Client Overview
Who is our client?
Office of New Student Service

What is Golden Bear Orientation?
- A mandatory week-long program for new undergraduates
- The largest new student orientation in the U.S
- To help New students
  - Make connections to peers, faculty, and staff
  - Learn about available resources
  - Experience campus traditions
  - Broaden understanding of this diverse and dynamic community
  - Create a sense of community with orientation group and trained GBO leaders
Project Goal

- Allows quick look-up for information
- Manages calendar and event data for easy scheduling
- Keeps track of annual orientation data for future reference and improvements
- Creates a cohesive evaluation system for the entire program by linking events, organizers, and students
EER Diagram

2

Client Overview  EER Diagram  Relational Schema  Database  Query  Normalization  Q&A
Relational Schema
Relational Schema

1. PEOPLE (PID, Lname, FName, Address, Email,
   a. PARTNER (PID1, CoordinatorPID1b, Organization7, Availability_Time
      i. EVENT PARTNER (PID1a, 
      ii. BUILDING MANAGER (PID1a, Tel,
      iii. ORGANIZATION CONTACT (PID1a, Tel, Provided_Off_CampusAdresse,
   b. PRO_STAFF/COORDINATOR (PID1, Tel, PositionHeld
   c. ORIENTATION_MENTOR (PID1, Tel, CoordinatorName, GPA, Train
   d. ORIENTATION_LEADER (PID1, Tel, MentorName, RecruitEventID10, GPA
   e. NEW_STUDENTS (PID1, Tel, Sname, PermanentAddress, ResidentialAddress, EmergencyContactPerson, StudentGroupGID8, Gender, Religion, Talents)
   f. APPLICATION_PROFILE(PID1, GPA, Experience_Duration, Position, Specialty_Area)
2. EVENT_EVALUATE_FORM (Event_Form_ID, Ename5, Filled_out_by_PID1,
3. PERSON_EVALUATE_FORM (Person_Form_ID, To_Be_Evaluate_PID1, Filled_out_by_PID1,
Relational Schema

4. LOCATION (LID, Area, GPS
   a. OFF_CAMPUS_ADDRESS (LID4, StreetAddress,)
   b. FIELD (LID4, Fieldname,)
   c. EDUCATIONAL_BUILDING (LID4, ERoomNum, Build_name,PID1a, Capacity
   d. RESIDENTIAL_BUILDING (LID4,StreetAddress, Build_name, PID1a

5. EVENT (Ename, LID4, OName7, Fixlocation{0,1}

6. EVENT_INSTANCE_STUDENT (Ename5, Event_InstanceID, Event_CoordinatorPID1b, Event_Mentor_PID1c, LID4, Student_GroupPID8, Event_PartnerName, Time13

7. ORGANIZATION (Organization, Leader, Type, Office_Location,

8. STUDENT_GROUP (GID, PartnerPID1a, OrientationLeaderPID1d,

9. EVENT_INSTANCE_TRAINING (TrainEventID, LID4,TrainPartner, TrainCoordinator, TrainMentor, AttendingLeader ,StartingTime13,EndingTime13
<table>
<thead>
<tr>
<th>No.</th>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.</td>
<td>EVENT_INSTANCE_RECRUITING</td>
<td>(RecruitEventID, LID4, RecruitedMentor, RecruitingCoordinator, StartingTime13, EndingTime13)</td>
</tr>
<tr>
<td>11.</td>
<td>TOPIC</td>
<td>(TopicName, PartnerName,</td>
</tr>
<tr>
<td>12.</td>
<td>EMERGENCY_CONTACT</td>
<td>(EMCname, StudentPID, Tel,</td>
</tr>
<tr>
<td>13.</td>
<td>30_MIN_TIME_SLOT</td>
<td>(Time, Start_at, End_at, Date,</td>
</tr>
<tr>
<td>14.</td>
<td>SECURITY</td>
<td>(SSN, OName7,</td>
</tr>
<tr>
<td>15.</td>
<td>ROOM</td>
<td>(LID4d, Room_no, Capacity,</td>
</tr>
<tr>
<td>16.</td>
<td>JOINT_PEOPLE_TOPIC</td>
<td>(PID1, TopicName11,</td>
</tr>
<tr>
<td>17.</td>
<td>JOINT_TOPIC_RELATE_TO</td>
<td>(Ename5, Event_InstanceID5,TopicName11),</td>
</tr>
<tr>
<td>18.</td>
<td>JOINT_HOST</td>
<td>(Organization7, Ename 5,</td>
</tr>
<tr>
<td>19.</td>
<td>JOINT_PEOPLE_STUDENT_EVENT</td>
<td>(PID1, Event_InstanceID5),</td>
</tr>
<tr>
<td>20.</td>
<td>JOINT_SECURITY_STUDENT_EVENT</td>
<td>(SSN14, Event_InstanceID5 ),</td>
</tr>
<tr>
<td>21.</td>
<td>JOINT_EVENT_PARTNER_STUDENT_EVENT</td>
<td>(PID1a, Event_InstanceID5)</td>
</tr>
</tbody>
</table>

