Team 4: Final Report for IEOR 115

Nicole Huxtable | Young Min Kim | Chaitanya Lall | Avi Sen | Jatin Raheja | Parth Rawat | Devansh Vaish | Srushti Vora

December 7, 2016
Table of Contents

Introduction

The Client 1
Express ECG Service Model 1
Current Database & Future Scenario 1

New Database Implementation

Enhanced Entity Relationship (EER) Diagram 2
MS Access Relationships 2
MS Access Forms 3
MS Access Reports 4

Relationship Schema 5

Normalization Analysis 7

Queries

Query 1: Service Cycle Time 10
Query 2: Reporting Doctor Distributions 11
Query 3: Reporting Doctor Efficiency 13
Query 4: Expansion Strategy 15
Query 5: Technician-Zone Efficiency 17

Discussion & Future Frameworks 18

Team Work Contributions 19
**Introduction**

**The Client:** Express ECG is a health-tech company based in India that specializes in remote diagnostic care. The aim of Express ECG is to provide electrocardiogram (ECG/EKG) tests in a swift manner that optimizes constraints of distance and time. The company looks to improve the accessibility of diagnostic services in rural and urban areas. Express ECG brings quick, affordable, sustainable, and accurate ECG services directly to the patient’s doorstep by using portable, server-based ECG devices. The ECG data can be accessed remotely, eliminating the need for an on-site doctor. Currently, Express ECG has services available in over 30 locations across India.

**Express ECG Service Model:** The company caters to direct patients, non-cardiac doctors, and hospitals. In every city/village, ECG technicians are assigned one or more geographical zones they serve. We also have a pool of online, reporting doctors who analyze the ECG data.

The procedure of an episode with Express ECG is as follows: Upon experiencing symptoms, the client will either submit an online request or call the call center for an ECG. After noting all the necessary information, the call center staff will contact one of the technicians inside the patient’s zone and the technician will go to the client’s address with the portable ECG kit. The technician will then perform the ECG tests and upload the data to a server. The reporting doctor looks at the data online and creates an ECG report that includes interpretation and further recommendations. The entire service cycle is promised to be within 30 min and cost around $3.50.

**Current Database & Future Scenario:** Express ECG currently functions as a not-for-profit entity and uses the MySQL platform to store data. The database records basic information on customers, employees, and ECG episodes. Given that the company is scaling and transitioning into a for-profit entity, this data does not provide much insight. Through the improved database, Express ECG can increase its efficiency and better manage its business. Our goal for the future is to expand the current database for the company so it can look at much more useful data; this would allow the firm to derive the most profitable market, analyze employee productivity, obtain more detailed information about the demand the firm faces, the consistency of each of the firm’s procedures, and their respective bottlenecks. Eventually, the data provided from the new database will allow the company to optimize the size and location of its workforce, eliminate costs associated with inefficiencies in the procedure and workforce, and even prioritize areas of expansion into more urban areas.
New Database Implementation

Enhanced Entity Relationship Diagram:

![EER Diagram](image1)

MS Access Relationships:

![MS Access Relationships](image2)
MS Access Forms:

Reporting Data Form

![Screenshot of MS Access Form: Form to add/edit reporting data](image1)

Reporting Doctor Form

![Screenshot of MS Access Form: Form to add/edit information about a reporting doctor](image2)
MS Access Reports:

Episode Data Report

<table>
<thead>
<tr>
<th>EpisodeID</th>
<th>ECG_Data_Uploaded_Time</th>
<th>ECG_Reported_Time</th>
<th>Reporting_Doctor_ID</th>
<th>Technician_ID</th>
<th>Timestamp</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11:10 PM</td>
<td>11:23:00 PM</td>
<td>1</td>
<td>1</td>
<td>12/31/1899</td>
<td>11:23:00 PM</td>
</tr>
<tr>
<td>2</td>
<td>10:05 PM</td>
<td>10:45:00 PM</td>
<td>2</td>
<td>2</td>
<td>12/31/1899</td>
<td>10:45:00 PM</td>
</tr>
<tr>
<td>3</td>
<td>9:07 PM</td>
<td>9:36:00 PM</td>
<td>3</td>
<td>1</td>
<td>12/31/1899</td>
<td>9:36:00 PM</td>
</tr>
<tr>
<td>4</td>
<td>8:30 PM</td>
<td>9:08:00 PM</td>
<td>4</td>
<td>1</td>
<td>1/1/1900</td>
<td>9:08:00 PM</td>
</tr>
<tr>
<td>5</td>
<td>8:15 PM</td>
<td>8:54:00 PM</td>
<td>5</td>
<td>1</td>
<td>1/2/1900</td>
<td>8:54:00 PM</td>
</tr>
<tr>
<td>6</td>
<td>8:27 PM</td>
<td>8:54:00 PM</td>
<td>6</td>
<td>1</td>
<td>1/4/1900</td>
<td>8:54:00 PM</td>
</tr>
<tr>
<td>7</td>
<td>8:03 PM</td>
<td>8:31:00 PM</td>
<td>7</td>
<td>1</td>
<td>1/3/1900</td>
<td>8:31:00 PM</td>
</tr>
<tr>
<td>8</td>
<td>7:51 PM</td>
<td>8:22:00 PM</td>
<td>8</td>
<td>1</td>
<td>1/4/1900</td>
<td>8:22:00 PM</td>
</tr>
<tr>
<td>9</td>
<td>7:46 PM</td>
<td>8:02:00 PM</td>
<td>9</td>
<td>2</td>
<td>1/1/1900</td>
<td>8:02:00 PM</td>
</tr>
<tr>
<td>10</td>
<td>7:23 PM</td>
<td>7:55:00 PM</td>
<td>10</td>
<td>2</td>
<td>1/2/1900</td>
<td>7:55:00 PM</td>
</tr>
<tr>
<td>11</td>
<td>7:20 PM</td>
<td>7:41:00 PM</td>
<td>11</td>
<td>2</td>
<td>1/1/1900</td>
<td>7:41:00 PM</td>
</tr>
<tr>
<td>12</td>
<td>7:08 PM</td>
<td>7:38:00 PM</td>
<td>12</td>
<td>2</td>
<td>1/4/1900</td>
<td>7:38:00 PM</td>
</tr>
<tr>
<td>13</td>
<td>7:01 PM</td>
<td>7:34:00 PM</td>
<td>13</td>
<td>2</td>
<td>1/3/1900</td>
<td>7:34:00 PM</td>
</tr>
<tr>
<td>14</td>
<td>6:12 PM</td>
<td>6:32:00 PM</td>
<td>14</td>
<td>3</td>
<td>1/1/1900</td>
<td>6:32:00 PM</td>
</tr>
<tr>
<td>15</td>
<td>5:40 PM</td>
<td>6:16:00 PM</td>
<td>15</td>
<td>3</td>
<td>12/31/1899</td>
<td>6:16:00 PM</td>
</tr>
<tr>
<td>16</td>
<td>5:42 PM</td>
<td>6:10:00 PM</td>
<td>16</td>
<td>3</td>
<td>1/1/1900</td>
<td>6:10:00 PM</td>
</tr>
<tr>
<td>17</td>
<td>5:48 PM</td>
<td>6:10:00 PM</td>
<td>17</td>
<td>3</td>
<td>1/2/1900</td>
<td>6:10:00 PM</td>
</tr>
<tr>
<td>18</td>
<td>5:22 PM</td>
<td>5:54:00 PM</td>
<td>18</td>
<td>3</td>
<td>1/3/1900</td>
<td>5:54:00 PM</td>
</tr>
<tr>
<td>19</td>
<td>5:25 PM</td>
<td>5:50:00 PM</td>
<td>19</td>
<td>3</td>
<td>1/4/1900</td>
<td>5:50:00 PM</td>
</tr>
<tr>
<td>20</td>
<td>5:27 PM</td>
<td>5:50:00 PM</td>
<td>20</td>
<td>3</td>
<td>1/5/1900</td>
<td>5:50:00 PM</td>
</tr>
</tbody>
</table>

Fig 4. Screenshot of MS Access Report: Report that lists the various ECG Episode Data

Customer-Employee Contact Report

<table>
<thead>
<tr>
<th>Employee_ID</th>
<th>Customer_ID</th>
<th>Report_ID</th>
<th>Timestamp</th>
<th>ECG_Reported_Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>1</td>
<td>11:10:00 PM</td>
<td>11:23:00 PM</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>2</td>
<td>10:05:00 PM</td>
<td>10:45:00 PM</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>3</td>
<td>9:07:00 PM</td>
<td>9:36:00 PM</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>4</td>
<td>8:30:00 PM</td>
<td>9:08:00 PM</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>5</td>
<td>8:15:00 PM</td>
<td>8:54:00 PM</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>6</td>
<td>8:27:00 PM</td>
<td>8:54:00 PM</td>
</tr>
<tr>
<td>6</td>
<td>19</td>
<td>7</td>
<td>8:03:00 PM</td>
<td>8:31:00 PM</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
<td>8</td>
<td>7:51:00 PM</td>
<td>8:22:00 PM</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>9</td>
<td>7:44:00 PM</td>
<td>8:02:00 PM</td>
</tr>
<tr>
<td>8</td>
<td>17</td>
<td>10</td>
<td>7:23:00 PM</td>
<td>7:55:00 PM</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>11</td>
<td>7:20:00 PM</td>
<td>7:41:00 PM</td>
</tr>
<tr>
<td>9</td>
<td>24</td>
<td>12</td>
<td>7:08:00 PM</td>
<td>7:38:00 PM</td>
</tr>
<tr>
<td>7</td>
<td>28</td>
<td>13</td>
<td>7:01:00 PM</td>
<td>7:34:00 PM</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
<td>14</td>
<td>6:12:00 PM</td>
<td>6:32:00 PM</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>15</td>
<td>5:40:00 PM</td>
<td>6:16:00 PM</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>16</td>
<td>5:42:00 PM</td>
<td>6:10:00 PM</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>17</td>
<td>5:48:00 PM</td>
<td>6:10:00 PM</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>18</td>
<td>5:22:00 PM</td>
<td>5:54:00 PM</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>19</td>
<td>5:25:00 PM</td>
<td>5:50:00 PM</td>
</tr>
</tbody>
</table>

