Garrett Strommen President/Founder

**Strommen Tutoring:** private language tutoring company  
Los Angeles, CA  
Founded 2009

2 full-time Employees

Over **70** contracted Teachers

Over **1,700** students  
Clients include Sylvester Stallone’s kids
Provided Services:

**Group Classes:**
Recurring classes scheduled for certain times in certain locations.

**Private Tutoring:**
One-on-one or small group tutoring.

**Translation Services:**
Translations for documents or books.
Project Goals.

Existing Database Support: MySQL web database and PHP back end
New Database Support: Microsoft Access

- Observe Company Operations
- Track Company Transactions
- Manage Marketing Campaigns
- Analysis and Reporting
**Project Progress.**

**DP 1**
- Preliminary database design
- Simplified EER Diagram (25 entities)
- Familiarization with the Strommen business model

**DP 2**
- 5 Queries conceptualized
- EER expanded, a lot
- Relational design schema developed
- Relations implemented in Access
Project Progress.

DP 3
• Major query revision
• Complexities added with AMPL and sub-queries
• Query implementation in SQL
• Relations defined in database

Final Result
• Queries implemented in Access
• Practical forms and reports created
• Relationships altered slightly
• Normalization analysis
eer diagram.
eer diagram.
relational schema.
Relational Schema.

1. **Person** (PersonID, FName, LName, MInitial, Address, Phone, Email, Gender, BirthDate)
2. **Prospect** (ProspectID, PersonID, ContactMethod, DateAdded)
3. **EmployeeProspect** (EmployeeProspectID, ProspectID, YearsExperience, Skills, Position)
4. **TeacherProspect** (TeacherProspectID, ProspectID, YearsExperience, Occupation)
5. **ContractorProspect** (ContractorProspectID, ProspectID, YearsExperience, CompanyName, ContractorType)
6. **ClientProspect** (ClientProspectID, ProspectID, EventID, ZipCode)
7. **Contractor** (ContractorID, PersonID, ContractorProspectID, CompanyName, ContractorType)
8. **Employee** (EmployeeID, PersonID, EmployeeProspectID, Skills, Position, StartDate, Salary, BankAccount)
9. **Client** (ClientID, PersonID, EventID, ClientProspectID, ZipCode, ReferredByID)
10. **Teacher** (TeacherID, PersonID, TeacherProspectID)
11. **TeacherEvaluation** (TeacherEvaluationID, TeacherID, ClientID, TopicID, EvaluationDate, TeacherRating, Comments)
12. **Reward** (RewardID, ClientID, RewardType, Description, DateAwarded, Received)
13. **Translation** (TranslationID, InitialLanguage, FinalLanguage, ClientID, EstCost, FinalCost, StartDate, FinishDate, TeacherID, ContractorID)
14. **ZipCode** (ZipCode, City, Description, Latitude, Longitude)
15. **TimeSlot** (TimeSlotID, TimeSlotDate, TimeSlotTime)
16. **Campaign** (CampaignID, CampaignName, StartDate, EndDate, Budget, EmployeeID)
17. **Event** (EventID, CampaignID, Date, Cost, Location, ZipCode)
18. **Bonus** (BonusID, Amount, EmployeeID, TeacherID, DateEarned)
19. **ContractJob** (ContractJobID, ContractJobName, Description, EstCost, FinalCost, ContractorID, StartDate, FinishDate)
20. **ContractItem** (ContractItemID, ContractJobID, Description, EstHours, FinalHours, HourlyRate, EstCost, FinalCost)
Relational Schema Continued.

21. **Topic** *(TopicID, TopicName, TopicType, Description)*
22. **Session** *(SessionID, SessionName, ExperienceLevel, TopicID, TeacherID, ZipCode, PrivateSessionID, GroupClassID)*
23. **PrivateSession** *(PrivateSessionID, TravelDistance, AtClientHome)*
24. **GroupClass** *(GroupClassID, Location, Capacity)*
25. **SessionMaterial** *(SessionMaterialID, SessionID, MaterialName, Cost, Description, Required, Provided)*
26. **Transaction** *(TransactionID, Amount, OpenDate, CloseDate, Paid, WithdrawalID, DepositID)*
27. **Deposit** *(DepositID, SessionID, TranslationID, ClientID)*
28. **Withdrawal** *(WithdrawalID, CampaignID, BonusID, SessionID, TranslationID, ContractJobID, EmployeeID)*
29. **ExperienceLevel** *(ExperienceLevelID, Level, Description)*
30. **Event_TimeSlot** *(EventID, TimeSlotID)*
31. **Session_Client** *(SessionID, ClientID)*
32. **Session_TimeSlot** *(SessionID, TimeSlotID)*
33. **Teacher_TimeSlot** *(TeacherID, TimeSlotID)*
34. **Teacher_Topic** *(TeacherID, TopicID)*
35. **Teacher_ZipCode** *(TeacherID, ZipCodeID)*
36. **TeacherProspect_Topic** *(TeacherProspectID, TopicID)*
37. **TeacherProspect_ZipCode** *(TeacherProspectID, ZipCode)*
38. **Client_Topic** *(ClientID, TopicID)*
39. **ClientProspect_Topic** *(ClientProspectID, TopicID)*
Relationship Diagram.
normalization
12. **Reward** (RewardID, ClientID, RewardType, Description, DateAwarded, Received)

