Code***

***DonationData***

```
SELECT DATEDIFF("d", p.StartDate, d.DonationDate) AS DS, COUNT(d.Amount) AS Total,
SUM(d.Amount) AS Amount
FROM Donation d, Project p
WHERE p.ProjectID=### AND d.ProjectID = p.ProjectID
GROUP BY DATEDIFF("d", p.StartDate, d.DonationDate);
SORT BY (d.Day - p.StartDate) AS DS ASD;
```

***EventData***

```
SELECT e.EventID, e.Type, DATEDIFF("d", p.StartDate, e.DateOfEvent) AS DS
FROM Event e, Project p
WHERE p.ProjectID=e.ProjectID AND p.ProjectID=###;
```

```
UNION
SELECT a.AdID, ad.Type, DATEDIFF("d", p.StartDate, a.DateofAd) AS DS
FROM Advertisement a, AdPlatform ad, AdIsPutOnAdplatform ada, Project p
WHERE p.ProjectID = ### AND a.ProjectID = p.ProjectID AND ada.AdID = a.AdID AND
ad.AdPlatformID = ada.AdPlatformID;
```

Query3a

| DS | Total | Amount |
|---|---|---|
| 2 | 2 | $200.00 |
| 3 | 4 | $380.00 |
| 4 | 3 | $250.00 |
| 5 | 1 | $100.00 |
| 6 | 1 | $80.00 |
| 8 | 1 | $100.00 |
| 9 | 2 | $150.00 |
| 11 | 2 | $200.00 |
| 13 | 1 | $110.00 |
| 20 | 7 | $650.00 |
| 21 | 8 | $770.00 |
| 22 | 3 | $320.00 |
| 23 | 2 | $230.00 |
| 24 | 1 | $140.00 |
| 25 | 1 | $50.00 |

Query3b

| EventID | Type | DS |
|---|---|---|
| 21000 | Youtube | 1 |
| 21001 | Print | 8 |
| 40182 | Fund. Dinner | 20 |

**Figure 3.1: Resulting tables from Access with "DS" standing for days since project started**

*2. R*

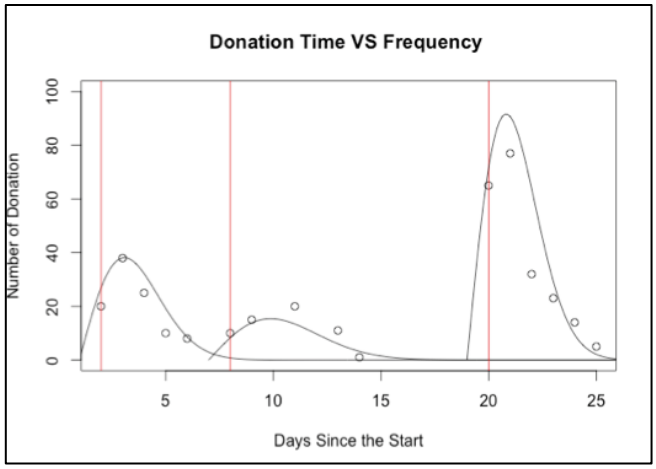
Export these tables in CSV format and export into R. Here we used a larger set of data than the one generated by our query to better imitate a realistic setting (we used Excel to generate the .csv file instead of Access, but this should come directly from Access as Figure 3.1 shows in a working version of this database.):

| | A | B | C |
|---|---|---|---|
| 1 | DS | Total | Amount |
| 2 | 2 | 20 | 2000 |
| 3 | 3 | 38 | 3800 |
| 4 | 4 | 25 | 2500 |
| 5 | 5 | 10 | 1000 |
| 6 | 6 | 8 | 800 |
| 7 | 8 | 10 | 1000 |
| 8 | 9 | 15 | 1500 |
| 9 | 11 | 20 | 2000 |
| 10 | 13 | 11 | 1100 |
| 11 | 14 | 1 | 100 |
| 12 | 20 | 65 | 6500 |
| 13 | 21 | 77 | 7700 |
| 14 | 22 | 32 | 3200 |
| 15 | 23 | 23 | 2300 |
| 16 | 24 | 14 | 1400 |
| 17 | 25 | 5 | 500 |

| | A | B | C | D |
|---|---|---|---|---|
| 1 | AdID/Event | Ad/Event T | DS | Cost |
| 2 | 20036 | Youtube | 1 | 8000 |
| 3 | 20124 | Print Ad | 8 | 4500 |
| 4 | 40286 | Fund. Dinne | 20 | 12500 |
| 5 | | | | |