Fig 5. Screenshot of MS Access Report: Report that lists the various Customer-Employee contact episodes
Relational Schema

Our relational schema is representative of our EER diagram.

**Strong Entities:**

1. **Employee** (Employee_ID, Fname, MI, Lname, Citizen_number, DOB, Home_Address, Zipcode, Gender, Contact_Number, Email, Job_Title, Highest_Education_Level, Joining_Date, Monthly_Salary_Amount, Yearly_Bonus_Amount)
   - Call_Center_Staff (Shift_Hours)
   - IT_Staff (Projects)
   - Management (Department)
   - Reporting_Doctor (Med_School_Graduation_Year, Year_Since_Reporting_ECG, Physician_Or_Cardiologist, Shift_Hours)
   - Other
2. **Customer** (Customer_ID, Contact_Number, Email)
   - Patient (Fname, MI, Lname, Family_Doctor_Name, DOB, Home_Address, Past_Cardiac_Issues, Family_Cardiac_Issues, Affiliated_Healthcare_Org, Referring_Doc_ID)
   - Hospital (Organization_Name, Hospital_Name, Representative_Name, Date_Since_Partnership, Speciality)
   - Rural_Clinics (Organization_Name, Representative_Name)
   - Referring_Doctor (Affiliated_Hospitals)
3. **Technician** (Technician_ID, Fname, MI, Lname, Citizen_Number, DOB, Home_Address, Gender, Contact_Number, Email, Joining_Date, ECG_Kit_ID, Default_Location)
4. **Transaction** (Transaction_ID, Date, Amount)
   - Revenue
     - Service_Revenue (Technician_ID)
     - Donation (Donating_Entity_Name)
     - Investment (Investor_Name)
   - Payment
     - Salary (Employee_ID)
   - Tax
5. **Equipment** (Equipment_ID, Equipment_Name)
   - Office_Equipment
   - ECG_Equipment (Manufacturing_Year)
6. **Reporting_Data** (Report_ID, Reporting_Doctor_ID, Reporting_Doctor_FName, Reporting_Doctor_LName, Date)
   - Patient_Data (ECG_Image, ECG_Measurements, Blood_Pressure, Symptoms, Vitals, Height, Weight, Gender, Age, Patient_ID, Patient_Fname, Patient_Lname)
   - Episode_Data (ECG_Data_Upload_Time, ECG_Reported_Time, Technician_ID, Technician_Fname, Technician_Lname)
7. **Zone** (Zone_ID, Zip_Code, No_of_clinics)
8. **Location** (Address, Apt/House_Number, Locality, Zip_Code, Zone_ID)
Weak Entities:

9. ECG_Report (Reporting_Doctor_ID\textsuperscript{1,4}, Report_ID\textsuperscript{5}, Timestamp, Customer_ID, Finding, Impression, Recommendation)
10. Maintenance (Equipment_ID\textsuperscript{5}, Service_Date)

N:M Relationships:

11. Uses (Employee_ID\textsuperscript{1}, Equipment_ID\textsuperscript{5})
12. Is_Contacted_By (Report_ID, Employee_ID\textsuperscript{1}, Customer_ID\textsuperscript{2}, Timestamp)
13. E_Contacts_T (Employee_ID\textsuperscript{1}, Technician_ID\textsuperscript{3}, Timestamp)
14. Records (Employee_ID\textsuperscript{1}, Address\textsuperscript{8})
15. C_Contacts_C (Customer_ID\textsuperscript{2}, Employee_ID\textsuperscript{1}, Time_interval)
16. Associates (Hospital_ID\textsuperscript{2,2}, Patient_ID\textsuperscript{2,2})
17. Conducts (Customer_ID\textsuperscript{2}, Technician_ID\textsuperscript{3})
18. C_Contacts_T (Technician_ID\textsuperscript{3}, Customer_ID\textsuperscript{2})
19. Zone_Assign (Zone_ID\textsuperscript{7}, Technician_ID\textsuperscript{3})
20. Generates_Rev (Technician_ID\textsuperscript{5}, Transaction_ID\textsuperscript{4})
21. Maintains (Equipment_ID\textsuperscript{5}, Date)
## Normalization Analysis