**Functional Dependencies**

{RewardID} → {RewardType}

{RewardID, ClientID} → {DateAwarded, Received, Description}
1NF There are no multi-valued attributes.

**Reward** *(RewardID, ClientID⁹, RewardType, Description, DateAwarded, Received)*

2NF & 3NF & BCNF

**RewardType** *(RewardID, RewardType)*

**RewardInfo** *(RewardID, ClientID⁹, Description, DateAwarded, Received)*
25. **SessionMaterial** *(SessionMaterialID, SessionID\textsuperscript{22}, MaterialName, Cost, Description, Required, Provided)*

**Functional Dependencies**

\{SessionMaterialID\} \rightarrow \{MaterialName\}
\{SessionMaterialID, SessionID\} \rightarrow \{Required, Provided\}
\{MaterialName\} \rightarrow \{Cost, Description\}
Normalization.

1NF Currently in 1NF because there are no multi-valued attributes.

**SessionMaterial** *(SessionMaterialID, SessionID\textsuperscript{22}, MaterialName, Cost, Description, Required, Provided)*

2NF Removes partial dependencies that non-prime attributes have on the Candidate Key

**SessionMaterial** *(SessionMaterialID, SessionID\textsuperscript{22}, Required, Provided)*

**Material** *(SessionMaterialID, MaterialName, Cost, Description)*
Normalization.

3NF Removes transitive dependency from Material Name determining cost and description

**SessionMaterial** *(SessionMaterialID, SessionID	extsuperscript{22}, Required, Provided)*

**Material** *(SessionMaterialID, MaterialName)*

**MaterialInfo** *(MaterialName, Cost, Description)*

This is also in BCNF!
queries.
Query 1: ROI for Marketing Campaigns

Objective: Determine the most cost effective marketing campaigns and events in terms of Return on Investment.
Query 1:
ROI for Marketing Campaigns

Purpose: Marketing Campaigns with the highest ROI are the most cost effective. ROI analysis would allow Strommen Tutoring to replicate the design of effective campaigns to better target profitable clients.
Query 1: ROI for Marketing Campaigns

Query: Select the name, cost, revenue, and ROI for each Marketing Campaign.

Let \( P_j = \text{net profit per client converted by a marketing event} \)
\( C_j = \text{cost of a marketing event} \)

Then \( ROI = \frac{\sum_{j=1}^{n} \sum_{i=1}^{m} P_{ij} - \sum_{j=1}^{n} C_j}{\sum_{j=1}^{n} C_j} \)
Query 1:
ROI for Marketing Campaigns

SQL>

SELECT RevenueQuery.Campaign AS CampaignID, RevenueQuery.CampName AS CampaignName,
CostQuery.Cost AS TotalCost, RevenueQuery.Revenue AS TotalRevenue, IIF(TotalCost > 0,
TotalRevenue*100 / TotalCost & "%",0 & "%") AS ROI
FROM (SELECT c.CampaignID AS Campaign, c.CampaignName AS CampName,
   IIF(SUM(dt.amount)>0, 0.3*SUM(dt.amount),0) AS Revenue
FROM Campaign AS c, Event AS e, Client AS cl, Session_Client AS sc, [Session] AS s,
   (SELECT * FROM Deposit AS d, [Transaction] AS t1 WHERE
   d.DepositID = t1.DepositID) AS dt
   cl.ClientID = sc.ClientID AND sc.SessionID = s.SessionID AND dt.SessionID =s.SessionID
   AND dt.ClientID = cl.ClientID
GROUP BY c.CampaignID, c.CampaignName) AS RevenueQuery,
(SELECT c2.CampaignID AS Campaign, IIF(SUM(wt.Amount) > 0, SUM(wt.Amount), 0) AS Cost
FROM Campaign AS c2,
   (SELECT * FROM Withdrawal AS w, [Transaction] AS t2 WHERE
   w.WithdrawalID=t2.WithdrawalID) AS wt
WHERE wt.CampaignID = c2.CampaignID
GROUP BY c2.CampaignID) AS CostQuery
WHERE RevenueQuery.Campaign = CostQuery.Campaign
GROUP BY RevenueQuery.Campaign, RevenueQuery.CampName, CostQuery.Cost,
RevenueQuery.Revenue;
Query 1:
ROI for Marketing Campaigns

