IEOR 115 DATABASE DESIGN PROJECT FINAL REPORT Teens in Public Service (Team 8)

Members: Tuana Arin, Elias Castro-Hernandez, Michael Chung, Neel Davar, Amanda Wu, Janet Xu, Edward Yang, Talya Zalipsky

Date: 11 December 2018

1. CLIENT DESCRIPTION:

Teens in Public Service (TIPS) is a not-for-profit based in Seattle, WA, dedicated to developing future leaders committed to their communities. Every summer, TIPS selects a cohort of outstanding high school youth leaders and connects them with paid summer internship opportunities within local nonprofits. TIPS interns help nonprofits carry out organizational initiatives by providing invaluable support to their summer programs. In turn, interns learn how to work in the nonprofit community while also building leadership skills to meet the needs of their organization. TIPS works with a variety of organizational partners in areas of social rights activism, arts and sciences, community health, youth education, and homelessness. TIPS raises funds for their efforts through events, direct donations, and grants.

TIPS can benefit from a well-designed relational database that helps them improve operations and fundraising efforts. For example, TIPS could use the database in selecting and evaluating interns and in managing internships. Similarly, keeping searchable records of alumni and other individuals according to various criteria would speed up the intern selection process, would facilitate the development of mentors, and would improve fundraising efforts by means of the insights gained on donor behavior and relationships.

1.1 PREVIOUS APPROACH:

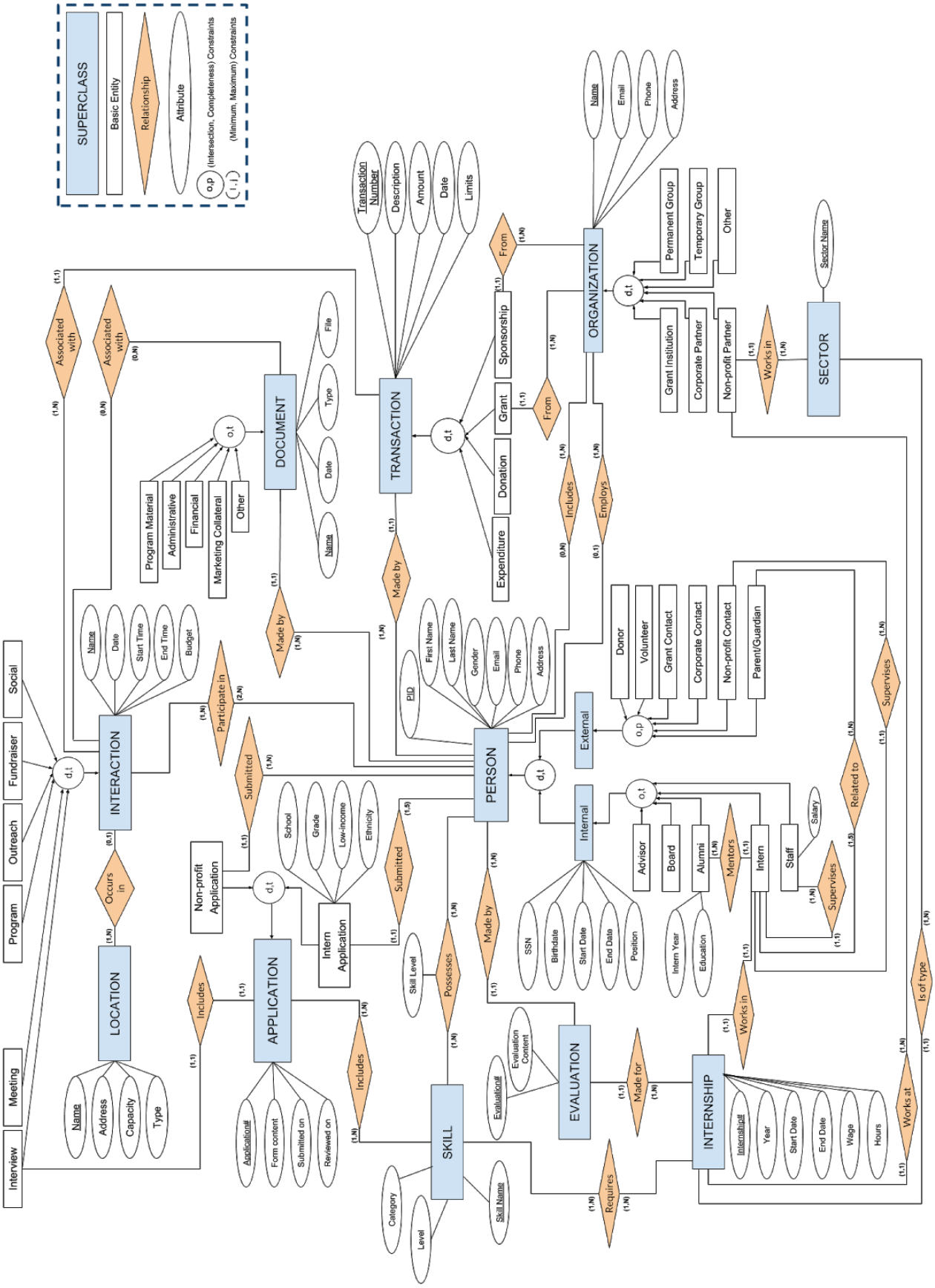
To date, TIPS has been using a number of enterprise platform products to manage their data, including Salesforce, EventBrite, QuickBooks, ClickTime, DropBox, and Google Drive. They have no other formal database support other than these applications, and are facing significant issues with data fragmentation and upkeep. Much of the existing data they have is recorded in Excel spreadsheets or in Google Sheets.

1.2 GOALS AND BENEFITS:

For the team: understanding how a fast-growing non-profit can improve its processes for managing, accessing, and analyzing its internal data. The team can learn about and apply technical database concepts via this hands-on project outside of the classroom, exploring and creating a relational database prototype that may serve as a model for a commercial project that TIPS may decide to undertake to revamp its database systems. The team will also be able to develop and practice skills such as client relationship management, user research, and project management.

For the client: TIPS is an organization that stands to benefit significantly from stronger and more effective data management practices than those that it has used before -- especially as they transition to a new executive director and seek to expand their program. Short of being able to use a functional database prototype produced by the team (maintenance and continued service will not be provided), TIPS will at least be able to learn about its current data issues in depth and how it might benefit from an intentionally designed, well-engineered relational database.

