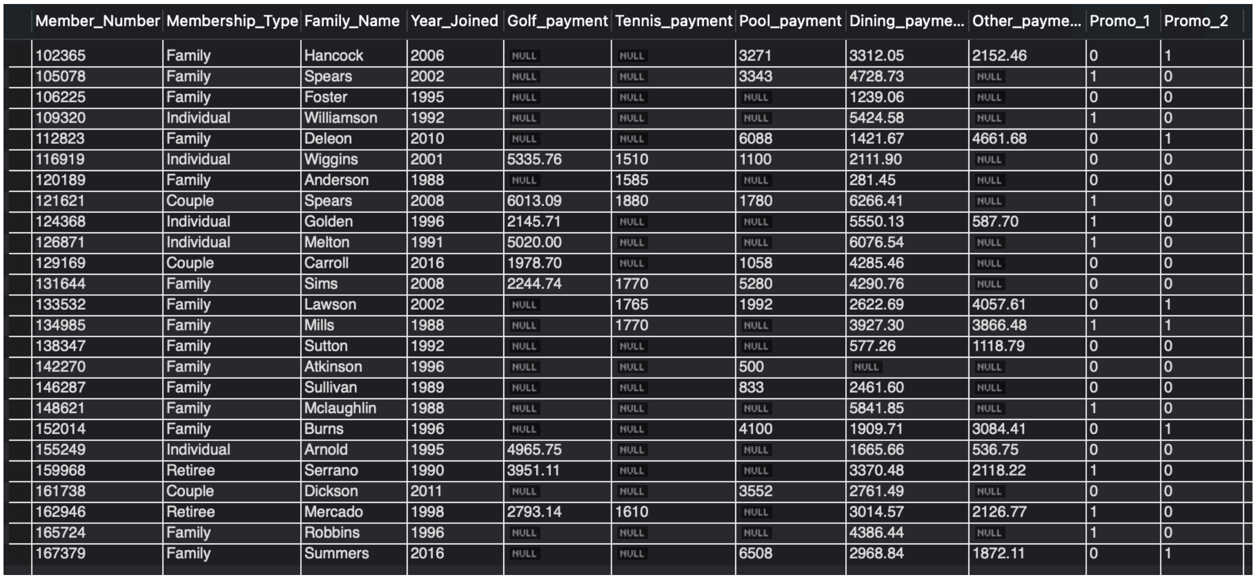
HW 5&6
Sirui Tang

1. Design and create a data warehouse for the country club database. The decisions about which fields to include and how to aggregate the data are left to you. You do not need to include every single data point from the 10 tables given. Do your best and use your judgement as to what is interesting/useful.

   DW revenue:
   I have developed a data warehouse focusing on revenue that the club can receive from providing golf courses, tennis courts, dining, pool, and miscellaneous activities. I aggregated payments in areas such as golf, tennis, pool, other, and dining that have been recorded in the dataset. Additionally, I've documented dates for members with promo one and identified those with promo two. Its primary purpose is to facilitate analysis and reporting of financial transactions across various dimensions, like membership_type, activities, and year of joined.