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Functional Dependencies</th>
<th>Current NF</th>
</tr>
</thead>
</table>
| 1 Zone (Zone_ID, Zip_Code, No_of_clinics) | - Zip_Code→Zone_ID  
- Zone_ID→No_of_clinics                                                                 | 1NF        |
| 2 Location (Address, Apt/House_Number, Locality, Zip_Code, Zone_ID) | - Address→Locality  
- Zip_Code→Zone_ID                                                                 | 3NF        |
- Technician_ID→{Tech_Fname, Tech_Lname}  
- Report_ID→{Date, ECG_Data_Upload_Time, ECG_Reported_Time, ReportDoc_ID, Technician_ID}                                                                 | 2NF        |
| 4 ECG_Equipment (Equipment_ID, Equipment_Name, Manufacturing_Year) | - Equipment_ID→Equipment_Name                                                            | BCNF       |
- Patient_ID→{Patient_Fname, Patient_Lname, DOB, Gender}  
- DOB, Date→Age  
- ECG_Image→{ECG_Measurements}  
- Report_ID→ReportDoc_ID, Date, Patient_ID, ECG_Image, ECG_Measurements, Blood_Pressure, Symptom, Vitals, Height, Weight | None       |

### Relationship #1: Zone (Zone_ID, Zip_Code, No_of_clinics)

This relationship is in 1NF but not in 2NF, therefore it is not completely normalized. Zone_ID, which is a proper subset of the candidate key determines No_of_Clinics, which is a non-prime attribute. To normalize:

- Zone ("")
- ZoneClinic (Zone_ID, No_of_clinics)
- ZipZone(Zip_Code, Zone_ID)

The new relationship is in 3NF since for all FDs X Y either (1) X is a superkey or (2) Y is a prime-attribute. The new relationship is also in BCNF since for all FDs X Y, X is a superkey.
Relationship #2: Location \{Address, Apt/House_Number, Locality, Zip_Code, Zone_ID\}

By itself, this relationship is not completely normalized because it is not in BCNF. It is in 3NF because in the functional dependency, Zip_Code→ Zone_ID, even though Zip_Code is not a super key, Zone_ID is a prime attribute (because it is a foreign key). This issue is, however, resolved by the implementation of the above normalization. Nonetheless, the normalization that makes this BCNF is:

\[
\text{Location ("""
} \rightarrow
\begin{itemize}
  \item \text{Add (Address, Apt/House_Number, Locality, Zip_Code, Zone_ID)}
  \item \text{ZipZone(Zip_Code, Zone_ID)}
\end{itemize}
\]

Relationship #3: Episode_Data
\{Report_ID, ReportDoc_ID\^{1,4}, ReportDoc_Fname, ReportDoc_Lname, Technician_ID\^{3}, Tech_Fname, Tech_Lname, Date, ECG_DataUpload_Time, ECG_Reported_Time\}

By itself, this relationship is not completely normalized because it is not in 3NF and BCNF. For the FDs related to Technician_ID and ReportDoc_ID, X→Y in this relationship, neither X is a super key of the relationship nor Y is a prime attribute. So:

\[
\text{Episode_Data ("""
} \rightarrow
\begin{itemize}
  \item \text{EpisodeD (Report_ID, ReportDoc_ID, Technician_ID, Date, ECG_DataUpload_Time, ECG_Reported_Time)}
\end{itemize}
\]

We don’t need to make new relation for Reporting Doctor and Technician since they already exist.

Relationship #4: ECG_Equipment \{Equipment_ID, Equipment_Name, Manufacturing_Year\}

This relationship is in BCNF because for all FDs X→Y, X is a super key.
Relationship #5: Patient_Data (Report_ID, ReportDoc_ID^{1,4}, ReportDoc_Fname, ReportDoc_Lname, Timestamp, Date, ECG_Image, ECG_Measurements, Blood_Pressure, Symptom, Vitals, Height, Weight, Gender, DOB, Age, Patient_ID^{2,1}, Patient_Fname, Patient_Lname)

This relationship is not in 1NF because this relationship has multi-valued attributes.

This relationship is in 2NF since no proper subset of the CK defines a non-prime attribute. However, it is not in 3NF since for the FDs X → Y pertaining to Reporting Doctor, Patient, DOB, and ECG image, neither X is a super key nor Y is a prime attribute. The new relationships are in BCNF as well and are as follows:

- PatientD (Report_ID, Reporting_Doctor_ID^{1,4}, Reporting_Doctor_Fname, Reporting_Doctor_Lname, Timestamp, Date, ECG_Image, Blood_Pressure, Symptom, Height, Weight, Gender, DOB, Age, Patient_ID^{2,1}, Patient_Fname, Patient_Lname)
- Vitals (Report_ID, Vitals, HR, Pulse, BP, Temp)
- ECG_M (Report_ID, ECG_Measurements, PR, ST, R-R, QT, Qtc, QRS, R, QRs, T)

- PatientD ("""")
- Vitals ("""")
- ECG_M ("""")
Query 1: Service Cycle Time

**Question:**
Isolate cycle times over 30 minutes, including the employees connected to each event.

**Business Justification:**
In order to fulfill Express ECG’s promise to complete entire cycles within 30 minutes, it is important to find outliers in terms of service cycle time so that any issues can be addressed. Specifically, this query will allow Express ECG to locate any bottlenecks and inspect possible reasons for it, retrain technicians who cause multiple errors, and improve overall average delivery time.