2. SIMPLIFIED ENHANCED ENTITY-RELATIONSHIP (EER) DIAGRAM:



3.1 RELATIONAL DESIGN SCHEMA: ENTITY AND SUPERCLASS/SUBCLASS RELATIONS

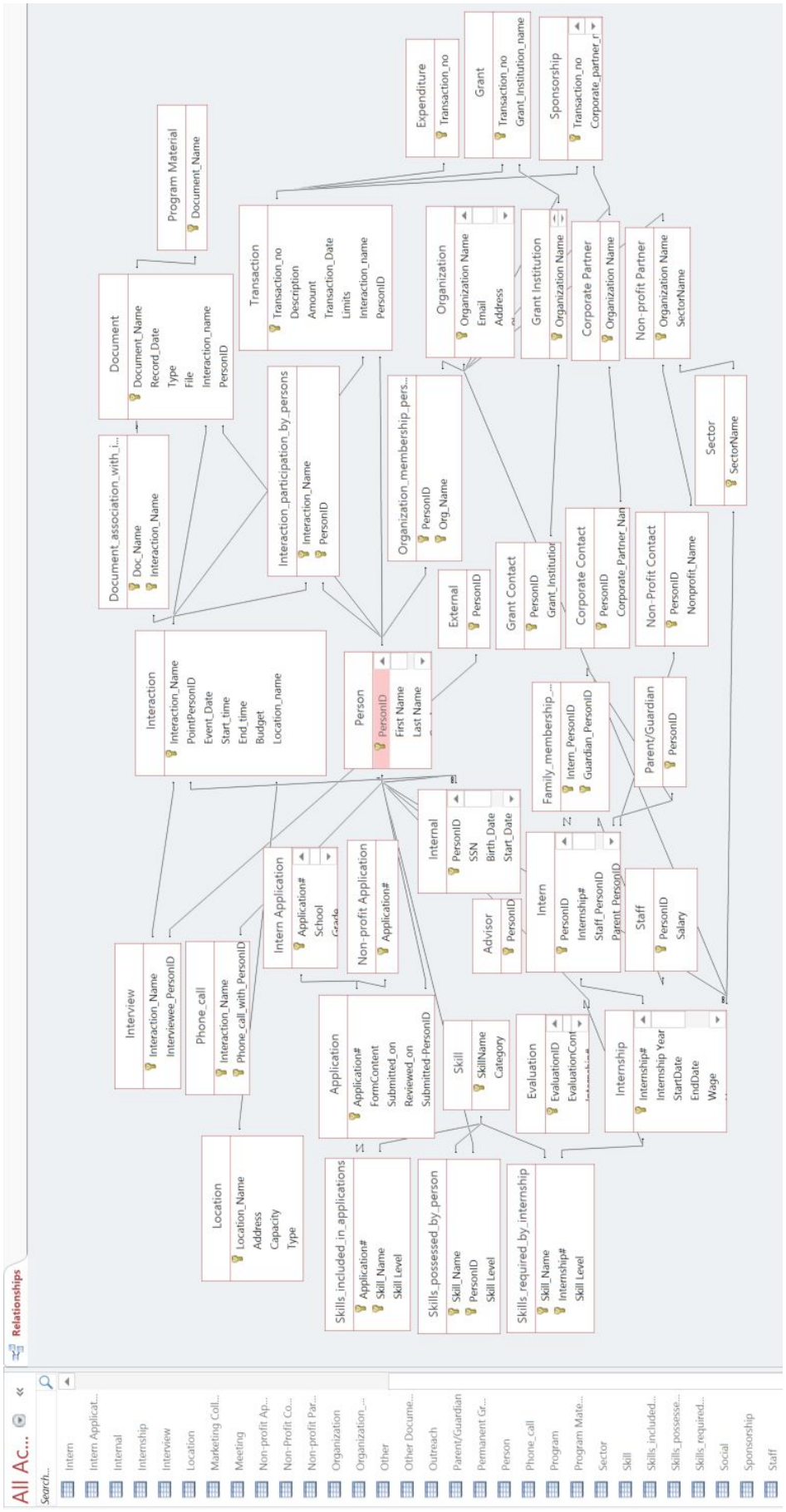
#	**Entity_name**(Principal_Key(s), Attribute(s), *Foreign_key(s)*)		
1	**Person**(PID, First Name, Last Name, Gender, Email, Phone, Address, *OrgName*[4])		
	Relation(*Foreign_key*, Partial_key, Attributes)		
1a	**Internal**(PID[1], SSN, Birth_Date, Start_Date, End_Date, Position)	**1b**	**External**(PID[1])
1a.i	**Advisor**(PID[1])	**1b.i**	**Donor**(PID[1])
1a.ii	**Board**(PID[1])	**1b.ii**	**Volunteer**(PID[1])
1a.iii	**Alumni**(PID[1], Intern_Year, Education)	**1b.iii**	**Parent/Guardian**(PID[1])
1a.iv	**Intern**(PID[1], Staff_PID[1a.v], Internship#[2], NonProfitContact_PID[1b.iv])	**1b.iv**	**Non-profit Contact**(PID[1])
1a.v	**Staff**(PID[1], Salary)	**1b.v**	**Corporate Contact**(PID[1])
-		**1b.vi**	**Grant Contact**(PID[1])
2	**Internship**(Internship#, Year, StartDate, EndDate, Wage, Hours. *SectorName*[5], *OrgName*[4], *Intern_PID*[1a.iv])		
3	**Evaluation**(Evaluation#, Evaluation_Content, *Internship#*[2], *Made_by_PID*[1])		
4	**Organization**(Name, phone, email, address)		
	Relation(*Foreign_key*, Partial_key, Attributes)		
4a	**Non-profit Partner**(Name[4], *SectorName*[5])	**4d**	**Permanent Group**(Name[4])
4b	**Corporate Partner**(Name[4])	**4e**	**Temporary Group**(Name[4])
4c	**Grant Institution**(Name[4])	**4f**	**Other**(Name[4])
5	**Sector**(SectorName)		
6	**Skill**(SkillName, Category, Level)		

#	**Entity_name**(Principal_Key(s), Attribute(s), *Foreign_key(s)*)		
7	**Application**(Application#, FormContent, Submitted_on, Reviewed_on,*Submitted_PID*[1], *InterviewName*[8])		
	Relation(*Foreign_key*, Partial_key, Attributes)		
7a	**Intern Application**(Application#[7], School, Grade, Low_income, Ethnicity)	**7b**	**Non-profit Application**(Application#[7])
8	**Interaction**(Name, Date, Start_time, End_time, Budget, Location_name[9], Document_name[10])		
	Relation(*Foreign_key*, Partial_key, Attributes)		
8a	**Interview**(Name[8], Interviewee_PID[1])	**8e**	**Outreach**(Name[8])
8b	**Meeting**(Name[8], Meeting_with_PID[1])	**8f**	**Fundraiser**(Name[8])
8c	**Program**(Name[8])	**8g**	**Social**(Name[8])
9	**Location**(Name, Address, Capacity, type, *Interaction_name*[8])		
10	**Document**(Name, Date, Type, File, Interaction_name[8], Made_by_PID[1])		
	Relation(*Foreign_key*, Partial_key, Attributes)		
10a	**Program Material**(Name[10])	**10d**	**Marketing Collateral**(Name[10])
10b	**Administrative**(Name[10])	**10e**	**Other Document**(Name[10])
10c	**Financial**(Name[10])	**-**	
11	**Transaction**(Transaction_no, Description, Amount, Date, Limits, Interaction_name[8], Made_by_PID[1])		
	Relation(*Foreign_key*, Partial_key, Attributes)		
11a	**Expenditure**(Transaction_no[11], Expenditure_name)	**11c**	**Grant**(Transaction_no[11], Grant_name, fghcryud[p`-organization_name[4])
11b	**Donation**(Transaction_no[11])	**11d**	**Sponsorship**(Transaction_no[11], Sponsorship_name, organization_name[4])