Results:

<table>
<thead>
<tr>
<th>CampaignID</th>
<th>CampaignName</th>
<th>TotalCost</th>
<th>TotalRevenue</th>
<th>ROI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tabling</td>
<td>$60.00</td>
<td>373.5</td>
<td>622.5%</td>
</tr>
<tr>
<td>2</td>
<td>Flyering</td>
<td>$50.00</td>
<td>171</td>
<td>342%</td>
</tr>
</tbody>
</table>
Query 2: Optimal Teacher Assignment

Objective: Find the best-suited teacher for each session not yet assigned a teacher based on Teacher Rating, Zip Code, Availability and Topic.
Query 2: Optimal Teacher Assignment

Purpose: To reward the most highly-rated teachers by assigning them the most new clients, while also providing the best possible first impression of Strommen Tutoring.
Query 2: Optimal Teacher Assignment

Query: Select a teacher for each session not yet assigned a teacher that can teach for the entire session and in the same zipcode. If there are multiple feasible teachers, select the one with the highest average teacher rating from teacher evaluations.
Query 2: Optimal Teacher Assignment

```sql
FROM (SELECT SuperQuery.SubSession AS SuperSession, Max(SuperQuery.SubRating) AS MaxRating
FROM (SELECT SubQuery.Sessions AS SubSession, SubQuery.Teachers AS SubTeacher,
(SELECT Avg(te.TeacherRating) FROM TeacherEvaluation te WHERE te.TeacherID = SubQuery.Teachers) AS SubRating
FROM (SELECT s.SessionID AS Sessions, t.TeacherID AS Teachers FROM Teacher AS t, [Session] AS s, Teacher_ZipCode AS tzc, Teacher_Topic AS tt
```
# Query 2: Optimal Teacher Assignment

Sessions without Teachers assigned

<table>
<thead>
<tr>
<th>SessionID</th>
<th>SessionName</th>
<th>ExperienceL</th>
<th>TopicID</th>
<th>TeacherID</th>
<th>ZipCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>76</td>
<td>Tutoring</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>90001</td>
</tr>
<tr>
<td>77</td>
<td>Tutoring</td>
<td>3</td>
<td>2</td>
<td></td>
<td>94129</td>
</tr>
<tr>
<td>78</td>
<td>Tutoring</td>
<td>2</td>
<td>3</td>
<td></td>
<td>93109</td>
</tr>
<tr>
<td>79</td>
<td>Tutoring</td>
<td>1</td>
<td>1</td>
<td></td>
<td>94129</td>
</tr>
<tr>
<td>80</td>
<td>Tutoring</td>
<td>3</td>
<td>2</td>
<td></td>
<td>90001</td>
</tr>
<tr>
<td>81</td>
<td>Tutoring</td>
<td>5</td>
<td>1</td>
<td></td>
<td>33128</td>
</tr>
<tr>
<td>82</td>
<td>Tutoring</td>
<td>6</td>
<td>2</td>
<td></td>
<td>90210</td>
</tr>
</tbody>
</table>
Query 2: Optimal Teacher Assignment

Results

<table>
<thead>
<tr>
<th>SessionID</th>
<th>TeacherName</th>
<th>TeacherID</th>
</tr>
</thead>
<tbody>
<tr>
<td>77</td>
<td>Anne Eddison</td>
<td>2</td>
</tr>
<tr>
<td>78</td>
<td>Shaughn Reilly</td>
<td>5</td>
</tr>
<tr>
<td>79</td>
<td>Shaughn Reilly</td>
<td>5</td>
</tr>
<tr>
<td>80</td>
<td>Monica Brynne-Wi</td>
<td>4</td>
</tr>
<tr>
<td>81</td>
<td>John Dorian</td>
<td>1</td>
</tr>
<tr>
<td>82</td>
<td>Monica Brynne-Wi</td>
<td>4</td>
</tr>
</tbody>
</table>
Query 3:
Reward Allocation System

Objective: Find clients who are eligible for rewards and what rewards they are eligible for.
Purpose: Rewards are a way to encourage customer loyalty, through a variety of different incentives that may appeal to the clients. Rewards also draw new clients.
Query 3: Reward Allocation System

Reward Types Examples

- Shirts: After 5 Sessions
- Water Bottle: After 10 Sessions
- Sessions: Every 10 Sessions
Query 3: Reward Allocation System

Query: For each client, determine the number of rewards he/she is eligible for but has not yet been assigned.

