CNIT 37200 Final Report

Youtube Statistics Analysis

Yooto Joo
Adam Hafez
Stewart
Minseong Jeong

Project Background:

For our final project we researched and organized data relating to youtube statistics. Our initial dataset was unfiltered, disorganized, and contained unnecessary records. Our goal for this project was to learn and employ foundational techniques of database administration to design, implement, and optimize a robust and scalable database that effectively manages and organizes data.

Before we created our database we first needed to determine what records and columns were unnecessary for the project. More than half of the columns in the dataset were insignificant, and therefore we removed them from the dataset. A key component of optimization within a database is ensuring that only significant data is stored and managed. The process of filtering also involves cleaning the data. Before we insert data into a database, we need to ensure that all data is correctly structured and formatted. This includes, removing duplicates, fixing character strings, and configuring the database. Through employing the necessary techniques of a database administrator we as a group were able to successfully create a relational database that contained all relevant and significant data from the statistical dataset.

Database Description:

VideoInfo Table	CREATE TABLE VideoInfo (
	    videoID VARCHAR2(255) PRIMARY KEY,
	    title VARCHAR2(255),
	    uploadDate DATE,
	    viewCount INT
	);
	
	The primary key for the table is videoID, the data type is varchar as the ID's contain special characters.
	
	Title is a column containing the names of the videos.
	
	uploadDate is a column containing the date in which the video was uploaded.
	
	viewCount contains the number of views for the video.

CommentInfo Table	CREATE TABLE CommentInfo (
	    CommentID INT PRIMARY KEY,
	    videoID VARCHAR2(255) REFERENCES VideoInfo(videoID),
	    Comments VARCHAR2(1000),
	    CommentLikes INT
	);

	The primary key for the table is CommentID, the data type is an integer. VideoID is referenced as the foreign key. Comments is a column that contains the description of each comment. CommentLikes is a column that contains the number of likes for each comment.

EngagementMetrics Table	CREATE TABLE EngagementMetrics (
	    metricID INT PRIMARY KEY,
	    videoID VARCHAR2(255) REFERENCES VideoInfo(videoID),
	    title VARCHAR2(255),
	    likes INT,
	    comments INT
	);
	The primary key for the table is metricID.
	VideoID is referenced as the foreign key.
	Title is the column containing the name of the video.
	Likes contain the number of likes the video received.
	Comments contains the number of comments the video has.

Questions:

Our questions gain insight into crucial statistics, keywords, and characteristics of popular videos for tailoring content to audience preferences, maximizing viewer engagement, and ultimately growing the channel.

<table>
<tr>
<td>Question #1</td>
<td>

How would you retrieve the top 3 videos with the most comments?

Retrieve the top 3 videos with the most comments. SELECT the videoID and get the SUM of comments. Only output the first three rows of the query. You must use the FETCH statement.

Code:

```
SELECT
   v.VIDEOID,
   COUNT(c.COMMENTID) AS comment_total
FROM
   VideoInfo v
JOIN
   CommentInfo c ON v.VIDEOID = c.VIDEOID
GROUP BY
   v.VIDEOID
ORDER BY
   comment_total DESC
FETCH FIRST 3 ROWS ONLY;
```

Output:

```
ZgeorpjGJC0  20
LeC5yJq4tsI   20
96mrgd8-3yE  18
```

</td>
</tr>
<tr>
<td>Question #2</td>
<td>

How would you retrieve the average view count of all videos and output them in descending order?

</td>
</tr>
</table>

Retrieve the average view count of all videos and put them in descending order.
Add error handling for null values for viewcount, and output all records side by side.

Code:

```
DECLARE
   CURSOR video_avg_view_cursor IS
      SELECT videoID, ROUND(AVG(NVL(viewCount, 0)), 2) AS
averageViewCount
      FROM VideoInfo
      GROUP BY videoID
      ORDER BY averageViewCount DESC;
   video_id VideoInfo.VIDEOID%TYPE;
   avg_view_count NUMBER;
BEGIN
   FOR video_avg_view_rec IN video_avg_view_cursor LOOP
      video_id := video_avg_view_rec.videoID;
      avg_view_count := video_avg_view_rec.averageViewCount;
      DBMS_OUTPUT.PUT_LINE('Video ID: ' || video_id || ' | Average View
Count: ' || avg_view_count);
   END LOOP;
END;
/
```