Member_Number	Membership_Type	Family_Name	Year_Joined	Golf_payment	Tennis_payment	Pool_payment	Dining_payme...	Other_payme...	Promo_1	Promo_2
102365	Family	Hancock	2006	NULL	NULL	3271	3312.05	2152.46	0	1
105078	Family	Spears	2002	NULL	NULL	3343	4728.73	NULL	1	0
106225	Family	Foster	1995	NULL	NULL	NULL	1239.06	NULL	0	0
109320	Individual	Williamson	1992	NULL	NULL	NULL	5424.58	NULL	1	0
112823	Family	Deleon	2010	NULL	NULL	6088	1421.67	4661.68	0	1
116919	Individual	Wiggins	2001	5335.76	1510	1100	2111.90	NULL	0	0
120189	Family	Anderson	1988	NULL	1585	NULL	281.45	NULL	0	0
121621	Couple	Spears	2008	6013.09	1880	1780	6266.41	NULL	1	0
124368	Individual	Golden	1996	2145.71	NULL	NULL	5550.13	587.70	1	0
126871	Individual	Melton	1991	5020.00	NULL	NULL	6076.54	NULL	1	0
129169	Couple	Carroll	2016	1978.70	NULL	1058	4285.46	NULL	0	0
131644	Family	Sims	2008	2244.74	1770	5280	4290.76	NULL	0	0
133532	Family	Lawson	2002	NULL	1765	1992	2622.69	4057.61	0	1
134985	Family	Mills	1988	NULL	1770	NULL	3927.30	3866.48	1	1
138347	Family	Sutton	1992	NULL	NULL	NULL	577.26	1118.79	0	0
142270	Family	Atkinson	1996	NULL	NULL	500	NULL	NULL	0	0
146287	Family	Sullivan	1989	NULL	NULL	833	2461.60	NULL	0	0
148621	Family	Mclaughlin	1988	NULL	NULL	NULL	5841.85	NULL	1	0
152014	Family	Burns	1996	NULL	NULL	4100	1909.71	3084.41	0	1
155249	Individual	Arnold	1995	4965.75	NULL	NULL	1665.66	536.75	0	0
159968	Retiree	Serrano	1990	3951.11	NULL	NULL	3370.48	2118.22	1	0
161738	Couple	Dickson	2011	NULL	NULL	3552	2761.49	NULL	0	0
162946	Retiree	Mercado	1998	2793.14	1610	NULL	3014.57	2126.77	1	0
165724	Family	Robbins	1996	NULL	NULL	NULL	4386.44	NULL	1	0
167379	Family	Summers	2016	NULL	NULL	6508	2968.84	1872.11	0	1

SQL CODE:

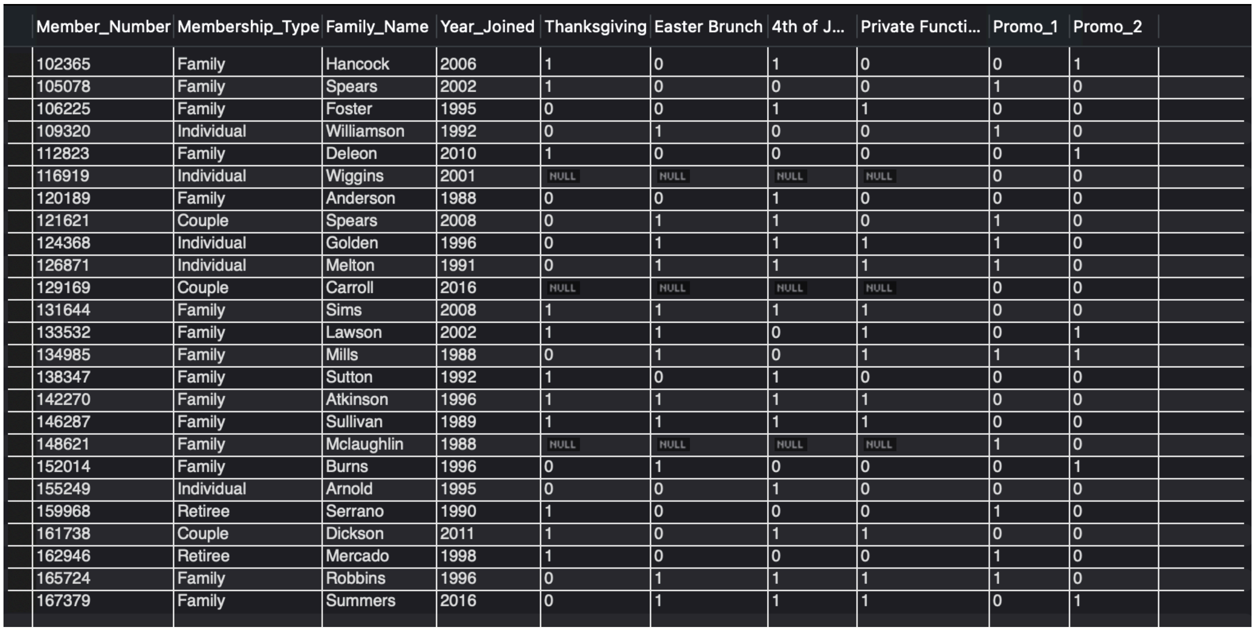
```
CREATE OR REPLACE VIEW revenue AS
SELECT memberships.Member_Number, memberships.Membership_Type, memberships.Family_Name, memberships.Year_Joined,
                  Golf_payment, Tennis_payment, Pool_payment, Dining_payment, Other_payment,
                  CASE WHEN Promo_1_date IS NOT NULL THEN 1 ELSE 0 END AS Promo_1,
                  CASE WHEN promotwo.Member_Number IS NOT NULL THEN 1 ELSE 0 END AS Promo_2
FROM memberships
         LEFT JOIN ( SELECT golf.Member_Number, SUM(Amount) AS Golf_payment
                                    FROM golf
        #WHERE Description = "Golf_Lessons"
                                    GROUP BY golf.Member_Number
                                    ) AS golf_lesson
                  ON golf_lesson.Member_Number = memberships.Member_Number
         LEFT JOIN ( SELECT tennis.Member_Number, SUM(Amount) AS Tennis_payment
                                    FROM tennis
                                    GROUP BY tennis.Member_Number
                                    ) AS tennis_play
                  ON tennis_play.Member_Number = memberships.Member_Number
```

```
    LEFT JOIN ( SELECT pool.Member_Number, SUM(Amount) AS Pool_payment
                FROM pool
                GROUP BY pool.Member_Number
                ) AS pool
        ON pool.Member_Number = memberships.Member_Number
    LEFT JOIN ( SELECT dining.Member_Number, SUM(Total) AS Dining_payment
                FROM dining
                GROUP BY dining.Member_Number
                ) AS dining
        ON dining.Member_Number = memberships.Member_Number
    LEFT JOIN ( SELECT other.Member_Number, SUM(Amount) AS Other_payment
                FROM other
                GROUP BY other.Member_Number
                ) AS other
        ON other.Member_Number = memberships.Member_Number
    LEFT JOIN (SELECT promoone.Member_Number, Date AS Promo_1_date
            FROM promoone) AS pone ON pone.Member_Number = memberships.Member_Number
    LEFT JOIN promotwo ON promotwo.Member_Number = memberships.Member_Number
ORDER BY memberships.Member_Number;

SELECT *
FROM revenue
LIMIT 25;
```