\( n = \# \text{ sessions a client has purchased} \)

\[
\begin{align*}
\text{T Shirt} & = 1 \text{ if } n \geq 5 \\
& = 0 \text{ otherwise} \\
\text{Water Bottle} & = 1 \text{ if } n \geq 10 \\
& = 0 \text{ otherwise} \\
\text{Sessions} & = \text{Floor}(n / 10)
\end{align*}
\]

\( \# \text{ of Each Reward Already Assigned} \)
FROM (
    SELECT CountCalc.Client AS Client, RefCount, FreeSCount, ClientSCount, WaterBottle, TShirt
    FROM (SELECT sc.ClientID AS ClientID FROM Client AS c INNER JOIN Session_Client AS sc ON sc.ClientID = c.ClientID) AS ClientSessions
    AS Client, ClientSCount, NeverWater, NeverTShirt FROM (SELECT sc.ClientID AS ClientID FROM Client AS c INNER JOIN Session_Client AS sc ON sc.ClientID = c.ClientID) AS ClientSessions
    AS Client, ClientSCount, NeverWater, NeverTShirt FROM (SELECT sc.ClientID AS ClientID FROM Client AS c INNER JOIN Session_Client AS sc ON sc.ClientID = c.ClientID) AS ClientSessions
    GROUP BY ClientSessions.ClientID) AS final;

SQL>
### Query 3: Reward Allocation System

#### Results

<table>
<thead>
<tr>
<th>Client</th>
<th>ClientName</th>
<th>WaterBottle</th>
<th>TShirt</th>
<th>EligibleFreeSession</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ian Herbert</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Michael Michaels</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Bobby Jones</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Sarah Brown</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Tim Thee</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
**Query 3:**

**Reward Allocation System**

**Form Application:**

<table>
<thead>
<tr>
<th>ClientName</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>WaterBottle</td>
<td>1</td>
</tr>
<tr>
<td>TShirt</td>
<td>1</td>
</tr>
<tr>
<td>EligibleFreeSession</td>
<td>3</td>
</tr>
</tbody>
</table>

**Click this Button!**

**Create Rewards!**
Query 3: Reward Allocation System

Query updates the # of eligible rewards to 0 for each Reward Type.

New Rewards are created in the database using Visual Basic.
Objective: Determine trends concerning the # of sessions for different topics over the last 2 years. Analyze the significance of outliers in the data and actions to support or prevent these outliers.
Purpose: Help target appropriate markets at appropriate times. Adjust hiring practices to seasonal trends.
Query 4: Monthly Session Process Control

Query: For each topic for each month (for the last 24 months) find the number of sessions provided and the deviation from the average.

Query Results:

\[ E[x_i] = \mu \]
\[ \text{Var}[x_i] = \frac{\sigma^2}{n} \]
\[ \text{Deviation} = x_i - \mu \]

for all \( i \) months for each topic

Statistical Process Control:

X-Chart

\[ \mu \pm 2\frac{\sigma}{\sqrt{n}} \]