**SQL Code - Finding the Outliers:**

```sql
FROM Episode_Data e, Is_Contacted_By i
WHERE e.Report_ID = i.Report_ID
HAVING Service_Cycle_Time > 30;
```

The SQL code isolates events where the difference in minutes between the time the employee is contacted (i.Timestamp) and the time the technician uploads data from the episode (e.ECG_Report_Time) is greater than 30 minutes.

**Access Implementation:**

With these results, we can identify individuals who are acting as bottlenecks for the service cycle. Here, we may want to take a closer look at the performance of technicians with the IDs 1, 2, 3, and 4, as well as reporting doctors with the IDs 1, 2, and 3 since they have gone over 30 minutes on multiple occasions.
Query 2: Reporting Doctor Distributions

Question:
For each reporting doctor, plot the distribution for the time taken to generate an ECG report after receiving reporting data. How can these visuals help our client monitor reporting doctor performance and improve efficiency?

Once again, because our client promises to complete the service cycle within 30 mins, it is imperative that all players in the cycle complete their tasks in a timely manner. The reporting doctors’ performance is distinct because there are not variables that would consistently cause them to generate ECG reports slower or faster than usual; each time the data is received, in addition to keeping the time they take to generate the reports relatively low, they should generate reports with minimal variance of this time. In summary, reporting doctors must be efficient and reliable.

In order to visualize each reporting doctor’s reporting time distribution, we will use box plots (aka box and whisker plots). This type of plot is particularly helpful because it will allow our client to analyze each reporting doctor’s performance and consistency in relation to one another.

Business Justification:
By implementing this distribution analysis, our client will be able to:
- incentivize reporting doctors who are most efficient/reliable
- cut reporting doctors who are taking too long/are too inconsistent
- improve service cycle time

Procedure:
The box plots can be created at any time by through the following steps:

1. Run the following SQL Query in our Access database:
   ```sql
   SELECT Reporting_Doctor_ID,
   Report_ID,
   DATEDIFF("n", ECG_Data_Upload_Time, ECG_Reported_Time) AS Report_Creation_Time
   FROM Reporting_Data
   ORDER BY Reporting_Doctor_ID;
   ```
   Table 2A shows the SQL query output.

2. In the Access database, under the “EXTERNAL DATA” tab, export the table generated by the query to an Excel workbook by clicking the “Export to Excel spreadsheet” button:

3. Save the Excel spreadsheet as a CSV
4. Run the **R code** to the right, replacing the pathname in the first line with the location of the location of the CSV file.

The R code above would output plot in Fig 2A.

Analysis of the box plots provides our client with valuable information regarding each reporting doctor’s performance; more specifically, the box plot shows each reporting doctor’s median (a good representation of average when there are many independent data points) reporting time and the four quartiles of reporting time which collectively reveal each reporting doctor’s consistency in reporting time.

**Demonstration:**
Because our hardcoded database was limited in size and might not necessarily reflect the real world performance of a reporting doctor, we will show a boxplot that will more closely resemble what our client might see using artificial/synthetic data.

Because the performance of a reporting doctor can be summarized by their average reporting time and the variance in their reporting times, we will use the Normal Distribution (whose parameters are mean and variance) to generate reporting times for 5 different doctors, each with distinct reporting behavior.

We can generate the data in R using the “rnorm” function which outputs a vector of normally distributed numbers based on a user-determined mean and standard deviation. After generating these artificial reporting times, we can use essentially use the same code as before to create box plots.

The first code to the right creates the vectors necessary to model the client’s possible data. Note that for each artificial reporting doctor, the “rnorm” function is given different values for the “mean” and “sd” (standard deviation; variance is the standard deviation squared) parameters so their reporting behavior is distinct. The final line corrects all values less than 1.5 (we decided that 1.5 minutes is the absolute fastest an ECG report can be generated).

The code to the right puts the vectors into a table and creates box plots. The implementation of the code can be seen in Fig 2B. on the following page.
From this visual distribution, the client can note that the reporting doctor 2 is the model reporting doctor as he/she has the lowest median and a very low variance. Reporting doctors 1, 3, and 5 are significantly more inconsistent than 2 and 4. It seems as if 1 and 3 are objectively better than 5.

So, who should be cut? With these box plots, each doctor’s reporting time distribution is easily compared and it is ultimately up to the client to determine how these distributions should be weighted. For example, although 4 has a higher average than both 1 and 5, 4 is much more consistent; depending on the amount of risk our client wants to take on (probably not a lot), it might make more sense to keep 1 over 4 or vice versa.