**Figure 3.2: The actual set of data used in regression in R**

Implement this data into R to find the parameter for the best-fit Rayleigh distribution of donation frequency for each event using the nonlinear least squares regression function (nls). Our assumption is that the number of donations made after each event reflects the effectiveness of the event. In a short time after an event took place, a large number of donations will be made and as time progresses the number of donations decreases, following a curve with the shape similar to that of the Rayleigh distribution multiplied by some factor to increase it's height beyond 1 (since it is a probability density function). Shown is a plot of the original data of donation frequency vs. time for a particular project as well as the best-fit Rayleigh distribution curve with the time of events as red vertical lines. The corresponding R code is also included.

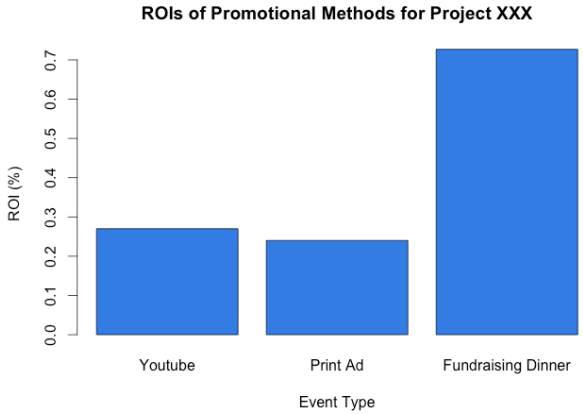



```r
TotalDonation = read.csv("query3_1-1.0.csv")
e = read.csv("query3_2-1.0.csv")
tf=c(e$DS,max(TotalDonation$DS)) # timeframe for the events, starting from 0,
                    #counting each event date, ending at last donation time
dt=TotalDonation# donations made
Return=NULL
plot(dt$DS,dt$Total, ylim=c(0,100), xlab='Days Since the Start', ylab='Number of Donation
     main='Donation Time VS Frequency ')
l=length(TotalDonation$Total)
for(i in 1:nrow(e)) {
  abline(v=e$DS[i], col='red')
  y=dt$Total[dt$DS<tf[i+1]]
  x=dt$DS[dt$DS<tf[i+1]]
  mini=min(x)-1
  x=x-mini
  RD=y~(x/s^2)*exp(1)^(-x^2/(2*s^2))*sum(y)*1.3
  fit=nls(RD,start=list(s=1))
  new = data.frame(x = seq(0,max(TotalDonation$DS),len=200))
  lines(new$x+mini,predict(fit,newdata=new))
  s=coef(fit)
  x=dt$DS[dt$DS>mini]
  x=x-mini
  m=(x/s^2)*exp(1)^(-x^2/(2*s^2))*sum(y)*1.3
  RD=y~(x/s^2)*exp(1)^(-x^2/(2*s^2))*sum(y)*1.3
  dt$Total=dt$Total-m
  dt=dt[dt$DS>=tf[i+1],]
  m=c(rep(0,l-length(m)),m)
  Return=cbind(Return,m)
}
FunctionSum=NULL
for(i in 1:l) {
  FunctionSum=c(FunctionSum,sum(Return[i,]))
}
EventReturn=NULL
for(i in 1:nrow(e)) {
  EventReturn=c(EventReturn,sum(TotalDonation$Amount*(Return[,i]/FunctionSum)))
}
```

**Graph 3.3 Donation Frequency vs. Time graph, with best fit Rayleigh Distribution and R Code**

Knowing this distribution we can find the estimated total donation amount of each event. For an event after more than one event has taken place, we regard the total amount of donations received as an aggregate result of all events. Before finding the corresponding Rayleigh distribution, we subtract the preceding distributions. For a particular day after more than one event has happened, we split up the total donation amount according to the percentage of total height each Rayleigh curve takes up. We allocate the total donation amount to each event that day by multiplying the number of donations by this percentage for each event. This amount gets stored in the database as EstTotalAmntRaised in the Event and Advertisement tables. Divide this result by the cost of the event to find the ROI for each event. Madera Group can see which type of event has the highest ROI and determine future project procedures accordingly, which will be discussed in the following queries.

With this sample dataset, the costs of events are $8,000, $4,500 and $12,500 respectively, with total donations of approximately $10,200, $5,600 and $21,600. Thus the rate of return is 27%, 24% and 72.7%, with the fundraising dinner being the event with highest return and thus the most effective event for this project. Below is a bar plot generated by R for the calculated ROIs.