Upper and Lower Control Limits for 95% Probability Range
SELECT tc.TopicID, tc.TopicName, tc.Year, tc.Month, DateValue(tc.Month & "/1/" & tc.Year) AS ProcessDate, TopicCount, AverageTopicCount, TopicCount - AverageTopicCount AS DeviationFromAverage, StandardDeviation AS SD, IIF(StandardDeviation >0 And (DeviationFromAverage / StandardDeviation <= -2 OR DeviationFromAverage / StandardDeviation >= 2), "Out Of Control", "In Control") AS Status
FROM (SELECT t.TopicID AS TopicID, t.TopicName AS TopicName, Year(ts.TimeSlotDate) AS [Year], Month(ts.TimeSlotDate) AS [Month], COUNT(s.SessionID) AS TopicCount FROM Topic AS t, [Session] AS s, TimeSlot AS ts, Session_TimeSlot AS sts WHERE sts.SessionID = s.SessionID AND s.TopicID = t.TopicID AND sts.TimeSlotID = ts.TimeSlotID AND ((Month(ts.TimeSlotDate) >= Month(Date()) AND Year(Date()) - Year(ts.TimeSlotDate) = 2) OR (Year(Date()) - Year(ts.TimeSlotDate) < 2)) GROUP BY t.TopicID, t.TopicName, Year(ts.TimeSlotDate), Month(ts.TimeSlotDate)) AS tc INNER JOIN (SELECT TopicID, ROUND(Avg(TopicCount), 2) AS AverageTopicCount, ROUND(StDev(TopicCount),2) AS StandardDeviation FROM ((SELECT t.TopicID AS TopicID, Year(ts.TimeSlotDate) AS [Year], Month(ts.TimeSlotDate) AS [Month], COUNT(s.SessionID) AS TopicCount FROM Topic AS t, [Session] AS s, TimeSlot AS ts, Session_TimeSlot AS sts WHERE sts.SessionID = s.SessionID AND s.TopicID = t.TopicID AND sts.TimeSlotID = ts.TimeSlotID AND ((Month(ts.TimeSlotDate) >= Month(Date()) AND Year(Date()) - Year(ts.TimeSlotDate) = 2) OR (Year(Date()) - Year(ts.TimeSlotDate) < 2)) GROUP BY t.TopicID, Year(ts.TimeSlotDate), Month(ts.TimeSlotDate)) AS SubQuery) GROUP BY TopicID) AS atc ON tc.TopicID = atc.TopicID ORDER BY tc.TopicID, tc.Year DESC , tc.Month DESC;
Query 4: Monthly Session Process Control

<table>
<thead>
<tr>
<th>TopicID</th>
<th>TopicName</th>
<th>Year</th>
<th>Month</th>
<th>ProcessDate</th>
<th>TopicCount</th>
<th>AverageTop</th>
<th>DeviationFromAverage</th>
<th>SD</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2013</td>
<td>5</td>
<td>5/1/2013</td>
<td>2</td>
<td>1.45</td>
<td></td>
<td>0.55</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2013</td>
<td>4</td>
<td>4/1/2013</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2013</td>
<td>3</td>
<td>3/1/2013</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2013</td>
<td>2</td>
<td>2/1/2013</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2013</td>
<td>1</td>
<td>1/1/2013</td>
<td>2</td>
<td>1.45</td>
<td>0.55</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2012</td>
<td>12</td>
<td>12/1/2012</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2012</td>
<td>11</td>
<td>11/1/2012</td>
<td>3</td>
<td>1.45</td>
<td>1.55</td>
<td>0.74</td>
<td>0.74 Out Of Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2012</td>
<td>10</td>
<td>10/1/2012</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2012</td>
<td>8</td>
<td>8/1/2012</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2012</td>
<td>7</td>
<td>7/1/2012</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2012</td>
<td>6</td>
<td>6/1/2012</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
<tr>
<td>1</td>
<td>Spanish</td>
<td>2012</td>
<td>5</td>
<td>5/1/2012</td>
<td>1</td>
<td>1.45</td>
<td>-0.45</td>
<td>0.74</td>
<td>0.74 In Control</td>
</tr>
</tbody>
</table>
Query 4: Monthly Session Process Control

Report Application:

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th># Sessions</th>
<th>Deviation</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>5</td>
<td>2</td>
<td>0.55</td>
<td>In Control</td>
</tr>
<tr>
<td>2013</td>
<td>4</td>
<td>1</td>
<td>-0.45</td>
<td>In Control</td>
</tr>
<tr>
<td>2013</td>
<td>3</td>
<td>1</td>
<td>-0.45</td>
<td>In Control</td>
</tr>
<tr>
<td>2013</td>
<td>2</td>
<td>1</td>
<td>-0.45</td>
<td>In Control</td>
</tr>
<tr>
<td>2013</td>
<td>1</td>
<td>2</td>
<td>0.55</td>
<td>In Control</td>
</tr>
<tr>
<td>2012</td>
<td>12</td>
<td>1</td>
<td>-0.45</td>
<td>In Control</td>
</tr>
<tr>
<td>2012</td>
<td>11</td>
<td>3</td>
<td>1.55</td>
<td>Out Of Control</td>
</tr>
</tbody>
</table>
Query 4: Monthly Session Process Control

Report Application:

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Count</th>
<th>Average</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>7</td>
<td>4</td>
<td>2.2</td>
<td>Out Of Control</td>
</tr>
<tr>
<td>2011</td>
<td>6</td>
<td>2</td>
<td>0.2</td>
<td>In Control</td>
</tr>
<tr>
<td>2011</td>
<td>5</td>
<td>2</td>
<td>0.2</td>
<td>In Control</td>
</tr>
</tbody>
</table>