Relational Schema

22. JOINT_PEOPLE_EVENT_INSTANCE_STUDENT(PID1a, Ename5, Event_InstanceID)
23. JOINT_ORGNIZATION_CONTACT_OFF_CAMPUS_ADDRESS (PID1ai, LID4a)
24. JOINT_RESIDENTIAL_BUILDING_MANAGER (LID4d, PID1ali)
25. JOINT_EVENT_PARTNER_TOPIC (TopicName11, PID1ai)
26. JOINT_STUDENT_EMERGENCY_CONTACT (PID1E, EMCname, StudentPID12)
27. JOINT_MENTOR_LEADER (MentorPID1c, LeaderPID1d)
28. JOINT_EVENT_TRAINING_LEADER (TrainEventID9, PID1d)
29. JOINT_MENTOR_RECRUITING (PID1C, RecruitEventID10)
30. JOINT_MENTOR_EVENT_TRAINING (PID1c, TrainEventID9)
31. JOINT_PARTNER_EVENT_TRAINING (PID1a, TrainEventID9)
32. JOINT_COORDINATOR_EVENT_TRAINING (PID1b, TrainEventID9)
33. JOINT_COORDINATOR_RECRUITING (PID1b, RecruitEventID10)
34. JOINT_LEADER_RECRUITING (PID1d, RecruitEventID10)
35. JOINT_EVENT_INSTANCE_STUDENT_GROUP(EName, Event_InstanceID, Student_GroupPID8,
<table>
<thead>
<tr>
<th>Client Overview</th>
<th>EER Diagram</th>
<th>Relational Schema</th>
<th>Database</th>
<th>Query</th>
<th>Normalization</th>
<th>Q&amp;A</th>
</tr>
</thead>
</table>

Database 4
Query 1 - Pick The Top Applicants Based On Past Evaluation Data

Benefits - GBO will be able to pick out a certain number of applicants without going through every Application

Automate Recruiting Process:
- Reduce time spent on selecting potential candidates
- Minimize human error on comparing applicants

Approaches
1) Build Linear Model
   - Convert categorical variables in applicant information to dummy variables
   - Select features from applicant information based on the linear model
2) Create Measure Metric
   - Assign scores to each applicant

SQL Code
```
FROM People_Evaluation_Form AS E,
Applicant_Profile AS p
WHERE (((p.PID)=[E].[PID]));
```

Access Query

<table>
<thead>
<tr>
<th>PID</th>
<th>Average</th>
<th>GPA</th>
<th>Experience_Duration</th>
<th>Past_Position</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>6001</td>
<td>2.666666667</td>
<td>3.75</td>
<td>1</td>
<td>Volunteer, Ass</td>
<td>Community</td>
</tr>
<tr>
<td>6002</td>
<td>3</td>
<td>3.94</td>
<td>0.5</td>
<td>Researcher</td>
<td>Academic</td>
</tr>
<tr>
<td>6003</td>
<td>4.333333333</td>
<td>3.43</td>
<td>0.4</td>
<td>Coordinator</td>
<td>Private_Sector</td>
</tr>
<tr>
<td>6004</td>
<td>4.666666667</td>
<td>3.36</td>
<td>2</td>
<td>Volunteer, Ass</td>
<td>Community</td>
</tr>
<tr>
<td>6005</td>
<td>3</td>
<td>2.96</td>
<td>3</td>
<td>Assistant,Coord</td>
<td>Private_Sector</td>
</tr>
</tbody>
</table>

Improving Procedure
In the end, we can achieve an ordered list of candidates based on decreasing predicted scores.
Query 2 - How Many Staff Members are Needed for Check-Ins at Events?

Benefits
- Personels are limited for GBO
- Allocate optimal number of staff members according to needs
- Increase students satisfaction (Over 1 hour of waiting time from last year)

What’s Required
- Student arrival data from previous year
- Any event that requires check-in
- Increase students satisfaction

How?
- SQL provides arrival data
- Arrival data fit into distribution
- SIGMA simulation used to find optimal number of staff
Query 2 - How Many Staff Members are Needed for Check-ins at Events?