**Query4: Return on Investment and Forecasting for Contractors**

*Justification*

This query can help Madera group analyze and compare the performance of its contractors and choose those with most potential in terms of ROI for future projects as well as monitor each individual contractor's growth in terms of ROI over time.

*Implementation*

*1. SQL*

Construct a query to pull out the projects worked on, commission, and amounts fundraised (an output of query 3) for the same contractor. In this example, we use contractor 8 who helps Madera group create videos on Youtube. See below:

***SQL Code***

**Total Donations**

```
CREATE VIEW [Query 4] AS
SELECT d.ProjectID, Sum(d.Amount) AS TotalAmount
FROM donation AS d
GROUP BY d.ProjectID;
```

**Ads with contractorX in charge**

```
CREATE VIEW [Query4a] AS
SELECT a.DateofAd, a.projectID, a.AdvertisementName, a.EstTotalAmountRaised
FROM Advertisement AS a, Employee AS e, Contractor AS c
WHERE a.PersonInCharge = c.PID AND e.PID = c.PID AND c.PID = 8;
```

Total commission of projects for contractorX

```
CREATE VIEW [Query4b] AS
SELECT Query4.ProjectID, (Query4.TotalAmount*c.Commission) AS TotalCommission
FROM Query4, ContractorReceivesCommissionFromProject AS c, Advertisement AS a
WHERE (((c.PID)=8) AND ((a.PersonInCharge)=[c].[PID]) AND
((a.ProjectID)=[Query4].[ProjectID])) AND c.ProjectID = Query4.ProjectID;
```

**Combined**

```
SELECT DISTINCT a.DateofAd, a.projectID, a.AdvertisementName AS AdType,
a.EstTotalAmountRaised, b.TotalCommission, (a.EstTotalAmountRaised/b.TotalCommission)
AS ROI
FROM Query4a AS a, Query4b AS b
WHERE a.ProjectID = b.ProjectID;
```

Query4c

| DateofAd | projectID | AdType | EstTotalAmoı | TotalCommis | ROI |
|---|---|---|---|---|---|
| 11/4/2014 | 97125 | Youtube | $920.00 | 893 | 1.03023516237402 |
| 11/7/2014 | 97839 | Youtube | $823.00 | 1754 | 0.46921322690992 |
| 11/9/2014 | 97584 | Youtube | $923.00 | 169 | 5.46153846153846 |
| 11/11/2014 | 97236 | Youtube | $1,930.00 | 755 | 2.55629139072848 |
| 11/17/2014 | 97238 | Youtube | $192.00 | 769 | 0.249674902470741 |
| 11/17/2014 | 97283 | Youtube | $274.00 | 224 | 1.22321428571429 |
| 11/18/2014 | 98126 | Youtube | $542.00 | 463 | 1.17062634989201 |

**Figure X. SQL query result for contractor 8**

*2. R*

Export this query result to R and use a non-linear least squares regression to fit the ROI vs. time graph to a logarithmic curve of the form $a+b*log(x)$ where $a$ and $b$ are constants to be determined. We chose a logarithmic curve by assuming a contractor will increase his/her ROI over time as they gain experience but at a decreasing rate to model a learning curve. See Figure 4.1. We

can take this query further by using the curve to predict the contractor's ROI at a time in the future and even compare ROIs for multiple contractors who do the same task to find the best one to hire for the next project through estimating their future ROIs. See Figure 4.2 for an example.

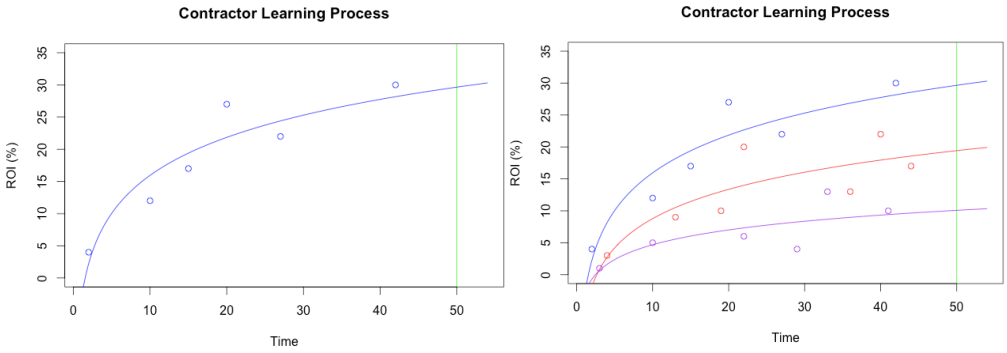



**Figure 4.1 and 4.2 Plot of ROI vs. Time for multiple contractors of the same type with a prediction line (at a future time in green)**