Output:

```
Video ID: FzG4uDgje3M | Average View Count: 4034122271
Video ID: gCYcHz2k5x0 | Average View Count: 1582262997
Video ID: XXYlFuWEuKI | Average View Count: 915457091
Video ID: qpgTC9MDx1o | Average View Count: 826423766
Video ID: jJPMnTXl63E | Average View Count: 524709805
Video ID: yjmp8CoZBIo | Average View Count: 434352213
Video ID: Ct6BUPvE2sM | Average View Count: 425478119
Video ID: mRD0-GxqHVo | Average View Count: 378164492
Video ID: Ha80ZaecGkQ | Average View Count: 321977550
Video ID: nhBorPm6JjQ | Average View Count: 308501014
Video ID: NvR60Wg9R7Q | Average View Count: 302125099
Video ID: 0e3GPea1Tyg | Average View Count: 285526909
Video ID: fyIcQ1Xl-rs | Average View Count: 250793774
Video ID: EqboAI-Vk-U | Average View Count: 239207161
Video ID: nCg3ufihKyU | Average View Count: 208293677
Video ID: 7BJ3ZXpserc | Average View Count: 193685278
```

<table>
<tr>
<td></td>
<td>

```
Video ID: 9bqk6ZUsKyA | Average View Count: 191988678
Video ID: zxYjTTXc-J8 | Average View Count: 180766617
Video ID: h4UqMyldS7Q | Average View Count: 168546247
Video ID: XnitQYkYYcw | Average View Count: 161103805
Video ID: QxGVgXf_LNk | Average View Count: 160818375
Video ID: F4Y3Pkn95GI | Average View Count: 156000008
Video ID: 4-43lLKaqBQ | Average View Count: 153478497
Video ID: xuCn8ux2gbs | Average View Count: 147339243
Video ID: ul1H_p_FeaA | Average View Count: 145167323
Video ID: erQ_9yEz0ls | Average View Count: 136439525
Video ID: A-vX1AGBGsc | Average View Count: 134377650
Video ID: 3jS_yEK8qVI | Average View Count: 125670049
Video ID: =-EjsCBHEbbk | Average View Count: 122994457
Video ID: dg2Ag3e8W-Q | Average View Count: 116842899
Video ID: LeYsRMZFUq0 | Average View Count: 110124989
Video ID: xRwy_rKc7gI | Average View Count: 107126467
```

</td>
</tr>
<tr>
<td>Question #3</td>
<td>

How would you find the 10 videos with the lowest amount of comments?

The 10 videos with the lowest amount of comments. Use FETCH and LOOP.

Code:

```
DECLARE
   CURSOR comment_count_cursor IS
      SELECT c.videoID, v.title, COUNT(c.CommentID) AS commentCount
      FROM CommentInfo c
      JOIN VideoInfo v ON c.videoID = v.videoID
      GROUP BY c.videoID, v.title
      ORDER BY commentCount ASC
      FETCH FIRST 10 ROWS ONLY;
   video_id CommentInfo.VIDEOID%TYPE;
   title VideoInfo.TITLE%TYPE;
   comment_count NUMBER;
BEGIN
   FOR comment_count_rec IN comment_count_cursor LOOP
      video_id := comment_count_rec.videoID;
      title := comment_count_rec.title;
      comment_count := comment_count_rec.commentCount;
      DBMS_OUTPUT.PUT_LINE('Video ID: ' || video_id || ' | Title: ' || title || '
| Comment Count: ' || comment_count);
   END LOOP;
END;
```