3.2 RELATIONAL DESIGN SCHEMA: MANY-TO-MANY RELATIONS

#	**Relation**(Principal_Key(s), Attribute(s), *Foreign_key(s)*)
12	**Organization_membership_persons**(PersonID[1], Org_Name[4])
13	**Family_membership_guardian_of_persons**(PersonID[1a.iv], PersonID[1b.iii])
14	**Skills_possessed_by_person**(Skill_Name[6], PersonID[1])
15	**Skills_required_by_internship**(Skill_Name[6], Internship#[2])
16	**Skills_included_in_applications**(Application#[7], Skill_Name[6])
17	**Interaction_participation_by_persons**(Interaction_Name[8], PersonID[1])
18	**Document_association_with_interactions**(Doc_Name[10], Interaction_Name[8])

4. RELATIONAL DESIGN IMPLEMENTED IN MS ACCESS RELATIONSHIP VIEW



Note: Classes are loosely organized so as to reflect the spatial arrangement of database objects in the EER diagram (Page 2)..

5. DATABASE QUERY ANALYSES:

For the extended data analysis on SQL extractions (queries 1 and 3), we used Jupyter Notebook software suite and programming mostly in Python. Moreover, the following external libraries (packages) were utilized:

External Software: (1) pandas, (2) sklearn, (3) seaborn, (4) NumPy

****1. ANALYZING INTERN CHURN**

Tables Used: Internship, Intern Application, Person

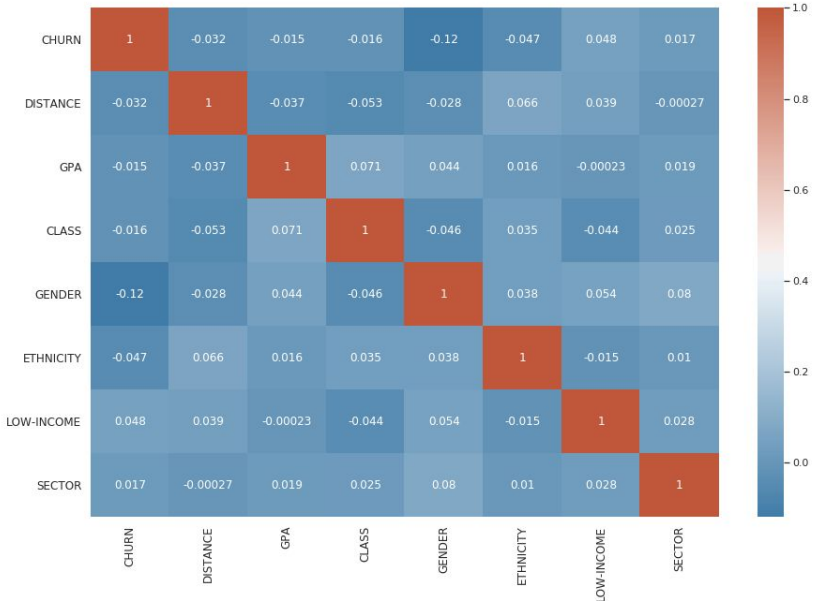
Description	Query 1 SQL
SQL: Find all internships where End_date is before (less than) the modal end_date. Return data from application and person tables for those internships, joining on the same personID.	`SELECT i.orgName, i.sector, i.year, i.internID, ia.formcontent, p.gender, p.bdate, p.schoolname, p.address FROM internship AS i, intern_application AS ia, person AS p WHERE i.internID = ia.submitted_personid, i.internID = p.personID, i.end_date < (SELECT TOP 1 i.end_date AS modal_end_date FROM internship i GROUP BY i.end_date ORDER BY COUNT(*) DESC);`

ANALYSIS: in an effort to reduce churn, we will take time stamped results from the above query and train both a **logistic regression classifier** and a **random forest classifier** model(s) to identify the intern candidates least likely to complete their internships; these "failures" cause a significant issue for the client's relationship with partner non-profit organizations who don't have the benefit of working with a consistent and reliable intern.

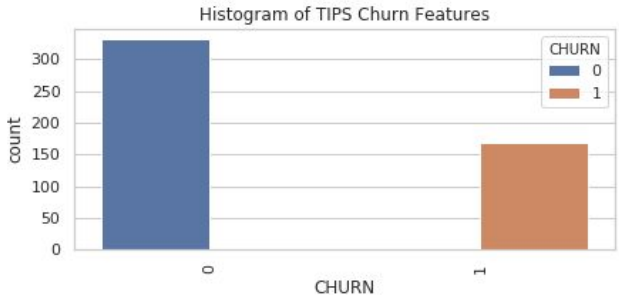
Summary statistics on intern churn dataset, using a realistic seeding sample informed by client:

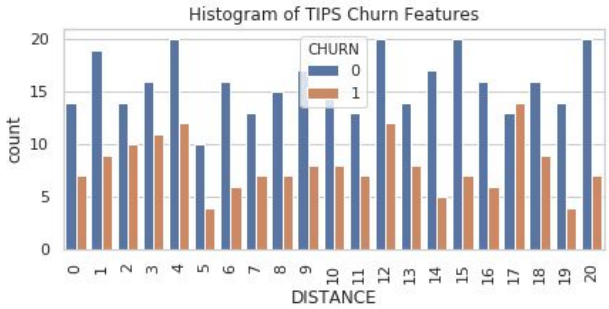
	CHURN	DISTANCE	GPA	CLASS	GENDER	ETHNICITY	LOW-INCOME	SECTOR
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	0.336000	10.008000	2.789800	1.484000	0.322000	2.000000	0.384000	4.032000
std	0.472812	6.108565	0.734926	1.111853	0.467711	1.356348	0.486845	2.607792
min	0.000000	0.000000	1.500000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	4.000000	2.200000	1.000000	0.000000	1.000000	0.000000	2.000000
50%	0.000000	10.000000	2.800000	1.000000	0.000000	2.000000	0.000000	4.000000
75%	1.000000	15.000000	3.400000	2.000000	1.000000	3.000000	1.000000	6.000000
max	1.000000	20.000000	4.100000	3.000000	1.000000	4.000000	1.000000	8.000000

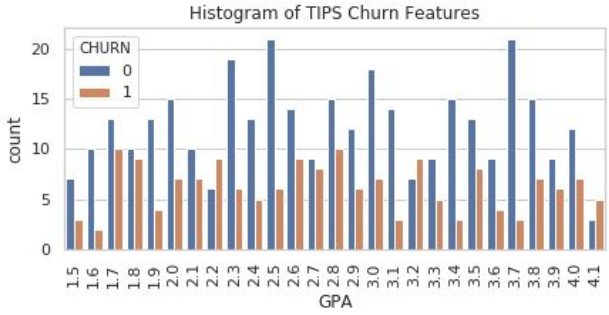
Feature correlation matrix allows us to analyze whether any specific features have more influence:

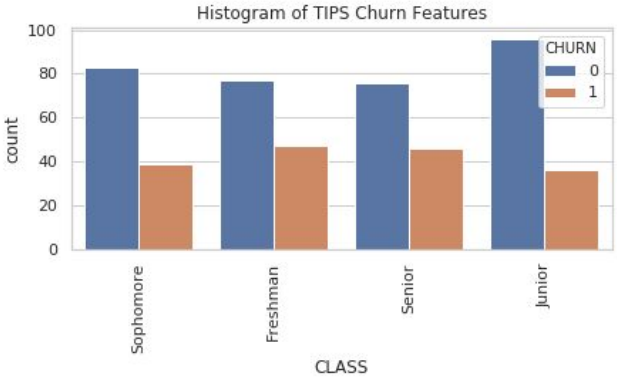


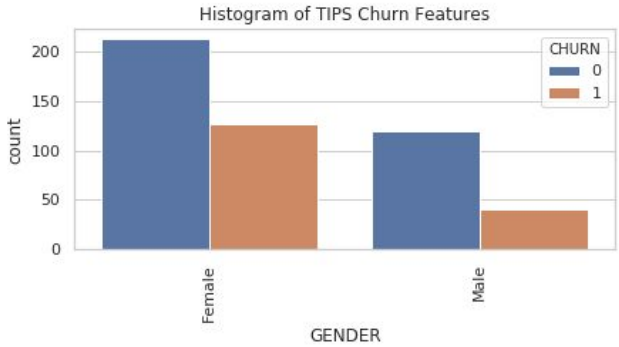
Visualizations of analysis output in terms of histograms comparing churn rates across various features:

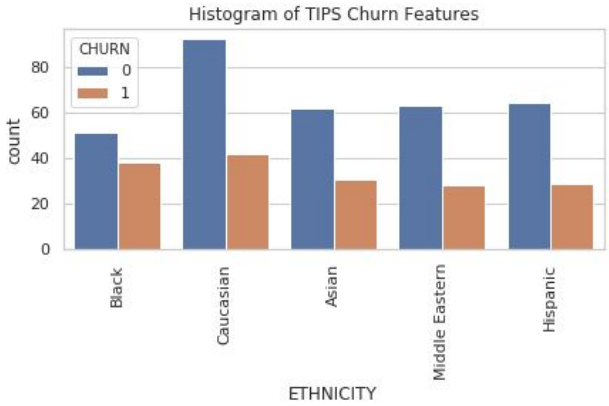


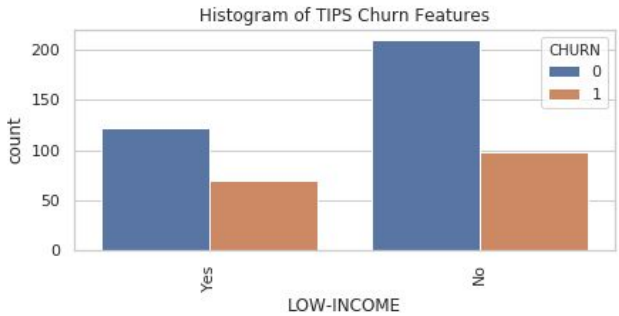


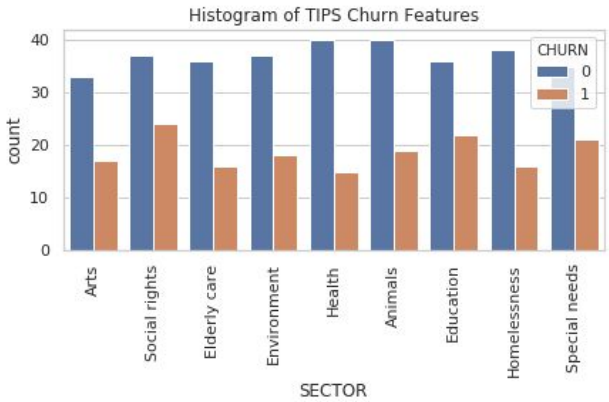






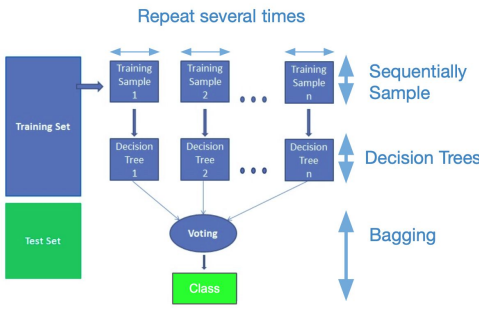


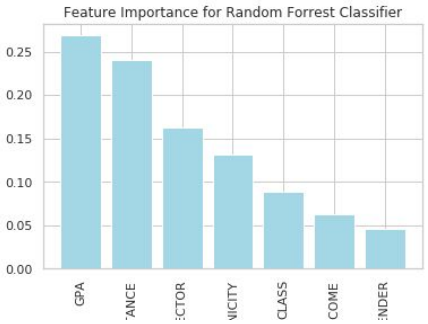


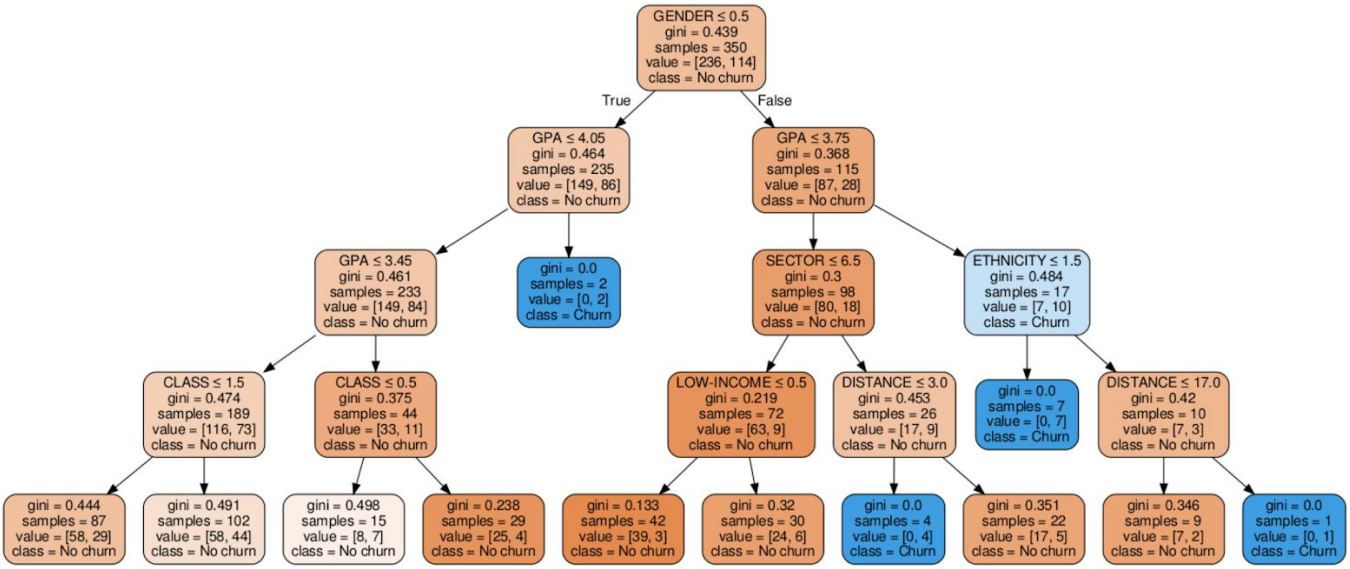


Testing churn predictions of different models showed that **random forest classifier** was the most accurate.