**Query 5: Project Promotion Optimization**

*Justification*

This query can help Madera group decide between a set of potential promotional methods through analyzing how similar methods in the past have affected the returns of each event in terms of its interactions with other events held within the same project and area of interest. This analysis is made into a suggestion of the best set of events to hold given a budget and target fundraising amount.

*Implementation*

*1. SQL*

In Access, allow the user to enter a set of parameters corresponding to his/her target fundraising amount, the area(s) of interest of the project, the type of each proposed promotional method, and the estimated budget of each proposed method. Then let SQL pull out all costs, returns, ROIs, and types of past data from projects similar to the one proposed as below.

In this example, we suppose the proposed project is environmental and has the possibilities of holding a fundraising dinner, creating a print advertisement, and making a Youtube video:

***SQL Code***

```
SELECT e.Type, w.IsForEvent as IsFor, e.EstTotalAmountRaised as Return,
e.Cost, -1*( e.EstTotalAmountRaised+e.Cost)/e.Cost as ROI
FROM Withdrawal as w, Event as e, ProjectIsOfAOI as pa, Transaction as t
WHERE pa.Area = 'Environmental' AND w.TransactionNumber = t.TransactionNumber AND
t.ProjectID = pa.ProjectID AND e.ProjectID = pa.ProjectID AND e.EventID = w.IsForEvent
UNION
SELECT adplat.Type, w.IsForAd as IsFor, a.EstTotalAmountRaised as Return, a.Cost, -
1*(a.EstTotalAmountRaised+a.Cost)/a.Cost as ROI
FROM Withdrawal as w, AdIsPutOnAdplatform as ad, Advertisement as a, ProjectIsOfAOI as
pa, Transaction as t, AdPlatform as adplat
WHERE pa.Area = 'Environmental' AND w.TransactionNumber = t.TransactionNumber AND
t.ProjectID = pa.ProjectID AND a.AdID = ad.AdID AND adplat.AdPlatformID = ad.AdPlatformID
AND a.ProjectID = t.ProjectID AND w.IsForAd = a.AdID;
```

Query5

| Type | IsFor | Return | Cost | ROI |
|---|---|---|---|---|
| Fund. Dinner | 40286 | $2,583.00 | ($1,487.00) | 0.737054472091459 |
| Fund. Dinner | 40294 | $874.00 | ($365.00) | 1.39452054794521 |
| Fund. Dinner | 40582 | $883.00 | ($562.00) | 0.571174377224199 |
| Fund. Dinner | 40672 | $2,957.00 | ($1,683.00) | 0.756981580510992 |
| Print | 20124 | $487.00 | ($376.00) | 0.295212765957447 |
| Print | 20573 | $975.00 | ($538.00) | 0.812267657992565 |
| Print | 20848 | $647.00 | ($413.00) | 0.566585956416465 |
| Print | 20987 | $1,047.00 | ($529.00) | 0.979206049149338 |
| Youtube | 20036 | $812.00 | ($515.00) | 0.576699029126214 |
| Youtube | 20049 | $1,593.00 | ($982.00) | 0.622199592668024 |
| Youtube | 20117 | $958.00 | ($627.00) | 0.527910685805423 |
| Youtube | 20135 | $1,355.00 | ($854.00) | 0.586651053864169 |