SQL Code

```
SQL <- SELECT S.checkin_date, count(s.Sid)
FROM New_Student as s
GROUP by s.Checkin_date
```

Assume exponential distribution

Lambda = arrival rate

Service rate = 5min/person

<table>
<thead>
<tr>
<th>Day</th>
<th>Number of Arrivals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>327</td>
</tr>
<tr>
<td>2nd</td>
<td>413</td>
</tr>
<tr>
<td>3rd</td>
<td>860</td>
</tr>
</tbody>
</table>
## Query 2 - How Many Staff Members are Needed for Check-ins at Events?

### Results:

<table>
<thead>
<tr>
<th>Day</th>
<th>Number of Staff</th>
<th>Average Queu Size</th>
<th>Server Utilization</th>
<th>Average Waiting Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>4</td>
<td>4.60</td>
<td>0.80</td>
<td>5.69</td>
</tr>
<tr>
<td>2nd</td>
<td>5</td>
<td>3.65</td>
<td>0.77</td>
<td>5.67</td>
</tr>
<tr>
<td>3rd</td>
<td>9</td>
<td>11.97</td>
<td>0.81</td>
<td>5.68</td>
</tr>
</tbody>
</table>

### What this means

- If we put 4 staff members on day 1, 5 on day 2, and 9 on day 3
- Server utilization: how busy they are
- Waiting time: how long students must wait in line
- The office can set up goals for these parameters
Query 3 - Which and How Many Events Should GBO Hold?

Benefits:
- Better student engagement and positive experience
- GBO can avoid hosting events that are not popular to save resources
- New events can be planned according to the predicted interests

Improving Procedure:

SQL Code <- SELECT s.SID, s.gender, s.Admit_type, s.college, s.religion, s.Sexorientation, s.ethic, j1.Tname AS Interest1, j2.Tname AS Interest2
FROM New_Student AS s, Joint_PEOPLE_TOPIC AS j1, Joint_PEOPLE_TOPIC AS j2
WHERE j1.Tname<j2.Tname AND s.admit_year <= 2018 AND j1.pid=s.pid AND j2.pid=s.pid;
## save as training set

SQL Code <- SELECT s.SID, s.gender, s.Admit_type, s.college, s.religion, s.Sexorientation, s.ethic
FROM New_Student AS s, Joint_PEOPLE_TOPIC AS j1, Joint_PEOPLE_TOPIC AS j2
WHERE s.admit_year=2018 AND j1.pid=s.pid AND j2.pid=s.pid;
## save as Testing set
Query 3 - LDA v.s. CART

Result (LDA):

```r
# Code for LDA
mdl1 = fitcdiscr(traindata(:,1:2); traindata(:,end-1));
mdl2 = fitcdiscr(traindata(:,1:2); traindata(:,end));
result(1) = mdl1;
result(2) = mdl2;
mdl3 = TreeBagger(4, traindata(:,1:2); traindata(:,end-1));
mdl4 = TreeBagger(4, traindata(:,1:2); traindata(:,end));
result(1) = mdl3;
result(2) = mdl4;

t1 = 0;
t2 = 0;
for i = 1:row2
    inte_lda(i,1) = predict(mdl1, testdata(1,2:end));
    inte_lda(i,2) = predict(mdl2, testdata(1,2:end));
end
out_lda = [testdata(:,1), inte_lda];
for i = 1:row2
    for j = 1:2
        stddata(j)(i) = parameter(out_lda(i,j+1),1);
    end
end
StudentID = num2str(out_lda(:,1));
FirstInterest = str2mat(stddata(1));
SecondInterest = str2mat(stddata(2));
T = table(StudentID, FirstInterest, SecondInterest);
writefile(T, 'lda_prediction.xlsx', 'Sheet', 1, 'Range', 'A1');
for i = 1:row2
    inte_tree(i,1) = predict(mdl3, testdata(1,2:end));
    inte_tree(i,2) = predict(mdl4, testdata(1,2:end));
end
out_trees = [testdata(:,1), inte_tree];
for i = 1:row2
    for j = 1:2
        stddata(j)(i) = parameter(out_tree(i,j+1),1);
    end
end
StudentID = num2str(out_trees(:,1));
FirstInterest = str2mat(stddata(1));
SecondInterest = str2mat(stddata(2));
```

Result (CART):

```r
# Code for CART
```
Query 4 - Finding a friend for each student