Model	Accuracy
Naive Baseline	34%
Random Forest Classifier	**65%**
Logistic Regression Classifier	64%
Decision Tree Classifier	61%
Gradient Boosting Classifier	60 %







To compute Gini impurity for a set of items with J classes, suppose $i \in \{1, 2, \ldots, J\}$, and let p_i be the fraction of items labeled with class i in the set.

$$\mathrm{I}_G(p) = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k = \sum_{i=1}^{J} p_i(1-p_i) = \sum_{i=1}^{J} (p_i - {p_i}^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} {p_i}^2 = 1 - \sum_{i=1}^{J} {p_i}^2$$

Visualization of **decision branching**: model differentiates between churn associated with various classes or attributes, accounting for different conditional probabilities of churn via Gini impurities (equation at bottom).

2. WEIGHTED CRITERIA FOR CANDIDATE PRE-SCREENING

Tables Used: Intern Application, Non-profit partners, Skills Included in Application

Description	Query 2 SQL

SQL: Collect all demographic and skill-related information from intern applications and from non-profits' specifications for their internships.	`SELECT sa.skill_name, sa.skill_level, ia.formcontent, p.address`
	`FROM skills_included_in_application sa, intern_application ia, person p`
	`WHERE p.personID = ia.submitted_personid, sa.application_no = ia.application_no`
	`GROUP BY p.personID;`
	
	`SELECT si.skill_name, si.skill_level, np.formcontent, np.address`
	`FROM skills_required_by_internship si, nonprofit np`
	`WHERE si.internship_no = np.internship_no`
	`GROUP BY np.orgName`

ANALYSIS: In order to expedite the intern selection process and minimize the number of "bad" assignments, where interns either don't possess the right skills or characteristics for their internships or live very far away from their placements, we intend to use the above query to : (1) extract differently weighted features of non-profits' specifications and (2) candidates' applications (for example location, skills, age) as vectors, we will then (3) use a **K-nearest neighbors approach** to find a shortlist of candidates who would be the most appropriate fit for each internship opening.

**3. PREDICTIVE MODEL FOR DONOR CONTRIBUTIONS	
Tables Used: Interaction, Fundraiser, Interaction_participation_by_persons, Donor	
Description	**Query 3 SQL**

SQL: Select all data for donations made by a specific donor and join with all data about interactions that the donor participated in.	`SELECT *`
	`FROM fundraiser f, interaction_participation_by_persons ipp`
	`WHERE ipp.interaction_name = 'fundraiser',`
	`(GROUP BY f.fundraiser_name`
	`(SELECT Count(*)`
	`FROM fundraiser f, donor d,`
	`interaction_participation_by_persons ipp`
	`WHERE ipp.interaction_name = fundraiser,`
	`d.person_id = ipp.person_id,`
	`GROUP BY f.fundraiser_name));`

ANALYSIS: Use data from SQL and Python's sklearn package to train a **logistic regression model** to predict the likelihood, and amount, of possible future donations relative to each donor -- based on the past activity, behavior, and prior amounts donated.	

Sample data for donations extracted using SQL:

DonorID	Amount	Year Qtr.	Method of P	Event assoc.	Parent	Alumni	Recurring	Board	Number of donations
972	2231.07276	1	check	0	0	1	0	0	0
977	574.995348	3	online	0	0	0	0	0	0
731	1421.75258	3	check	0	0	1	0	0	0
417	677.25195	3	check	1	0	0	0	0	0
730	4364.62901	4	online	0	0	1	0	0	0
533	2182.19065	4	check	1	0	0	1	0	2
1	6564.05512	2	online	0	0	0	0	0	0
116	763.979641	4	check	0	0	0	0	0	0
867	1132.28495	3	credit card	1	0	1	0	0	0
410	3357.08311	2	check	1	0	0	0	0	0
320	2967.64465	1	check	1	0	0	0	0	0
636	3196.12757	4	online	0	0	0	0	0	0
516	7038.66773	3	credit card	1	0	0	1	0	3
839	4351.94887	3	check	1	0	0	0	0	0
552	4774.01596	3	check	0	0	0	0	0	0
933	879.257139	3	check	0	0	0	1	0	8
752	19.034392	3	online	1	1	0	0	0	0
426	3973.22526	4	credit card	0	0	0	1	0	5
941	659.562527	1	check	1	0	0	0	0	0
712	560.536115	4	check	1	0	0	0	0	0
328	20330.6204	2	check	1	0	0	1	0	5
43	58.6087252	2	online	1	0	0	0	0	0

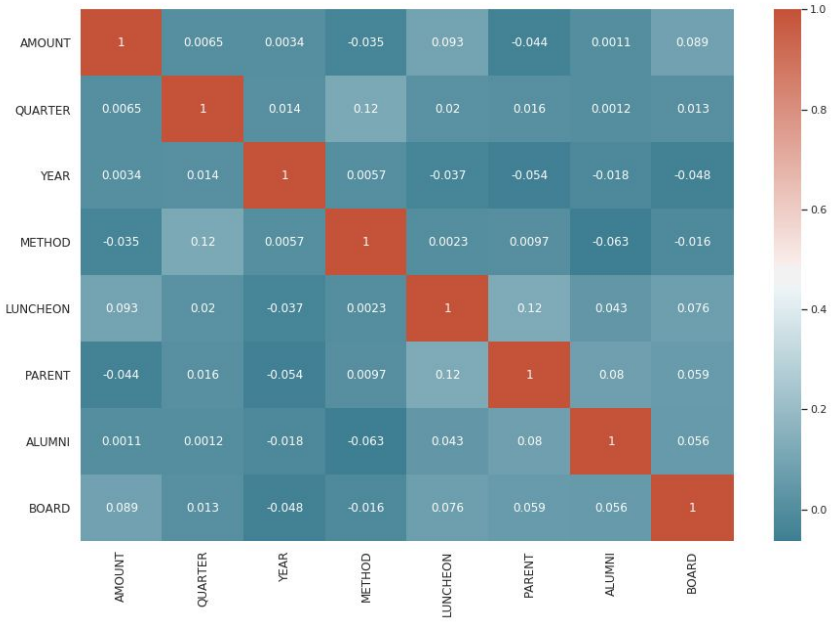
Summary statistics on donations sample data:

	AMOUNT	QUARTER	YEAR	METHOD	LUNCHEON	PARENT	ALUMNI	BOARD
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	149.408000	1.576000	1.478000	1.048000	0.606000	0.628000	0.610000	0.634000
std	79.156814	1.096645	1.075094	0.821612	0.489124	0.483822	0.488238	0.482192