</td>
</tr>
</table>

```
/
```

Output:

```
Video ID: voU9-39bicg | Title:
¿¿¿¿¿¿¿¿¿¿¿¿:¿¿¿¿¿¿¿¿¿¿¿¿/¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿/¿¿¿¿¿¿¿¿¿¿¿¿¿
¿¿¿/¿¿VS¿¿/¿¿¿¿¿¿¿¿¿/¿¿¿¿¿¿ ¿¿(¿¿¿NEWSLIVE) | Comment Count: 1
Video ID: 0vEdNPoeemQ | Title: Tech firms can&#39;t hide behind
advertising rules with financial products, says CFPB chief | Comment Count:
1
Video ID: oXGVGXAx9Ng | Title: Odyssey ¿ Smooth Lofi Beats To Relax,
Study, Vibe To (Lofi Mix) | Comment Count: 1
Video ID: LIdwACwZ7E4 | Title: Can You spot the Fake Rubik's Cube? ¿ |
Comment Count: 1
Video ID: be1frkduYbY | Title: Was bringt das Apple-September-2022-Event
- 4K | Apfeltalk | Comment Count: 1
Video ID: rlV10AFUES0 | Title: How to Create Weekly Planner For Business
- Canva For Beginners | Comment Count: 1
Video ID: ETsM6CGZ0_Y | Title: Canadian Ultimate Championships Juniors:
Open Final | Ultimate Frisbee | CBC Sports | Comment Count: 1
Video ID: 6lbJ55D7rKg | Title: Sony PlayStation is PAYING for Games NOT
to go on Xbox Game Pass?! | Comment Count: 1
Video ID: 9QW7CORIoFw | Title: SAY HALO To Animals!!!|Doodle with
me (Part 4) | Comment Count: 1
Video ID: E3R0mnnMP0k | Title: Rep. Boebert skips crypto reporting cutoff,
and cutting emissions with blockchain: CNBC Crypto World | Comment
Count: 1
```

Question #4

Can you list the top 3 videos with the most views, and then list the lowest 3 videos with the least?

List the top 3 videos with the most views, then list the lowest 3 videos with the least amount of views.

Code:

```
DECLARE
  CURSOR top3_desc_cursor IS
    SELECT videoid, title, NVL(viewcount, 0) AS viewcount
    FROM videoinfo
    ORDER BY viewcount DESC
    FETCH FIRST 3 ROWS ONLY;
    CURSOR top3_asc_cursor IS
        SELECT videoid, title, NVL(viewcount, 0) AS viewcount
        FROM videoinfo
        ORDER BY viewcount ASC
        FETCH FIRST 3 ROWS ONLY;
    video_id VideoInfo.VIDEOID%TYPE;
    title VideoInfo.TITLE%TYPE;
    view_count NUMBER;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Top 3 Videos (Descending Order):');
    FOR desc_rec IN top3_desc_cursor LOOP
        video_id := desc_rec.videoid;
        title := desc_rec.title;
        view_count := desc_rec.viewcount;
        DBMS_OUTPUT.PUT_LINE('Video ID: ' || video_id || ' | Title: ' || title || ' | View Count: ' || view_count);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Top 3 Videos (Ascending Order):');
    FOR asc_rec IN top3_asc_cursor LOOP
        video_id := asc_rec.videoid;
        title := asc_rec.title;
        view_count := asc_rec.viewcount;
        DBMS_OUTPUT.PUT_LINE('Video ID: ' || video_id || ' | Title: ' || title || ' | View Count: ' || view_count);
    END LOOP;
END;
/
```

Results:

```
Top 3 Videos (Descending Order):
Video ID: FzG4uDgje3M | Title: El Chombo - Dame Tu Cosita feat. Cutty Ranks (Official Video) [Ultra Music] | View Count: 4034122271
Video ID: gCYcHz2k5x0 | Title: Martin Garrix - Animals (Official Video) | View Count: 1582262997
Video ID: XXYlFuWEuKI | Title: The Weeknd - Save Your Tears (Official Music Video) | View Count: 915457091
Top 3 Videos (Ascending Order):
Video ID: V_vguZj_7FE | Title: Mathematics and Chemistry : MathChemistry.com : Masters Degree in Math | View Count: 25
Video ID: TlEpQgYBS0E | Title: Mathematics and Chemistry : MathChemistry.com : Masters Degree in Math | View Count: 63
Video ID: =-Mv5KqtuVrc | Title: How to Assess ANY Chess Position: The Ultimate Guide | View Count: 601
```

Question #5

Using a loop, can you find the top 3 videos with the most comments?

Select the top 3 videos with the most comments. Use a LOOP.