Average # Sessions: 1.8
Standard Deviation: 0.89
Process Capability (Cpk): 2
Upper Control Limit: 3.58
Lower Control Limit: 0.02
Query 5: Optimal Class Location

Objective: Find the optimal location (zip code) for future group classes based on the number of current clients and prospective clients who have expressed interest in the topic in each zip code (Set Packing Problem)
Purpose: Strommen Tutoring would like to expand their group class services. Optimization of class location will increase the probability of high attendance (and profitability) by offering classes in areas with more interested people.
Query 5:
Optimal Class Location

Goal: Choose Best Zipcode – Topic pairs that Maximize # Interested Clients / Prospective Clients within a given radius

User Defined Parameters:
- Radius
- # of Classes
Query 5: Optimal Class Location

**SQL**

```sql
```
### Query 5: Optimal Class Location

#### Results (Access)

<table>
<thead>
<tr>
<th>ZipCode</th>
<th>Topic</th>
<th>Latitude</th>
<th>Longitude</th>
<th>NumberofClients</th>
</tr>
</thead>
<tbody>
<tr>
<td>90001</td>
<td>1</td>
<td>34</td>
<td>118</td>
<td>3</td>
</tr>
<tr>
<td>90001</td>
<td>2</td>
<td>34</td>
<td>118</td>
<td>2</td>
</tr>
<tr>
<td>93109</td>
<td>1</td>
<td>34.4</td>
<td>120</td>
<td>2</td>
</tr>
<tr>
<td>93109</td>
<td>2</td>
<td>34.4</td>
<td>120</td>
<td>2</td>
</tr>
<tr>
<td>93109</td>
<td>3</td>
<td>34.4</td>
<td>120</td>
<td>1</td>
</tr>
<tr>
<td>94709</td>
<td>1</td>
<td>37.9</td>
<td>122.3</td>
<td>1</td>
</tr>
<tr>
<td>90210</td>
<td>3</td>
<td>34</td>
<td>118.4</td>
<td>1</td>
</tr>
<tr>
<td>90210</td>
<td>1</td>
<td>34</td>
<td>118.4</td>
<td>1</td>
</tr>
<tr>
<td>94709</td>
<td>3</td>
<td>37.9</td>
<td>122.3</td>
<td>1</td>
</tr>
<tr>
<td>94709</td>
<td>2</td>
<td>37.9</td>
<td>122.3</td>
<td>1</td>
</tr>
<tr>
<td>33128</td>
<td>1</td>
<td>25.8</td>
<td>80</td>
<td>1</td>
</tr>
<tr>
<td>78207</td>
<td>3</td>
<td>29.4</td>
<td>98.5</td>
<td>1</td>
</tr>
<tr>
<td>78207</td>
<td>2</td>
<td>29.4</td>
<td>98.5</td>
<td>1</td>
</tr>
<tr>
<td>78207</td>
<td>1</td>
<td>29.4</td>
<td>98.5</td>
<td>1</td>
</tr>
<tr>
<td>70112</td>
<td>3</td>
<td>30</td>
<td>90</td>
<td>1</td>
</tr>
<tr>
<td>70112</td>
<td>2</td>
<td>30</td>
<td>90</td>
<td>1</td>
</tr>
<tr>
<td>70112</td>
<td>1</td>
<td>30</td>
<td>90</td>
<td>1</td>
</tr>
<tr>
<td>33128</td>
<td>2</td>
<td>25.8</td>
<td>80</td>
<td>1</td>
</tr>
<tr>
<td>90001</td>
<td>3</td>
<td>34</td>
<td>118</td>
<td>1</td>
</tr>
</tbody>
</table>
spherical distance formula.

Let \( d = \text{distance between } p_1 \text{ and } p_2 \), \( r = \text{Earth's radius} \), \( \varphi_s = \text{latitude of } p_1 \), \( \varphi_f = \text{latitude of } p_2 \), \( \lambda_s = \text{longitude of } p_1 \), \( \lambda_f = \text{longitude of } p_2 \)

Then \( d = 2r \arcsin\left( \sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_s \cos\varphi_f \sin^2\left(\frac{\Delta\lambda}{2}\right)} \right) \)

linear program variables.

**Input Variables:**
- Let \( C_{jk} \) = number of clients located in zipcode \( j \) who are interested in topic \( k \)
- Let \( M_c \) = global maximum number of classes
- Let \( M_R \) = the maximum distance between clients' zipcode and classes
- Let \((x_i, y_i)\) = the latitude and longitude of the zipcode
- Let \( d(x_i, y_i, x_j, y_j) \) = spherical distance between zipcode \( i \) and zipcode \( j \)

**Decision Variables:**
- Let \( S_{ij} = 1 \) if zipcode \( i \) is selected for topic \( k \) and \( S_{ij} = 0 \) otherwise
- Let \( R_{ijk} = 1 \) if zipcode \( j \) is in the radius of selected zipcode \( i \) for topic \( k \) and \( R_{ijk} = 0 \) otherwise
Query 5: Optimal Class Location

Linear program formulation.