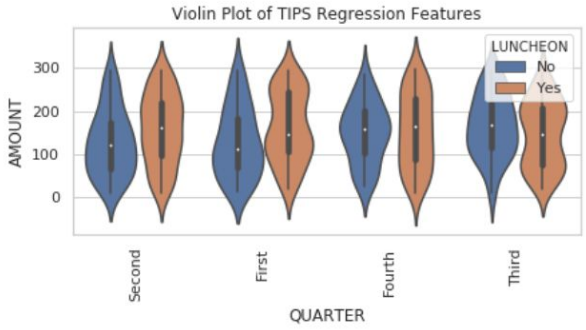
Writing code for the logistic regression model with the sklearn package in Python:

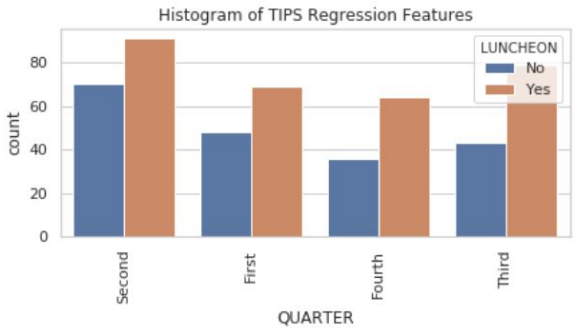
```
import pandas as pd
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression


df= pd.DataFrame.from_csv('/Users/Neel/Downloads/even_better_donation_data.csv')
y= df['Amount'].astype('int')
df['Method of Payment'] = df['Method of Payment'].astype("category").cat.codes
X_train, X_test, y_train, y_test = train_test_split(df, y, test_size=0.15)
logreg = LogisticRegression()
logreg.fit(X_train,y_train)
y_pred=logreg.predict(X_test)
print(y_test, y_pred)
df2 = pd.DataFrame({'Actual': y_test, 'Prediction':y_pred})
df2['RMSE']=((df2['Prediction'] - df2['Actual']) ** 2).mean() ** .5
df2.to_csv('/Users/Neel/Desktop/donorprediction1.csv')
```

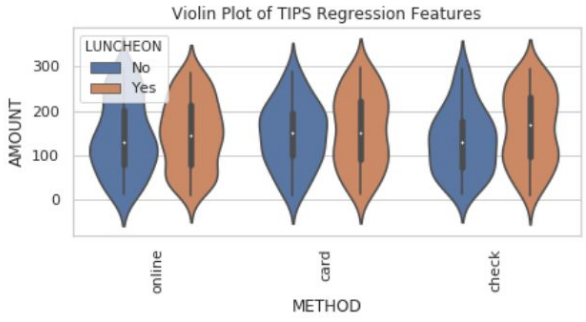
Feature correlation matrix allows us to analyze whether any specific features have more influence:

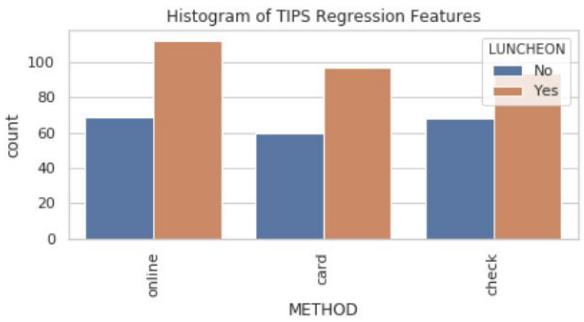


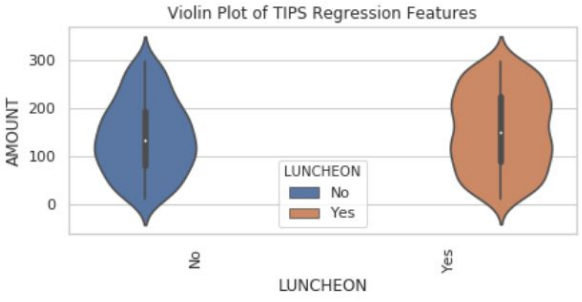
Visualizations of query output comparing amounts and numbers of likely donors on various features:

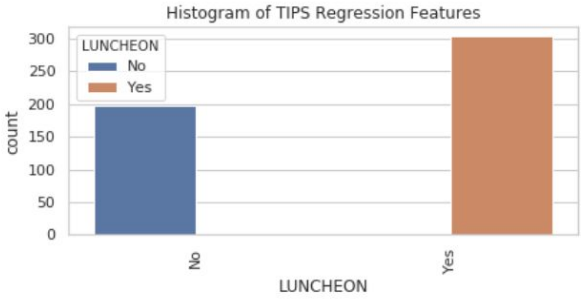












Examples of predicted amounts for donations by donorID (raw data):

DonorID	Actual	Prediction
904	10366	21104
489	686	1905
960	19584	20330
857	3594	10649
526	3181	7149
467	112	131
823	1778	6123
274	10510	11934
420	2177	1905
97	901	3285
611	3551	13644
970	4259	9844
639	7614	19942
367	686	797
504	4077	6123
977	1399	2592
772	1248	1905
590	3604	7600
703	8296	21104
959	225	279
488	5216	12586
679	239	746
930	2573	2700
123	2471	4491
434	1038	2920
731	3221	13644
111	987	2700
752	19	58
3	81	54

494	1708	6965
694	29681	57339
925	4548	10862
537	1835	7149
249	1610	3285
972	2231	9234
984	1078	1905
396	8377	21104
854	1921	1905
500	1633	3285
678	1991	10649
709	18061	19942
238	9821	13644
167	8291	9739
127	4509	4784
164	5214	14788
559	397	1456
185	2354	10649
67	5629	7600
549	10351	36947
313	1301	3973
611	483	1003
164	6507	20065
753	469	797
383	1406	1905
585	1734	2156
496	4645	13644
159	185	279
8	251	1905
508	429	543
944	187	279
988	1952	1905
466	529	1003
712	560	1905
924	9062	21104

494	1708	6965
694	29681	57339
925	4548	10862
537	1835	7149
249	1610	3285
972	2231	9234
984	1078	1905
396	8377	21104
854	1921	1905
500	1633	3285
678	1991	10649
709	18061	19942
238	9821	13644
167	8291	9739
127	4509	4784
164	5214	14788
559	397	1456
185	2354	10649
67	5629	7600
549	10351	36947
313	1301	3973
611	483	1003
164	6507	20065
753	469	797
383	1406	1905
585	1734	2156
496	4645	13644
159	185	279
8	251	1905
508	429	543
944	187	279
988	1952	1905
466	529	1003
712	560	1905
924	9062	21104

4. ALUMNI AND MENTOR SEARCH

Tables Used: Alumni, Person, Skills-Persons junction table

Description	Query 4 SQL
Return first name, last name, email, phone, internship year, and skills of all alumni who possess a certain skill (or interest).	`SELECT p.first_name, p.last_name, p.email, p.phone, a.intern_year, a.education, s.skill_name FROM alumni a, person p, skills_possessed_by_person s WHERE a.PersonID = p.PersonID AND s.Person ID = p.PersonID;`

ANALYSIS: With alumni's skills and other attributes such as career track or education extracted via SQL and arranged into column vectors, we can use a **K-nearest neighbors** analysis to help incoming interns search for and identify those alumni-mentors that are most similar to themselves or have certain qualities or experiences.