**Query 3: Reporting Doctor Rating**

**Question:** Create a fair criteria to evaluate the performance of reporting doctors, and give them a rating score.

With the expansion of the company, more doctors will need to be hired to analyze the ECG data and report it back to the patients. As a result, comparing and contrasting the performance of the doctors will get progressively harder.

In this query, we took inspiration from the IMDb formula for rating films, and came up with a formula to rate the reporting doctors on a scale of 10. Our formula uses weighted averages of key attributes, ensuring a well-rounded rating criteria based on consistency, efficiency and experience.

**Business Justification:**

These ratings help our client to:

- Rank each doctor by performance
- Use the data to weed out underperforming employees, thus eliminating possible bottlenecks.
- Form hiring/firing strategy
- Reward and promote the high performers

**Formula:** Rating = \( [(v/(v+m)) \times R] + [(m/(v+m)) \times c] \)

Where:

- \( R \) = Average time for each doctor to report
- \( V \) = Number or reports made by the doctor
- \( M \) = Minimum number of reports required to be considered (150)
- \( C \) = Mean time taken to report by all doctors
**SQL- Time Generation:**

```sql
CREATE VIEW [R] AS
SELECT avg (d.ECG_Reported_Time - d.ECG_Data_Upload_Time) as Average
FROM ECG_Report as e and Episode_Data as d

SELECT e. Reporting_Doctor_ID as Doctor ID,
     ((e.count(e.Report_ID)/(e.count(e.Report_ID)+150))*R.Average) +
     ((150/(e.count(e.Report_ID)+150))*(avg (d.ECG_Reported_Time - d.ECG_Data_Upload_Time)))
     as Average Time
FROM R, ECG_Report as e and Episode_Data as d
GROUP BY e.Reporting_Doctor_ID
ORDER BY Average Time;
```

**Implementation:**

<table>
<thead>
<tr>
<th>Doctor Name</th>
<th>Average Time Taken (in minutes)</th>
<th>Number of Reports Created</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deepak Vora</td>
<td>8.5</td>
<td>503</td>
<td>5.61</td>
</tr>
<tr>
<td>Alisha Vora</td>
<td>9</td>
<td>415</td>
<td>5.40</td>
</tr>
<tr>
<td>Sameer Dani</td>
<td>10</td>
<td>640</td>
<td>5.02</td>
</tr>
<tr>
<td>Anish Malhotra</td>
<td>14</td>
<td>357</td>
<td>3.64</td>
</tr>
<tr>
<td>Darshan Shah</td>
<td>8</td>
<td>167</td>
<td>5.57</td>
</tr>
</tbody>
</table>

Table 3A. Implementation of IMdB formula method

**Analysis:**

Since the formula takes a weighted average of multiple key attributes, it generates a rating not limited to just average time taken to report. As we can see from the first and last doctors in the table, the doctor with the minimum average time taken is not necessarily the one with the best rating.
Query 4: Expansion Strategy

Question:
Within a new district, find the 2 optimal locations to house our technicians. Generate types of locations to expand by analyzing previous demographics.

Since the company is still growing and looking for potential ways and districts to expand to we decided to try and analyze potential options for them. To do so we decided to gather data about the average call density from different locations and then narrow down the potential expansion options. After finding a zone, we wanted to find the 2 most optimal places to house the technicians so that their average demand-weighted travel time is minimized.

Once we got the call density data, we narrowed down each district into several small zones and based on average call density per day we gave each zone a weightage equivalent to the same. Next, we estimated the travel time between zones using Google Maps and created a demand-weighted transportation problem.

Finally, we used the greedy algorithm to solve the problem and find the 2 optimal locations within the zone to house the technicians and hence, maximize service efficiency as well as minimize technician travel time which would lead to reduction in overall service cycle time.

Business Justifications:
The key business justifications for this are:
- Drive marketing strategy towards specific clientele
- Determine where expansion is optimal- financially and operationally
- Minimize cost and travel time

SQL CODE- Demographics:
SELECT z.Zone_ID, z. No_of_clinics, count(r.customer_ID), avg(Today()-c.DOB), count(t.Technician_ID) FROM Zone as z, Customer as c, ECG_Report as r, Technician as t GROUP BY z.Zone_ID;

Further Analysis:
The Transportation Problem - About the greedy algorithm

The p-median problem is a specific type of a discrete location model. In this model, we wish to place p facilities to minimize the (demand-weighted) average distance between a demand node and the location in which a facility was placed.

In this model, there are no capacity constraints at the facilities.

The idea is to begin with a greedy placement of the p facilities in the first stage of the algorithm, and then to refine the placement of the facilities within neighborhoods in the second stage of the algorithm.
The first stage is:
1. Place the first facility using brute force enumeration to solve the 1-median problem;
2. For \( i = 2, \ldots, p \)
   (a) Keeping the location of already placed facilities fixed, place another facility to minimize \( \sum_{j \in J} \sum_{i \in I} h_{i,j}Y_{i,j} \).