**Figure 5.1 SQL Code Result in Access**

*2. R*

From Figure 5.1, we see that there were four similar projects that have a combination of these three promotional methods. This data and the user input is converted to a CSV file that is input into R where we find the average ROI for each promotional method and calculate the covariance matrix. A matrix structure is created to calculate the covariance for each set of two promotional methods. We then weight each entry of the covariance matrix by:
$(Percent\ of\ budget\ for\ i)x(Precent\ of\ budget\ for\ j) * Cov(i,j)$, where $i$ and $j$ are indexes for the two methods in the covariance matrix. This will later be the factors of the decision variables in the linear program. Lastly, we calculate the ROI (indicated as rho in the R code) for the proposed fundraising amount and budget. See Figure 5.2.

```
m = read.csv("query5_1.csv")
t = read.csv("query5_2.csv")
# get expected rate of return
avg = -1
names = "null" # vector of all the different types of events/ads
numproj = -1
i=1
x=1
while(i<=nrow(m)){
  proj = which(m[,2]==m[i,2])
  name = m[proj[1],2]
  names[x] = as.character(name)
  avg[x] = sum(m[proj,6])/length(proj)
  numproj[x] = length(proj)
  i=i+length(proj)
  x=x+1
}

# find the covariance matrix
covar = matrix(rep(0,length(numproj)^2), nrow = length(numproj))
colnames(covar) = names
rownames(covar) = names
k=1
expror=list()#list(rep(-1, numproj[1]))
for(j in 1:length(numproj)){
  expror[[j]]=m[k:(k+numproj[j]-1),6]
  k=k+numproj[j]
}
```

```
for(m in 1:length(numproj)){
  p = length(numproj)
  for(n in 1:p){
    covar[m,n] = cov(expror[[m]],expror[[n]])
  }
  p = p-1
}

#create constants of terms in objective function
index = -1
for(r in 1:length(names)){
  index[r] = which(t[,1] == names[r])
}

tb = which(t[,1]=="total budget")
percentcost = t[1:tb-1,2]/t[tb,2]
percentcost = percentcost[index]

for(m in 1:length(numproj)){
  p = length(numproj)
  for(n in 1:p){
    covar[m,n] = covar[m,n]*percentcost[m]*percentcost[n]
  }
  p = p-1
}
rho = (t[tb+1,2]-t[tb,2])/t[tb,2]
entry = list(avg,covar,percentcost,rho)
```

```
> entry
[[1]]
[1] 0.5783651 0.8649327 0.6633181

[[2]]
                 Youtube Fund. Dinner    Print Ad
Youtube      0.0005583259  0.002140676 0.001088039
Fund. Dinner 0.0021406755  0.010538739 0.004163786
Print Ad     0.0010880390  0.004163786 0.011979680

[[3]]
[1] 0.6075 0.2830 0.3670

[[4]]
[1] 0.9
```

**Figure 5.2. R code and output (stored as a list called "entry")**

*3. AMPL with CPLEX*

Using the outputs of R a linear program can be written to find the optimal set of events. We use the following general linear program:

Let $y_i$ be 1 if advertisement/event i is taken, 0 otherwise;
$R_i$ = expected rate of return for each event, calculated based on the past data for events of same type and area of interest [(return – cost)/cost]
ρ = (target fundraising amount)/budget
$B_i$= event's cost over budget

Min $\sum_j \sum_i B_i * B_j * COV(R_i * y_i, R_j * y_i)$
St $\sum_i R_i * y_i \geq \rho$
$\sum_i B_i * y_i \leq 1$
$y_i \in \{0,1\}$

See Figure 5.3 for the AMPL code and output.

```
var y1 binary;
var y2 binary;
var y3 binary;
var x12 binary;
var x13 binary;
var x23 binary;

minimize risk:
0.00055833*y1+0.01053874*y2+0.01197968*y3+0.00214068*x12*2+0.00108804*x13*2+0.
00416379*x23*2;

subject to
goal: 0.57836509*y1+0.86493274*y2+0.66331811*y3>=0.9;
budget: 0.6075*y1+0.283*y2+0.367*y3<=1;
do12: y1+y2-1<=x12;
do13: y1+y3-1<=x13;
do23: y2+y3-1<=x23;
```

```
ampl: reset;
ampl: option solver '/Users/yikun/Downloads/ampl/cplex';
ampl: model '/Users/yikun/Downloads/ampl/QUERY5.mod';
ampl: solve;
CPLEX 12.6.0.0: optimal integer solution; objective 0.01471409
3 MIP simplex iterations
0 branch-and-bound nodes
ampl: display _varname, _var;
: _varname _var    :=
1   y1      1
2   y2      0
3   y3      1
4   x12     0
5   x13     1
6   x23     0
;
```

| Event | Variable | Chosen? |
|---|---|---|
| Youtube | y1 | Yes |
| Fund. Dinner | y2 | No |
| Print Ad | y3 | Yes |