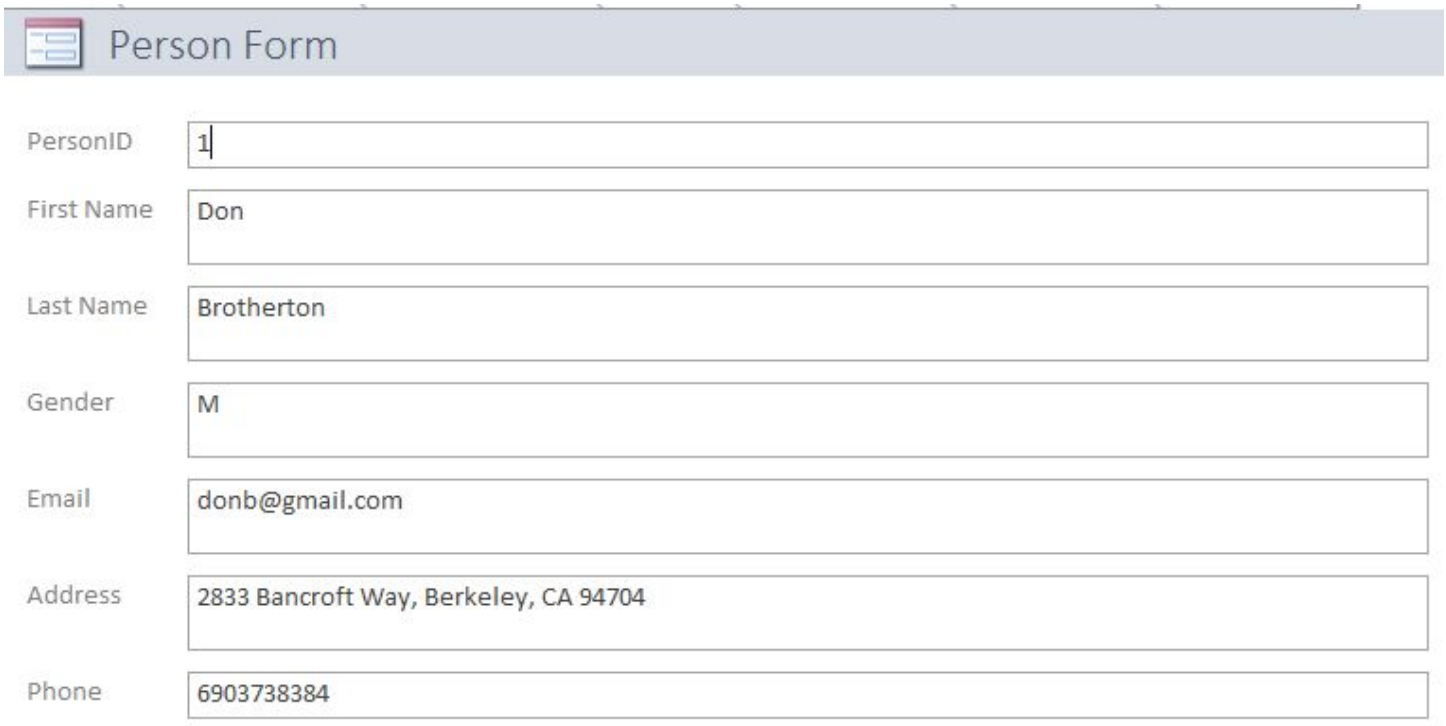
5. RANKING INTERNS BY PERFORMANCE METRIC FROM EVALUATIONS

Tables Used: Evaluation, Internship, Intern, Non-profit contact

Description	Query 5 SQL
Return internship number, intern's first name, non-profit name, and evaluation content for evaluations made by interns and non-profits to analyze differences in performance ratings.	`SELECT i.internship#, p.first_name, np.nonprofit_name, e.evaluation_content FROM person p, intern i, non-profit contact nc, evaluation e WHERE p.PersonID = i.PersonID AND np.personID = i.PersonID;`

ANALYSIS: Using the above query and available unstructured data (scaled response questionnaires from evaluations), we will create a **custom performance metric** for each intern's assignment. We will then create a **ranking of interns**, from best to worst, according to said performance metric which will help the client identify outstanding interns to receive program awards. If possible, we may also conduct a **cluster analysis** to identify what characteristics are dominant in interns with similar rankings.

6.1 ACCESS FORM: PERSON



This form allows us to fill in the Person table, with PersonID, First Name, Last Name, Gender, Email, Address, and Phone.

6.2 ACCESS FORM: INTERACTION

This form allows us to fill in the Interaction table, with Interaction_name, PointPersonID, Event_Date, Start_time, End_time, Budget, and Location_name.

6.3 ACCESS REPORT: PERSON INFORMATION

Person Report

PersonID	First Name	Last Name	Gender	Email	Address	Phone
1	Don	Brotherton	M	donb@gmail.com	2833 Bancroft Way, Berkeley, CA	6903738384
2	Bill	Stewart	M	bs@ymail.com	2831 Haste St, Berkeley, CA 9470	3847505830
3	Jay	Tee	F	eufi@yahoo.com	2189 A Street, San Francisco, CA	3910384811
4	Sarah	Sims	F	ssims13@hotmail.com	1823 California Ave, San Francisc	1122829384
5	Jessica	Jones	F	jj172@berkeley.edu	1234 Dwight Way, Berkeley, CA 9	1283474729
6	Carlos	Martinez	M	jneric@sjsharks.com	2810 Bernal Rd, San Jose, CA 951	4082172933
7	Ty	Guy	M	tgtgtg34@ymail.com	1722 22nd St, Oakland, CA 94182	1927338401
8	James	Logan	M	jlisaqo99@gg.com	3811 Packers Ave, San Mateo, CA	1819394841
9	Larry	Cable	M	LCD@msn.com	1293 Jump St, Sacramento, CA 91	8391193832
10	Jesse	Morty	F	duafa@diy.com	1389 Logan Ave, Elk Groce, CA 91	1238971413
11	Jack	Black	M	duwui@si.com	38 S St, Hollywood, CA 93712	1298371239
12	Kay	See	F	kc@aius.com	2821 Jar Rd, Sunnyvale, CA 9123	4238429342
13	Linda	Lance	F	lilalila@lila.com	342 Lila Ln, San Francisco, CA 931	2130912413

6.4 ACCESS REPORT: INTERNSHIP INFORMATION

Internship Report

Internship#	Internship Year	StartDate	EndDate	Wage	Hours	Intern PID	SectorName	OrgName
1	2018	5/1/2018	8/1/2018	$15.00	130	5	Technology	East Bay Watch
2	2018	5/15/2018	8/1/2018	$15.00	100	8	Finance	ABC Group
3	2018	5/15/2018	8/15/2018	$16.00	80	4	Government	US Federal
4	2018	6/1/2018	9/1/2018	$21.00	150	6	Chemistry	Zyntec
5	2018	5/15/2018	9/15/2018	$13.75	300	10	Education	UC Berkeley
6	2018	6/1/2018	8/31/2018	$18.00	200	1	Education	Ballard High Scho
7	2018	5/15/2018	9/1/2018	$17.00	300	9	Medical	Alta Bates Summi

7. NORMALIZATION ANALYSIS FOR DATABASE DESIGN

In this section, we indicate functional dependencies for five relations of our database design and demonstrate how those relations can be normalized into higher-order forms if necessary.