Code:

```
DECLARE
   CURSOR top3_comment_count_cursor IS
      SELECT c.videoID, v.title, COUNT(c.CommentID) AS commentCount
      FROM CommentInfo c
      JOIN VideoInfo v ON c.videoID = v.videoID
      GROUP BY c.videoID, v.title
      ORDER BY commentCount DESC
      FETCH FIRST 3 ROWS ONLY;
   video_id CommentInfo.VIDEOID%TYPE;
   title VideoInfo.TITLE%TYPE;
   comment_count NUMBER;
BEGIN
   DBMS_OUTPUT.PUT_LINE('Top 3 Videos with Most Comments:');
   FOR comment_count_rec IN top3_comment_count_cursor LOOP
      video_id := comment_count_rec.videoID;
      title := comment_count_rec.title;
      comment_count := comment_count_rec.commentCount;
      DBMS_OUTPUT.PUT_LINE('Video ID: ' || video_id || ' | Title: ' || title || ' | Comment Count: ' || comment_count);
   END LOOP;
END;
/
```

Results:

```
Top 3 Videos with Most Comments:
Video ID: LeC5yJq4tsI | Title: 20 Minecraft Block Facts You Maybe Didn&#39;t Know | Comment Count: 20
Video ID: ZgeorpjGJC0 | Title: Lofi For Reading ¿ Lofi Hip Hop | Study Music ¿ Study Beats | Lofi Study Music | Comment Count: 20
Video ID: mqc6QqoGNWI | Title: ASMR Gaming ¿ Fortnite 1 Kill = 1 Trigger Relaxing Mouth Sounds ¿¿ Controller Sounds + Whispering ¿ | Comment Count: 18
```

Question #6

Can you find the correlation between video popularity and the upload date?

Find the correlation between video popularity and upload date.
Find the date uploaded of all videos, and calculate the average viewcount of all videos
within that day.

Code:

```
DECLARE
   CURSOR upload_date_avg_views_cursor IS
      SELECT date_uploaded, average_views
      FROM (
         SELECT TO_CHAR(uploaddate, 'DD-MON-YYYY') AS
date_uploaded, AVG(viewcount) AS average_views
         FROM VideoInfo
         GROUP BY TO_CHAR(uploaddate, 'DD-MON-YYYY')
      ) ORDER BY date_uploaded;
   date_uploaded VARCHAR2(20);
   average_views NUMBER;
BEGIN
   DBMS_OUTPUT.PUT_LINE('Average Views Grouped by Upload Date
(Ascending Order):');
   FOR upload_date_avg_views_rec IN upload_date_avg_views_cursor
LOOP
      date_uploaded := upload_date_avg_views_rec.date_uploaded;
      average_views := upload_date_avg_views_rec.average_views;
      DBMS_OUTPUT.PUT_LINE('Date Uploaded: ' || date_uploaded || ' |
Average Views: ' || average_views);
   END LOOP;
END;
/
```

Results:

```
Average Views Grouped by Upload Date (Ascending Order):
Date Uploaded: 01-APR-2019 | Average Views: 143605
Date Uploaded: 01-APR-2020 | Average Views: 524709805
Date Uploaded: 01-APR-2022 | Average Views: 383734
Date Uploaded: 01-AUG-2022 | Average Views: 1706457
Date Uploaded: 01-DEC-2019 | Average Views: 689166
Date Uploaded: 01-DEC-2020 | Average Views: 1670627
Date Uploaded: 01-FEB-2011 | Average Views: 14800155
Date Uploaded: 01-FEB-2017 | Average Views: 10682194.5
Date Uploaded: 01-FEB-2018 | Average Views: 793228
Date Uploaded: 01-FEB-2022 | Average Views: 595260.5
Date Uploaded: 01-JAN-2022 | Average Views: 855620
Date Uploaded: 01-JUL-2018 | Average Views: 800364
Date Uploaded: 01-JUN-2018 | Average Views: 7906241
Date Uploaded: 01-JUN-2019 | Average Views: 37780348
Date Uploaded: 01-JUN-2020 | Average Views: 215489
Date Uploaded: 01-JUN-2022 | Average Views:
713127.3333333333333333333333333333333333
Date Uploaded: 01-MAR-2017 | Average Views: 1750656
Date Uploaded: 01-MAR-2022 | Average Views: 123228
Date Uploaded: 01-MAY-2021 | Average Views: 2354668
Date Uploaded: 01-MAY-2022 | Average Views: 4606124
Date Uploaded: 01-NOV-2017 | Average Views: 872008
Date Uploaded: 01-OCT-2020 | Average Views: 367200
Date Uploaded: 01-OCT-2021 | Average Views: 589907
Date Uploaded: 02-APR-2012 | Average Views: 8639567
Date Uploaded: 02-APR-2022 | Average Views: 4099070.2
Date Uploaded: 02-AUG-2017 | Average Views: 196484
Date Uploaded: 02-AUG-2022 | Average Views: 1786905
Date Uploaded: 02-DEC-2020 | Average Views: 2615459
Date Uploaded: 02-DEC-2021 | Average Views: 12703147
Date Uploaded: 02-JAN-2017 | Average Views: 844015
Date Uploaded: 02-JAN-2022 | Average Views: 305650
```