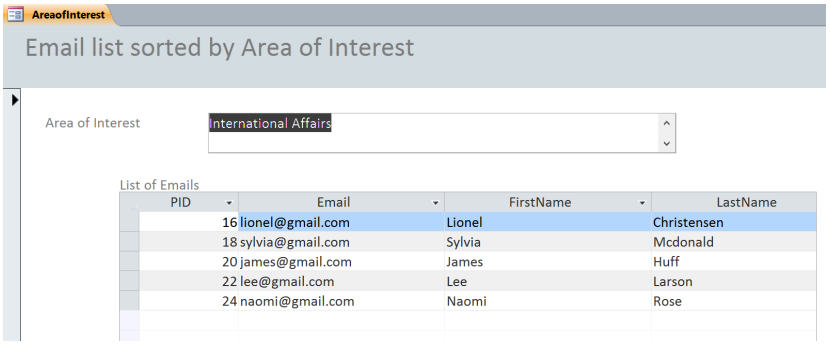
**Figure 5.3 AMPL code, results, and summary table of implementation in AMPL with CPLEX**

Thus, we see that for our example the best solution is to make a Youtube video and a Print ad. Due to budget constraints and risk, a fundraising dinner would not be feasible.

## Forms and Reports

A sample of the forms and reports that are possible with this database are shown below.

*1. Form 1: Emailing List*




This form shows an email list sorted by Area of Interest through use of a subform. Our client can easily access the email list according to different areas of interest so they can promote fundraising events and encourage donations for their targeted audience.

InternInfo

## Intern Information

PersonID: 6
IsWorking: ☑
AvailableHourPerWeek: 11
FirstName: Jay
LastName: Low
CellPhone:
SSN: 0
Wage: $12.00

Intern Skills

| Skill |
|---|
| Data Analyzing |
| Finance |
| Graphic Design |
| Microsoft Office |
| Writing |

Record: 1 of 5

This form includes information on each intern individually with their corresponding skills and available hours so the project manager can assign work to different interns accordingly. This form can further be used to add interns to the Person, Employee, and Interns tables at the same time to facilitate the continuity and consistency of the database.

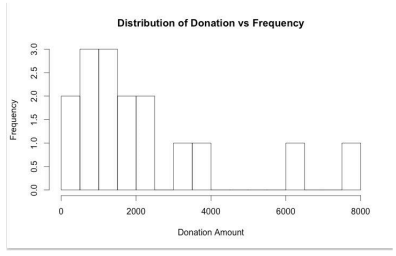
project_99123

## Report on Project 99123

Wednesday, December 10, 2014
5:23:38 PM

ProjectID: 99123
StartDate: 10/8/2014
EndDate: 11/12/2014
TotalAmountFundraised: $39,500.00

| Type | Amount Fundraised | Cost | ROI |
|---|---|---|---|
| Fund. Dinner | $10,500.00 | ($3,000.00) | 2.5 |
| Print | $400.00 | ($100.00) | 3 |
| Youtube | $600.00 | ($200.00) | 2 |
| | $11,500.00 | | |




This report provides a one-glimpse summary of a finished project. It includes details such as, project start and end date, the amount fundraised and the ROI of each event. In addition, the information on the donation frequency and amount is extracted to R and a graph is generated. This could be something that can be added to the Corporate Portfolio entity for Madera to show future clients.

octemployee

## Employee Monthly Report Oct 2014

| EmployeePID | Type | FirstName | LastName | For Project | Total Hours Worked |
|---|---|---|---|---|---|
| 16 | Contractor | Lionel | Christensen | | |
| | | | | 99123 | 8 |
| | | | | 97236 | 3 |
| | | | | 92818 | 2 |
| 17 | Intern | Sonia | Campbell | | |
| | | | | 93821 | 6 |
| 18 | Intern | Sylvia | Mcdonald | | |
| | | | | 96382 | 7 |
| | | | | 95674 | 4 |
| 19 | Contractor | Ester | Gardner | | |
| | | | | 99123 | 5 |
| 20 | Contractor | James | Huff | | |
| | | | | 97326 | 4 |

Wednesday, December 10, 2014 Page 1 of 1

This report summarizes monthly employee performance. It contains each employee who worked during the month including the type of employee, which projects they have been working on, and how many hours they have worked on each project.

**Normalization Analysis**

Next we present samples of different normalized forms used in our database and our reasoning for choosing to keep them at their respective stages.