The second stage is:
1. Find the neighborhood of each facility (meaning an assignment of demand nodes to each facility, such that the distance between a demand node and facility is minimum)
2. Do
   (a) Solve the 1-median problem in each neighborhood;
   (b) Find the neighborhood of each facility
3. While the neighborhoods have changed from the previous iteration

Our Scenario:

![Transportation Problem visualized at a potential location](image)

Steps:
(a) Solve for the 1-median problem. Locate 1st facility at B, with total travel distance 557.5
(b) Fix 1st facility at B; Compute total travel distance with 2nd facility opened at A, C, D, . . . , G. Locate 2nd facility at C. With B, C opened, total travel distance is 303.0
(c) Assign neighbors for B and C: \{A, C, D, F, G\} \rightarrow C and \{E, B\} \rightarrow E.
(d) Solve 1-median problem in each neighbor: i. In \{A, C, D, F, G\}, locate facility at C ii. In \{E, B\}, locate facility at E since E has larger demand
(e) Check whether there are any changes in the neighborhood and we realize that G is reassigned to E.
(f) Rerun Step 2 and we recognize that the termination condition is met and the final solution is: Location facilities at C and E with \{A, C, D, F\} \rightarrow C and \{B, E, G\} \rightarrow E.

Hence, all demand is met.
Query 5: Technician-Zone Efficiency

Question: Find the optimal number of technicians to hire for each zone to meet the projected demand.

Business Justification:
This query helps in determining the demographic in each zone, and hence forecasting demand for that zone. The number of clinics and technicians can determine the existing supply, hence helping derive expansion and resource allocation strategy. Apropos Query IV, where we ascertained the 2 ideal locations to set-up shop for new locations for technicians - by determining optimal number of technicians required on any given day each location, C and E, we minimize cost & travel time, which is essential to the success.

SQL Code- Projecting Demand
The SQL Code enables us to retrieve the time each technician takes to go reach a customer from each of the new locations and the total number of customers in each zone. Once we receive the SQL data, we need to derive the ideal number of technicians in each location C and E, to minimize total time all technicians take in a single day to reach all customers.

```
SELECT   z.Zone_ID, c.Time_interval, count(c.Customer_ID), count(c.Employee_ID)
FROM     Zone z, C_Contacts_C c, Location L, Customer Cu
WHERE    L.Zone_ID=z.Zone_ID
AND      L.Address=Cu.Home_Address
AND      Cu.Customer_ID=c.Customer_ID
GROUP BY c.Time_interval, z.Zone_ID;
```

Analysis of ideal distribution of technicians
We need to derive the ideal number of technicians in each location C and E, to minimize total time all technicians take in a single day to reach all customers. Since we have all the data for the number of customers in each zone and time each technician takes to reach each zone from the new locations, we can run a linear equation to minimize total time of travel from the new locations to each customer in a day.

Variables:
a: Technicians traveling from C to A
b: Technicians traveling from C to B
...
h: Technicians traveling from E to A
i: Technicians traveling from E to B
...
n: Technicians traveling from E to G
Hence, The number of technicians at C= a+b+c+d+e+f+g
And number of technicians at E= h+i+j+k+l+m+n
Running the Linear Equation in AMPL and subjecting it to non-negative and supply=demand constraints:

\[
\text{minimize } P: 32^t + 39^t + 50^d + 26^e + 62^f + 39^g + 42^h + 13^i + 59^j + 40^k + 38^m + 16^n; \\
\text{subject to } x: a-b = 10; \\
\text{subject to } y: b = 20; \\
\text{subject to } z: c = 12; \\
\text{subject to } A: d + e \leq 4.5; \\
\text{subject to } B: i = 25; \\
\text{subject to } C: f + g = 7.5; \\
\text{subject to } D: h + m = 7.5; \\
\]

We get:

\[
\begin{align*}
\text{output} & = \{\text{MINOS 5.51: optimal solution found.} \} \\
& \quad \text{4 iterations, objective 13R2} \\
a & = 10 \\
b & = 20 \\
c & = 12 \\
d & = 0 \\
e & = 0 \\
f & = 0 \\
g & = 0 \\
h & = 0 \\
i & = 0 \\
j & = 0 \\
k & = 4.5 \\
l & = 25 \\
m & = 8 \\
n & = 7.5
\end{align*}
\]

Hence,

\[
a + b + c + d + e + f + g = 42 \text{ (# of technician trips needed at C)} \\
h + i + j + k + l + m + n = 45 \text{ (# of technician trips needed at E)}
\]

Discussion & Future Framework

Throughout the project, our group ran into multiple challenges that required both quantitative analysis and creativity. Due to the sensitive nature of the data, our team did not have authority to utilize the actual client data stored on the existing database. To circumvent this obstacle, our group made up and entered data to replicate possible client data; however, considering how many relationships were present in the database, it was often challenging to come up with enough data points to produce satisfactory outputs to our linear model, IMBB ranking and Greedy algorithm. Furthermore, due to the overall challenges from translation and lack of a structured database, it was extremely challenging to write the SQL code for a database which we made, because often, we did not realize the need for a new attribute or entity, until we actually wrote out the SQL and realized there was not enough input to satisfy the query.