Question #7

How would you find the most popular upload date?

Find the most popular upload date and count the number of videos uploaded per date.
Use FETCH(First 10 Rows) and order by descending order of Upload_Count.

Code:

```
DECLARE
   CURSOR upload_count_cursor IS
      SELECT date_uploaded, upload_count
      FROM (
         SELECT TO_CHAR(uploaddate, 'DD-MON-YYYY') AS
date_uploaded,
                COUNT(*) AS upload_count,
                ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS
rn
         FROM VideoInfo
         GROUP BY TO_CHAR(uploaddate, 'DD-MON-YYYY')
      ) WHERE rn <= 10;
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('Top 10 Upload Dates with Upload Count (Descending Order):');
    FOR upload_count_rec IN upload_count_cursor LOOP
        DBMS_OUTPUT.PUT_LINE('Date Uploaded: ' || upload_count_rec.date_uploaded || ' | Upload Count: ' || upload_count_rec.upload_count);
    END LOOP;
END;
/
```

Results:

```
Date Uploaded: 24-AUG-2022 | Upload Count: 282
Date Uploaded: 23-AUG-2022 | Upload Count: 182
Date Uploaded: 22-AUG-2022 | Upload Count: 39
Date Uploaded: 20-AUG-2022 | Upload Count: 38
Date Uploaded: 21-AUG-2022 | Upload Count: 33
Date Uploaded: 17-AUG-2022 | Upload Count: 23
Date Uploaded: 18-AUG-2022 | Upload Count: 21
Date Uploaded: 19-AUG-2022 | Upload Count: 17
Date Uploaded: 11-AUG-2022 | Upload Count: 16
Date Uploaded: 24-JUL-2022 | Upload Count: 16
```

Question #8

Can you list out the videos with the most comments and likes?

Find the video with the most engagement (comments + likes).

Code:

```
DECLARE
    CURSOR engagement_cursor IS
        SELECT e.videoID, v.title, (e.likes + e.comments) AS totalEngagement
        FROM EngagementMetrics e
        JOIN VideoInfo v ON e.videoID = v.videoID
        ORDER BY totalEngagement DESC
        FETCH FIRST 1 ROW ONLY;
    video_id EngagementMetrics.VIDEOID%TYPE;
    title VideoInfo.TITLE%TYPE;
    total_engagement NUMBER;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Video with Highest Total Engagement:');
    FOR engagement_rec IN engagement_cursor LOOP
        video_id := engagement_rec.videoID;
        title := engagement_rec.title;
        total_engagement := engagement_rec.totalEngagement;
        DBMS_OUTPUT.PUT_LINE('Video ID: ' || video_id || ' | Title: ' || title || '
| Total Engagement: ' || total_engagement);
    END LOOP;
END;
/
```

Results:

```
Video with Highest Total Engagement:
Video ID: nmY2kgWYwyQ | Title: I bought the THINNEST Tech in the
world. | Total Engagement: 377380
```

Question #9

Can you make a function that calculates the ratio between likes and comments?

Create a FUNCTION:
Calculate the LIKES-COMMENTS ratio of a specified video:
-Accepts videoID input within the procedures parameters
-Must output in the format ('Like-Comment Ratio: ')
-Must handle user input errors
-Must handle OTHER errors
-Include the section EXECUTE statement for testing and running the code.