*1NF: Proposal*

***Proposal*** *(ProjectID, ProposalID, IsPresentedTo, ProposalAttachment, Description, ProposalType)*

***Functional Dependencies:***

*{ProjectID, ProposalID} → {ProposalAttachment, Description, ProposalType};* *(1)*

*{ProjectID} → {IsPresentedTo}* *(2)*

*{ProposalAttachment} → {ProjectI, ProposalID D}* *(3)*

In the finalized database, Proposal is in 1NF. It is a weak entity with primary keys ProjectID and ProposalID, which is a partial key unique to the project only. It is in 1NF because no attributes are multi-valued, but it violates 2NF because of (2) since a *IsPresentedTo* is partially dependent on the primary key. It is possible to further normalize the relation as shown below until BCNF but not necessary to do so. To normalize Proposal to 2NF, remove partial dependencies. We note that this actually generates 3NF as well since there were no transitive dependencies in the original relation:

***Proposal****(ProjectID, ProposalID,ProposalAttachment, ProposalType)*

***PresentTo****(ProjectID, IsPresentedTo)*

To normalize to BCNF, all determinants of FDs must be superkeys, which applies to (3) since *ProposalAttachment* is not in the primary key:

***Proposal****(ProjectID, ProposalID, Description, ProposalType)*

***AttachmentForProject****(ProposalAttachment, ProjectID)*

***AttachmentForProposal****(ProposalAttachment, ProposalID)*

***PresentTo*** *(ProjectID, IsPresentedTo)*

Our client would like to save all of the company's proposals and to have easy access to them. Normalization will increase the difficulty for our client to look for specific proposal documents as well as maintaining consistency throughout the database. The current design in 1NF sorts proposals by different projects with all attributes for the proposals, which meets our client's needs well. Therefore, to further break down Proposal is redundant and meaningless.

*2NF:Donation*

Donation and Person are two very important relations stored as 2NF in the database. For the purpose of query implementation both relations are not fully normalized. First we look at Donation:

***Donation*** *(DonationID, ProjectID, Amount, Source, ConfirmationNumber, SpecialRequest, PID, DepositedInto, DonationDate, Time, ReasonOfDonation)*

***Functional Dependencies:***

*{DonationID} → {ProjectID, Amount, Source, ConfirmationNumber, SpecialRequest, PID, DepositedInto, DonationDate, Time, ReasonOfDonation}* *(1)*

*{DonationID} → {ConfirmationNumber, Source}* *(2)*

*{ConfirmationNumber, Source} → {Amount, SpecialRequest, PID, DonationDate, Time, ReasonOfDonation}* *(3)*

In the Donation table, DonationID is a unique number assigned to each donation while ConfirmationNumber is a unique number generated by each fundraising platform (e.g. Indiegogo) and Source is a unique attribute representing each fundraising platform. The candidate key to this relation could be either {DonationID} or {ConfirmationNumber, Source}.

To normalize this relation into 3NF, we need to remove the transitive dependency in (1) since both *{DonationID}* and *{ConfirmationNumber, Source}* can determine the right hand side of (1). A possibility is suggested below:

***Donation*** *(DonationID, ConfirmationNumber, Source, ProjectID, DepositedInto)*

***ConfirmationNumberInfo*** *(ConfirmationNumber, Source, Amount, SpecialRequest, PID, DonationDate, Time, ReasonOfDonation)*

This actually also gives BCNF since *{ConfirmationNumber, Source}* is also a super key so it is okay to leave the Donation relation as is.

Donation is kept in 2NF because it is much easier to extract donation information from only one table for the implementation of query 1 and 3. Also it causes unnecessary duplication to further normalize the relation into 3NF or BCNF since a table is already generated in access to declare the relationship between each donation and the fundraising platform it comes from. The table is named DonationDonatedtoFundraisingPlatform (DonationID, FID) and is created due to the many-to-many relationship between donation and fundraising platform.

*2NF: Person*

***Person*** *(PID, FirstName, MI, LastName, HomePhone, CellPhone, OfficePhone, FaxNo, StreetNo, Street, Apt, City, Country, ZIP)*

***Functional Dependencies:***

*{PID} → {FirstName, MI, LastName, HomePhone, CellPhone, OfficePhone, FaxNo, StreetNo, Street, Apt, City, Country, ZIP}* *(1)*

*{ZIP} → {Country, City}*[1] *(2)*

To normalize to 3NF and BCNF, transitive dependencies are removed:

***Person****(PID, FirstName, MI, LastName, HomePhone, CellPhone, OfficePhone, FaxNo, StreetNo, Street, Apt, ZIP)*

***Address****(ZIP, City, Country)*

Person is also kept in 2NF in the finalized database to keep complete profiles of all people involved with Madera Group in one table. It is also easier to extract data for queries 1,2, 3 and 5. It is meaningless to break up an address just so there are no transitive dependencies.