Despite the challenges the group faced, our team managed to come up with many insights through our analysis. By utilizing R, we were able to come up with numerous data points to overcome our challenge with lack of available data points. This allowed us to gain a better understanding of how to analyze possible lack of efficiencies in the operations of reporting doctors. In addition, our team explored ways to increase service cycle efficiencies by developing methods to rank reporting doctor based off performance and identifying employees who fail to complete their duties within an acceptable timeframe. Furthermore, the group explored ways to allow Express ECG to achieve scale in the most efficient manner possible. By analyzing projected demands in each zone, we utilized linear programming and AMPL to construct a SQL query that ideally allocated new technician hires throughout the zones of coverage.

Looking forward, Express ECG wants to be a for-profit company and expand to urban locations. To make this development, Express ECG has the opportunity to utilize its new database to create queries that will provide insightful data that it can use to optimize its profit and efficiency as well as strategize expansion. Currently, Express ECG has a demand of 400 ECG tests per month, but as it looks to grow, it is projecting a much larger demand of roughly 2100 ECG tests per month. Additionally, as the company grows, it is also looking to increase its workforce from a little over 20 people to 220 people. This expansion will take time and calculation, and the new database we have created can play a crucial role in it.
Team Member Contributions

**Srushti Vora:** Srushti acted as the CEO of the project wherein she planned and attended all meetings and delegated tasks. She also was the point person for everything related to our client as she is actively involved in the company herself. In addition to presenting at DP Reviews, she led/helped create and update the following:

- Helped create and updated the EER diagram
- Led the creation of the Relational Schema
- Came up with the query questions, justifications, and SQL code in all 3 rounds
- Created the MS Access database to mimic the original database
- Performed all Normalization Analysis
- Compiled, created, and reviewed reports for DP Review III and final summary

**Parth Rawat:** Parth was very enthusiastic and present at nearly all meetings. He was very responsive to the group and worked well with everyone else. He worked on the following portions of the project:

- Creating cardinality constraints for the EER
- Aided in the development of the relation Schema
- Helped come up with first round of potential queries and why each would be useful
- Implementation of Access Database
- Reviewing and editing paper reports
- Compiling and creating PowerPoint Presentations

**Young Min Kim:** Young Min acted as the COO and assisted in the planning, preparation, and finalization of all deliverables for presentation. In addition to presenting at DP Reviews, Young Min contributed to the group by:

- Creating and maintaining relationship schemas
- Managing the Microsoft Access database
- Designing and creating PowerPoint presentations for all DP Reviews
- Brainstormed queries and justifications
- Edited reports and all other deliverables for submission

**Jatin Raheja:** Jatin acted as CCO to make sure all meetings were held on schedule and ensured meeting time worked for most people. Apart from this he presented at DP Reviews and worked on the following:

- Found the software ‘lucidchart’ and also helped design the final EER diagram on it
- Worked on MS access to create the Relationships and link all tables
- Helped come up with multiple queries and their justifications
- Wrote the code and explanation behind Query 4
- Helped come up with idea to use a Linear Program for Query 5

**Nicole Huxtable:** Nicole assisted in various roles throughout the project. She not only presented at DP reviews, but also helped create:

- Current database model & potential benefits
- Relational Schema
- Query 1
Avi Sen: Avi was a general member of the Express ECG team and took advantage of his non-leadership role to participate and contribute in as many facets of the project as possible. Avi was present at all DP reviews and presented information at each of them. In addition, Avi:

- Helped design the EER diagram
- Helped brainstorm query questions, their justifications, and SQL
- Wrote SQL for queries 1 and 2
- Implemented and debugged queries in Access
- Wrote report for query 2
- Presented Access implementation and query 2
- Reviewed Final Report

Chaitanya Lall: Chaitanya was the COO and helped compile work from all team members and assisted with various roles throughout the project. In addition to presenting at DP reviews, Chaitanya helped:

- Researched the company to create the Relationships and Entities of the EER diagram on Lucid Chart
- Redefined the EER Diagram and added Cardinality
- Created 3 Queries, coded 2 in SQL, solved 3 and presented 1

Devansh Vaish: Devansh took active part in discussion through the entire project, and worked on the delegated tasks. Additionally, he also helped compile and edit DP Review 1, 3, and present reviews 1 and 3. As part of his tasks, Devanish helped create and update the following:

- EER Diagram
- Relational Schema
- Query Questions, Justifications
- SQL Codes
- MS Access Database
- Query 3 Compilation