7.1 SECOND NORMAL FORM DEPENDENCIES ANALYSIS

10. Document(Name, Date, Type, File, Interaction_name[8], Made_by_PID[1])

This relation is already in 2NF because every non-prime attribute is fully dependent on the primary key. We don't need any more information other than the document name to identify every other component of the relation.

7.2 THIRD NORMAL FORM DEPENDENCIES ANALYSIS

2. Internship(Internship#, Internship Year, StartDate, EndDate, Wage, Hours, *SectorName[5], OrgName[4], Intern_PersonID[1]*)

In relation 2, the organization name determines the sector name because each organization belongs to only one sector in our database, so there is a transitive dependency in that Internship# determines OrgName and OrgName determines SectorName. In addition, the internship year could be used to determine the start and end dates of the internship, forming another transitive dependency in that Internship# determines Internship Year, which determines StartDate and EndDate.

To normalize this relation into third normal form, we decompose as follows:

2.1 Internship(Internship#, Wage, Hours, *Intern_PersonID[1]*)
2.2 InternshipDates(Internship_Year, StartDate, EndDate)
2.3 OrgSector(*OrgName[4], SectorName[5]*)

2.1, 2.2, and 2.3 are in 3NF, because 1) they have no multivalued attributes, 2) they have no partial dependencies, and 3) they no longer have any transitive dependencies. In 2.1-2.3, there is no attribute that is transitively dependent on the primary key.

---

7. Application(Application#, FormContent, Submitted_on, Reviewed_on, *InterviewName[8], Submitted_PersonID[1]*)

Application is already in 2NF because it has no multi-valued attributes and every non-prime attribute is fully dependent on the primary key. To normalize it, we decompose it into 7.1 and 7.2.

7.1 Interviewee*(*InterviewName[8]*,*Submitted_PersonID[1]*)*
7.2 Application(Application#, FormContent, Submitted_on, Reviewed_on, *InterviewName[8]*)

7.1 and 7.2 are now in 3NF. In 7, Submitted_PersonID was functionally dependent on InterviewName which was functionally dependent on Application number, because for each unique interview event there is exactly one person being interviewed. In 7.1, we have separated out that relationship, and 7.2 doesn't contain any transitive dependencies, therefore it is now in 3NF.

---

11. Transaction(Transaction_no, Description, Amount, Transaction_Date, Limits, Interaction_name[8])

In 11, Interaction_name could be used to identify the transaction date, because each transaction is associated with an interaction, so Transaction_Date was functionally dependent on Interaction_name, and Interaction_name is functionally dependent on Transaction_no.

11.1 Transaction_Date(Interaction_name[8], Transaction_Date)
11.2 Transaction(Transaction_no, Interaction_name[8] ,Description, Amount, Limits)

11.1 and 11.2 are in 3NF because there are no attributes transitively dependent on the primary key. 11.1 separates out the dependency of the date on the interaction name, leaving no transitive dependencies in 11.1 or 11.2, thus putting them in 3NF.

7.3 BOYCE-CODD NORMAL FORM DEPENDENCIES ANALYSIS

9. Location(Name, Address, Capacity, type, *Interaction_name[8]*)

9.1 Location_Address(Name, Address)
9.2 Location(Name, Capacity, type, *Interaction_name[8]*)

9.1 and 9.2 are in BCNF because 9 was already in 3NF. Also, in 9 address can be used to determine the name of the organization because every organization is linked to one address. However, it is not a superkey because It cannot be used to determine all of the other attributes: the interaction entity, for example, references the location name rather than address, because TIPS employees need to coordinate with the managers of the location, and it is more helpful to refer to it by name than by address alone. Therefore, it violates BCNF and can be normalized to 9.1 and 9.2.

8. TEAM MEMBER CONTRIBUTIONS

Edward Yang: Served as *de facto* CEO, primary liaison for all client communications. Major contributions to EER diagram, query design, presentation building for DP Review II, writing on DP Review III and Final Report. Revisions and edits based on team and instructor feedback for relational schema, technical query analysis, and Access relationship view implementation. Some contributions to technical data analysis. Led processes for internal project management and assisted in communicating to team and instructors.

Talya Zalipsky: Volunteered to be team CCO, kept team members updated on tasks, deadlines, and dates, organized meetings and summarized discussions and decisions. Major contributions to EER diagram revisions, relational schema, and normalization analysis. Helped build presentation for Final, also contributed to revisions and editing processes for all assignments based on team and instructor feedback.

Elias Castro-Hernandez: Served as *de facto* CTO, helped with internal project management and task assignment processes. Participated in one client phone call for brainstorming queries. Contributed to revisions and editing processes to improve assignments based on team and instructor feedback. Major contributions to query design, formatting of DP Review III, and (significantly) to technical data analysis.

Janet Xu: Co-lead liaison for client communications, participated in several email exchanges and phone calls to played a major role in helping to determine project goals, database requirements, query design. Major contributions to SQL queries for DP Review III and some contributions to queries for presentations.

Amanda Wu: Major contributions to building presentations for DP Review II and Final, as well as relational schema design and SQL queries for DP Review III. Also assisted with relational schema edits for Final.

Tuana Arin: Major contributions to presentation for DP Review I and helped build presentations for DP Review II and Final. Participated in a client phone call and helped communicate project updates regarding query design. Helped revise query analyses for DP Review III. Major contributions to normalization analysis.

Michael Chung: Volunteered to be COO. Worked on rough draft of initial ER diagram. Did not attend presentations for DP Reviews II, III or Final. Outlined other members' work but was unreliable in maintaining communication to follow up on assignments and meetings. Major contributions to Access implementation of relational schema (tables) for DP Review III, worked on Access forms and reports for final.

Neel Davar: Did not contribute any significant work to DP Reviews I, II, or III. Worked on rough draft of initial ER diagram. Worked on analysis for query 3, including generating and cleaning sample data, and running and debugging ML model for predictions. Unreliable in maintaining communication to follow up on assignments and meetings. Volunteered to be CTO.

9. DISCUSSIONS AND FUTURE WORK

As a member of the Teens in Public Service board of directors (and a program alumnus), Edward will be presenting a summary of the database design project work to other board members and staff at the next available opportunity, likely in late December or early January. TIPS is a very lean organization and as such it may not be able to fully implement a relational database based on this prototype in the short-term.

That being said, TIPS staff and board are very interested to see what lessons can learned from this project and the process of building a database that can support all of TIPS needs with varying levels of appropriate technical understanding. This experience will be quite valuable in helping TIPS identify and follow better database management practices in the future, and will also show them the potential of powerful data-based workflows and decision-making processes that can help them conduct operations much more efficiently.