**SQL Code:**

```sql
SELECT P1.PID, P2.PID, P2.GID
FROM New_Student AS P1,
     New_Student AS P2, EventdesignSQL AS E1, EventdesignSQL AS E2
WHERE P1.SID = E1.SID
     AND P2.SID = E2.SID
     AND E1.Interest1 = E2.Interest1
     AND E1.Interest2 = E2.Interest1
     AND E1.Interest2 = E2.Interest2
     AND P1.PID < P2.PID;
```

**Benefits:**

Golden Bear Orientation would be most students’ first time away from their family and old friends. When students are new to Cal, we want to make sure that every one of them has at least one peer to talk to. The pair we matched would share two interests.
Welcome,
You are Staff Member of GBO

Create and Edit
- Student Profile
- Leader Profile
- Mentor Profile
- Staff Profile
- Student Group
- Applicant Profile
- Event Design Wizard

View data
Enter the student PID.

BestFriend Lookup

Note: this magical button will list all other student PID who share the same interest with this student

Enter the GID you wish to look up

Look up a Group Schedule

GetBestFriend1

<table>
<thead>
<tr>
<th>P1.PID</th>
<th>P2.PID</th>
<th>GID</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>593</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>763</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>854</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>883</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>958</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>1510</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>1517</td>
<td>77</td>
</tr>
</tbody>
</table>

November 30, 2017

Page 1 of 1
Query 5- Generate GBO Week Event Schedule For Each Group

Benefits - Orientation leader will be able to retrieve GBO schedule as a report on Access.
- User friendly for everyone
- A student can only see its group’s schedule
- For staff, a look-up button is designed to help quickly finding a group’s schedule

### Query 5- Example Schedule Output

#### Generate Schedule

<table>
<thead>
<tr>
<th>Group</th>
<th>Day</th>
<th>Start</th>
<th>End Event</th>
<th>Location</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8/1/2017</td>
<td>10:00</td>
<td>11:00 Campus tour</td>
<td>RSF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/1/2017</td>
<td>11:00</td>
<td>12:00 Campus tour</td>
<td>Tang Center</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/1/2017</td>
<td>12:00</td>
<td>13:30 Lunch</td>
<td>Crossroad</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/1/2017</td>
<td>13:30</td>
<td>15:30 Fashion TED Talk</td>
<td>I-House</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/1/2017</td>
<td>20:00</td>
<td>21:00 Dinner</td>
<td>Crossroad</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/2/2017</td>
<td>11:00</td>
<td>12:00 Escape Room</td>
<td>Barrows 102</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/2/2017</td>
<td>12:00</td>
<td>13:30 Lunch</td>
<td>Crossroad</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/2/2017</td>
<td>13:30</td>
<td>15:30 Campus tour</td>
<td>Foothill</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/2/2017</td>
<td>20:00</td>
<td>21:00 Dinner</td>
<td>Crossroad</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/2/2017</td>
<td>21:00</td>
<td>22:00 Song Writing</td>
<td>Morrison Hall 100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8/3/2017</td>
<td>10:00</td>
<td>11:00 OAK Museum</td>
<td>Oakland Museum of Calif</td>
<td></td>
</tr>
</tbody>
</table>
Normalization: 1NF

PARTNER (PID¹, Email, Tel, Notes, Organization⁶)

Remove multivalued attribute

1. PARTNER (PID¹, Email, Tel, Note)
2. Partner_Organization(PID¹, Organization²)

After normalized, the relation is in Boyce_Coded Normal Form.
**Normalization: 2NF**

EMERGENCY_CONTACT (PID\(^{1a}\), EMCID\(^{2}\), Relationship, EMCLName, EMCFName, Tel, email)

Remove partial dependencies

1. Joint_STUDENT_EMC (PID\(^{1a}\), EMCID\(^{2}\), Relationship)
2. EMERGENCY_CONTACT (EMCID, EMCLName, EMCFName, Tel, email)

After normalized, the relation is in Boyce_Coded Normal Form.
Normalization: 3NF

LOCATION (LID, GPS_Longitude, GPS_Latitude, Location_Name, Address, Room, PublicTransitable, Managed_by_Oname)

Remove transitive dependencies

1. LOCATION (LID, GPS_Longitude, GPS_Latitude, Location_Name, Managed_by_Oname)
2. LOCATION_DETAIL (Location_Name, Address, Room, PublicTransitable)

After normalized, the relation is in Boyce_Coded Normal Form.
Q&A
Thank you!