Code:

```
CREATE OR REPLACE FUNCTION
CalculateLikesCommentsRatio(video_id IN VARCHAR2)
RETURN VARCHAR2
IS
    likes          NUMBER;
    comments       NUMBER;
    ratio          NUMBER;
    video_title    VARCHAR2(255);
    error_message  VARCHAR2(255);
BEGIN
    IF video_id IS NULL OR NOT REGEXP_LIKE(video_id,
'^[a-zA-Z0-9_]+$') THEN
        error_message := 'Invalid video ID';
        RETURN error_message;
    END IF;
    SELECT NVL(SUM(LIKES), 0)
    INTO likes
```

```
   FROM EngagementMetrics
   WHERE VIDEOID = video_id;
   SELECT NVL(COUNT(COMMENTID), 0)
   INTO comments
   FROM CommentInfo
   WHERE VIDEOID = video_id;
   SELECT TITLE
   INTO video_title
   FROM VideoInfo
   WHERE VIDEOID = video_id;
   DBMS_OUTPUT.PUT_LINE('The Name of Video Chosen is: ' || video_title);
   IF comments = 0 THEN
      ratio := -1;
   ELSE
      ratio := likes / comments;
   END IF;
   IF likes > comments THEN
      RETURN 'There are more likes than comments. Like-Comment Ratio: ' || TO_CHAR(ratio);
   ELSIF likes < comments THEN
      RETURN 'There are more comments than likes. Like-Comment Ratio: ' || TO_CHAR(ratio);
   ELSE
      RETURN 'Likes and comments are equal. Like-Comment Ratio: ' || TO_CHAR(ratio);
   END IF;

EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RETURN 'Video not found';
   WHEN OTHERS THEN
      error_message := SQLERRM;
      RETURN 'Error: ' || error_message;

END;
/

--Execution Block
DECLARE
   result_message VARCHAR2(255);
   video_id       VARCHAR2(50) := 'b3x28s61q3c';
BEGIN
   result_message := CalculateLikesCommentsRatio(video_id);
   DBMS_OUTPUT.PUT_LINE(result_message);
END;
```

```
/
```

Results:

The Name of Video Chosen is: The most EXPENSIVE thing I own.
There are more likes than comments. Like-Comment Ratio: 7677.9

Question #10

Can you write a query that outputs all of the significant data for a specified video? VIEWCOUNT, #OFCOMMENTS, #OFLIKES.

Query that outputs the total viewcount, number of comments, number of likes, of a specified video.
It outputs the title of the video, and the statistics based on the parameters of the videoID.

Code:

```
SET SERVEROUTPUT ON;

DECLARE
   video_id      VARCHAR2(50) := 'wAZZ-UWGVHI';
   video_name    VARCHAR2(255);
   total_views   NUMBER;
   comment_count NUMBER;
   likes         NUMBER;
BEGIN
   SELECT TITLE
   INTO video_name
   FROM VideoInfo
   WHERE VIDEOID = video_id;
   SELECT NVL(SUM(VIEWCOUNT), 0)
   INTO total_views
   FROM VideoInfo
   WHERE VIDEOID = video_id;
   SELECT NVL(COUNT(COMMENTID), 0)
   INTO comment_count
   FROM CommentInfo
   WHERE VIDEOID = video_id;
   SELECT NVL(SUM(LIKES), 0)
   INTO likes
```

```
    FROM EngagementMetrics
    WHERE VIDEOID = video_id;
    DBMS_OUTPUT.PUT_LINE('Video Name: ' || video_name);
    DBMS_OUTPUT.PUT_LINE('Total View Count: ' || total_views);
    DBMS_OUTPUT.PUT_LINE('Total Comment Count: ' || comment_count);
    DBMS_OUTPUT.PUT_LINE('Total Likes: ' || likes);
END;
/
```

Results:

```
Video Name: Apple Pay Is Killing the Physical Wallet After Only Eight Years
| Tech News Briefing Podcast | WSJ
Total View Count: 135612
Total Comment Count: 10
Total Likes: 3407
```

Team Contributions:

Name:	Contributions:
Yooto Joo	Created the document, re-coded Q1~Q10 SQL queries to be in the form of PL/SQL, edited all the queries to provide additional insights, checked for bugs, made the database, filtered and cleaned the data, granted access.
Adam Hafez	Re-coded blocks to PL/SQL, edited queries, debugging, document review
Stewart Fetzko	Coded original SQL queries, collected output, re-coded SQL to PL/SQL, debugging
Minseong Jeong	Debugging, Coded original SQL queries, made presentation