Maximize \( \sum_{i,j,k} R_{ijk} * C_{jk} \)

Subject to

\( \sum_{i,k} S_{ik} \leq M_c \)

\( \sum_i R_{ijk} \leq 1 \ \forall j,k \)

\( R_{ijk} \leq S_{ik} \ \forall i,j,k \)

\( d(x_i,y_i,x_j,y_j) * S_{ik} \leq M_R \ \forall i,j,k \)

\( S_{ik}, R_{ik} \in \{0,1\} \ \forall i,j,k \)
Query 5:
Optimal Class Location

Formulation of LP in AMPL model file.

```AMPL
# PARAMETERS
param MaxClasses := 0;
param MaxRadius := 0;
param NumberOfStudents {i in Zipcodes, j in Topics} >= 0;  # Total number of students for each zipcode - topic combination
param Latitude {i in Zipcodes};
param Longitude {i in Zipcodes};

# DECISION VARIABLES
var ZipcodeTopic {i in Zipcodes, j in Topics} binary;  # These are the chosen Zipcodes and Topics
var ZipcodeInRadius {i in Zipcodes, j in Zipcodes, k in Topics} binary;  # These are the Zipcodes within range of the chosen Zipcode and Topic

# OBJECTIVE FUNCTION
maximize TotalClients: sum {i in Zipcodes, j in Zipcodes, k in Topics} ZipcodeInRadius[i,j,k] * NumberOfStudents[j,k];

# CONSTRAINTS
subject to NumberOfClasses: sum {i in Zipcodes, j in Topics} ZipcodeTopic[i,j] <= MaxClasses;
subject to ZipcodeOnlyOnce {j in Zipcodes, k in Topics}: sum {i in Zipcodes} ZipcodeInRadius[i,j,k] <= 1;
subject to LimitZipcodeInRadius {i in Zipcodes, j in Zipcodes, k in Topics}: ZipcodeInRadius[i,j,k] <= MaxRadius;
```

data file.

```AMPL
94708;

set Topics :=
Spanish
French
SATPrep;

# set the parameters
param MaxClasses := 2;  # number of new group classes to add
param MaxRadius := 1500;  # in miles

# create the data for the zipcodes, their distances, topic, and number of clients
param NumberOfStudents:

<table>
<thead>
<tr>
<th>Zipcode</th>
<th>Spanish</th>
<th>French</th>
<th>SATPrep</th>
</tr>
</thead>
<tbody>
<tr>
<td>33128</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>70112</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>78207</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>90001</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>90210</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>93109</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>94708</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

param:
<table>
<thead>
<tr>
<th>Zipcode</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>33128</td>
<td>25.8</td>
<td>80</td>
</tr>
<tr>
<td>70112</td>
<td>30</td>
<td>90</td>
</tr>
<tr>
<td>78207</td>
<td>29.4</td>
<td>98.5</td>
</tr>
<tr>
<td>90001</td>
<td>34</td>
<td>118</td>
</tr>
<tr>
<td>90210</td>
<td>34</td>
<td>118.4</td>
</tr>
<tr>
<td>93109</td>
<td>34.4</td>
<td>120</td>
</tr>
<tr>
<td>94708</td>
<td>37.9</td>
<td>122.3</td>
</tr>
</tbody>
</table>
```
Query 5: Optimal Class Location

Results (AMPL)

optimal solution output

```
MINOS 5.51: ignoring integrality of 24 variables
MINOS 5.51: optimal solution found.
3 iterations
objective 18
varname
1  "ZipcodeTopic[33128,'Spanish']"  0
2  "ZipcodeTopic[33128,'French']"  0
3  "ZipcodeTopic[33128,'SATPrep']"  0
4  "ZipcodeTopic[70112,'Spanish']"  0
5  "ZipcodeTopic[70112,'French']"  0
6  "ZipcodeTopic[70112,'SATPrep']"  0
7  "ZipcodeTopic[78207,'Spanish']"  0
8  "ZipcodeTopic[78207,'French']"  0
9  "ZipcodeTopic[78207,'SATPrep']"  0
10 "ZipcodeTopic[90001,'Spanish']"  0
11 "ZipcodeTopic[90001,'French']"  0
12 "ZipcodeTopic[90001,'SATPrep']"  0
13 "ZipcodeTopic[90210,'Spanish']"  0
14 "ZipcodeTopic[90210,'French']"  0
15 "ZipcodeTopic[90210,'SATPrep']"  0
16 "ZipcodeTopic[93109,'Spanish']"  0
```
user interface.
# Prospective Clients List