*3NF:EmpWorksOnPrj*

***EmpWorksOnPrj*** *(EmployeePID, ProjectID, StartDate, EndDate, ContactEmail)*

***Functional Dependencies:***

*{EmployeePID, ProjectID} → { StartDate, EndDate, ContactEmail }* *(1)*

*{ContactEmail} → {EmployeePID}* *(2)*

ContactEmail is a unique email address determined by EmployeePID and ProjectID. For each project, each employee will specify a primary email address as ContactEmail mainly for communication purposes since in general a person can have multiple email addresses.

The relation is in 3NF as no multi-valued attributes, partial dependencies, or transitive dependencies exist. However the relation is not in BCNF because of the functional dependency between ContactEmail and EmployeePID. To further bring EmpWorksOnPrj into BCNF we need to remove the non-prime attribute ContactEmail from the original relation and store this information in another relation. As a result, we get the following:

***EmpWorksOnPrj*** *(EmployeePID, ProjectID, StartDate, EndDate)*

***EmpContactInfo*** *(ContactEmail, ProjectID)*

In the finalized database EmpWorksOnPrj is kept in 3NF because it is redundant to create a new table for ContactEmail since a relation for the multi-valued attribute email already exists. At the

[1] CellPhone cannot be prime attribute since values of this attribute can be empty to some persons. We are also making the assumption that a person can have at most one registered cellphone number.

same time it is inconvenient for our client to find all emails of employees involved in a project if *EmployeePID* and *ContactEmail* are in different tables.

*BCNF:Advertisement*

***Advertisement (****AdID, AdvertisementName, NumofPrint, NumofViews, NumOfLikeShare, PersonInCharge, ProjectID, AuthorizedBy, Cost, EstTotalAmountRaised)*

***Functional Dependencies:***

*{AdID} → {AdvertisementName, NumofPrint, NumofViews, NumOfLikeShare, PersonInCharge, ProjectID, AuthorizedBy, Cost, EstTotalAmountRaised}*

The relation is in 1NF because all attributes are single-valued (i.e. each AdID can only be assigned to one Advertisement, and all other attributes of a specific advertisement are singular). *Advertisement* is also in 2NF as it is the only primary attribute so no partial dependency exists. We assume *AdvertisementName* doesn't need to be unique within a project so *AdvertisementName ProjectID*,} does not determine *AdID;* thus, *Advertisement* is in 3NF since no transitive dependencies exist. Lastly, Advertisement is in BCNF because no non-prime attributes (or sets of them) can determine *AdID*.

## Discussion and Future Work

Given the time constraints, we were unfortunately unable to implement certain aspects that would enhance the user interface and experience of this database. Our client, Kath Delaney (founder of Madera) has expressed a great deal of enthusiasm to apply and use our database for her company. This vested interest requires communication on our part to leave a well-documented set of instructions on our database design for her to maintain and possibly further develop this database. This includes steps to run programs such as AMPL/CPLEX and R, which are required for the final query results. Ideally, queries would run with a single click of a button. As such, our future work on queries would include combining steps more seamlessly.

A great deal of effort has been devoted for the tailored specifications of Access' tables. At this time, Madera Group employees are not familiar with the use of this Microsoft program and adding information into the database will take explaining. Consequently, we'd like to create more forms for easier and more efficient data entry.

A design issue that arose was the lack of an entity for all promotional methods. In separating out advertisements and events we wanted to store more specific and detailed data on the differences between them. But the queries that required an aggregate data column over all promotional methods, for example Query 3, were difficult to implement. This was evident in the length and complexity of some of our SQL codes, which would have been more efficient had such an entity been used.We note that the Event and Advertisement tables had many attributes in common.

Lastly, possible query improvements for future work include importing a visual map on Google Maps platform for Query 1 and an assignment allocation strategy that involves assignments requiring multiple skills. Because of a limitation in the linear programming model, we have to assume that each assignment only requires one skill, which is not realistic and applicable. Thus a more complicated linear programming model is needed in order to incorporate multi-skill assignments.

Query 1 entails determining which donors to visit on one business trip while maximizing expected donations. Since this is a complicated multi-step problem, it would greatly improve user experience by simplifying the output to an interface a majority of people are familiar with-- Google Maps. Furthermore, Google Maps includes a feature called map layering which can be saved per google account. We'd like to explore this option further because the trip output business plan can be easily accessible and shared by multiple users while they are mobile.