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Date Added</th>
<th>Birth Date</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nicole</td>
<td>Kessler</td>
<td>4/2/2012</td>
<td>5/5/1991</td>
<td><a href="mailto:xxRoxxygurlxx@gmail.com">xxRoxxygurlxx@gmail.com</a></td>
</tr>
<tr>
<td>Nikolaj</td>
<td>Lannister</td>
<td>9/26/2013</td>
<td>11/20/1972</td>
<td><a href="mailto:Nickll@gmail.com">Nickll@gmail.com</a></td>
</tr>
<tr>
<td>Derek</td>
<td>Rose</td>
<td>5/16/2013</td>
<td>5/22/1992</td>
<td><a href="mailto:Drosay@gmail.com">Drosay@gmail.com</a></td>
</tr>
<tr>
<td>Colbie</td>
<td>Brian</td>
<td>1/3/2013</td>
<td>1/7/1990</td>
<td><a href="mailto:ColBrian@yahoo.com">ColBrian@yahoo.com</a></td>
</tr>
<tr>
<td>Thomas</td>
<td>Bradie</td>
<td>11/15/2012</td>
<td>2/20/1988</td>
<td><a href="mailto:Tbradz@hotmail.com">Tbradz@hotmail.com</a></td>
</tr>
<tr>
<td>Andrew</td>
<td>Breeze</td>
<td>11/21/2012</td>
<td>11/22/1940</td>
<td><a href="mailto:BreezezeA@gmail.com">BreezezeA@gmail.com</a></td>
</tr>
</tbody>
</table>
## Report: Pending Transactions

<table>
<thead>
<tr>
<th>TransactionID</th>
<th>Amount</th>
<th>Open Date</th>
<th>Transaction Type</th>
<th>Days Since Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>82</td>
<td>$21.00</td>
<td>5/1/2011</td>
<td>Deposit</td>
<td>732</td>
</tr>
<tr>
<td>9</td>
<td>$22.00</td>
<td>5/1/2011</td>
<td>Deposit</td>
<td>732</td>
</tr>
<tr>
<td>12</td>
<td>$48.00</td>
<td>6/1/2011</td>
<td>Deposit</td>
<td>701</td>
</tr>
<tr>
<td>76</td>
<td>$46.00</td>
<td>6/1/2011</td>
<td>Deposit</td>
<td>701</td>
</tr>
<tr>
<td>50</td>
<td>$31.00</td>
<td>7/1/2011</td>
<td>Deposit</td>
<td>671</td>
</tr>
<tr>
<td>81</td>
<td>$45.00</td>
<td>8/1/2011</td>
<td>Deposit</td>
<td>640</td>
</tr>
<tr>
<td>49</td>
<td>$21.00</td>
<td>9/1/2011</td>
<td>Deposit</td>
<td>609</td>
</tr>
<tr>
<td>83</td>
<td>$45.00</td>
<td>9/1/2011</td>
<td>Deposit</td>
<td>609</td>
</tr>
<tr>
<td>15</td>
<td>$14.00</td>
<td>9/1/2011</td>
<td>Deposit</td>
<td>609</td>
</tr>
<tr>
<td>24</td>
<td>$13.00</td>
<td>10/1/2011</td>
<td>Deposit</td>
<td>579</td>
</tr>
<tr>
<td>Teacher Name</td>
<td>John Dorian</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number Of Sessions</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number Of Clients</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Revenue</td>
<td>$1,405.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revenue Per Client</td>
<td>$281</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retention Rate</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Recommended Bonus:** $102.15

**Teacher Performance Bonus**

**Amount**

$102.15

Create Bonus

Annual Report For 2012 - 2013
Form: New Person

New Person

First Name: Marlen
MI: E
Last Name: Knusel
Gender: Female
Birth Date: 10/28/1990
Person ID: 1
Address: 456 Prospect Ave.
Phone: 8058073596
Email: mknusel@berkeley.edu

Prospect

Prospect ID: 1
Date Added: 4/24/2013
Contact Method:
- Email
- Phone
- Website
- Employee
- Other

Contractor Prospect

Contractor Prospect ID: 2
Years Experience: 9
Company Name: Aspen Co.
Contractor Type: Technology
questions?