DW members_activities:

I've established distinct data warehouses for members engaged in different special functions in 2018. Additionally, I've documented dates for members with promo one and identified those with promo two. The diverse interests of the members are evident. We can use the information from this data warehouse to send suitable promote of special events or offers to members. Its purpose is to centralize attendance and participation metrics for special occasions, making it easy to analyze trends, member engagement, and event popularity



Member_Number	Membership_Type	Family_Name	Year_Joined	Thanksgiving	Easter Brunch	4th of J...	Private Functi...	Promo_1	Promo_2
102365	Family	Hancock	2006	1	0	1	0	0	1
105078	Family	Spears	2002	1	0	0	0	1	0
106225	Family	Foster	1995	0	0	1	1	0	0
109320	Individual	Williamson	1992	0	1	0	0	1	0
112823	Family	Deleon	2010	1	0	0	0	0	1
116919	Individual	Wiggins	2001	NULL	NULL	NULL	NULL	0	0
120189	Family	Anderson	1988	0	0	1	0	0	0
121621	Couple	Spears	2008	0	1	1	0	1	0
124368	Individual	Golden	1996	0	1	1	1	1	0
126871	Individual	Melton	1991	0	1	1	1	1	0
129169	Couple	Carroll	2016	NULL	NULL	NULL	NULL	0	0
131644	Family	Sims	2008	1	1	1	1	0	0
133532	Family	Lawson	2002	1	1	0	1	0	1
134985	Family	Mills	1988	0	1	0	1	1	1
138347	Family	Sutton	1992	1	0	1	0	0	0
142270	Family	Atkinson	1996	1	1	1	1	0	0
146287	Family	Sullivan	1989	1	1	1	1	0	0
148621	Family	Mclaughlin	1988	NULL	NULL	NULL	NULL	1	0
152014	Family	Burns	1996	0	1	0	0	0	1
155249	Individual	Arnold	1995	0	0	1	0	0	0
159968	Retiree	Serrano	1990	1	0	0	0	1	0
161738	Couple	Dickson	2011	1	0	1	1	0	0
162946	Retiree	Mercado	1998	1	0	0	0	1	0
165724	Family	Robbins	1996	0	1	1	1	1	0
167379	Family	Summers	2016	0	1	1	1	0	1

SQL CODE:

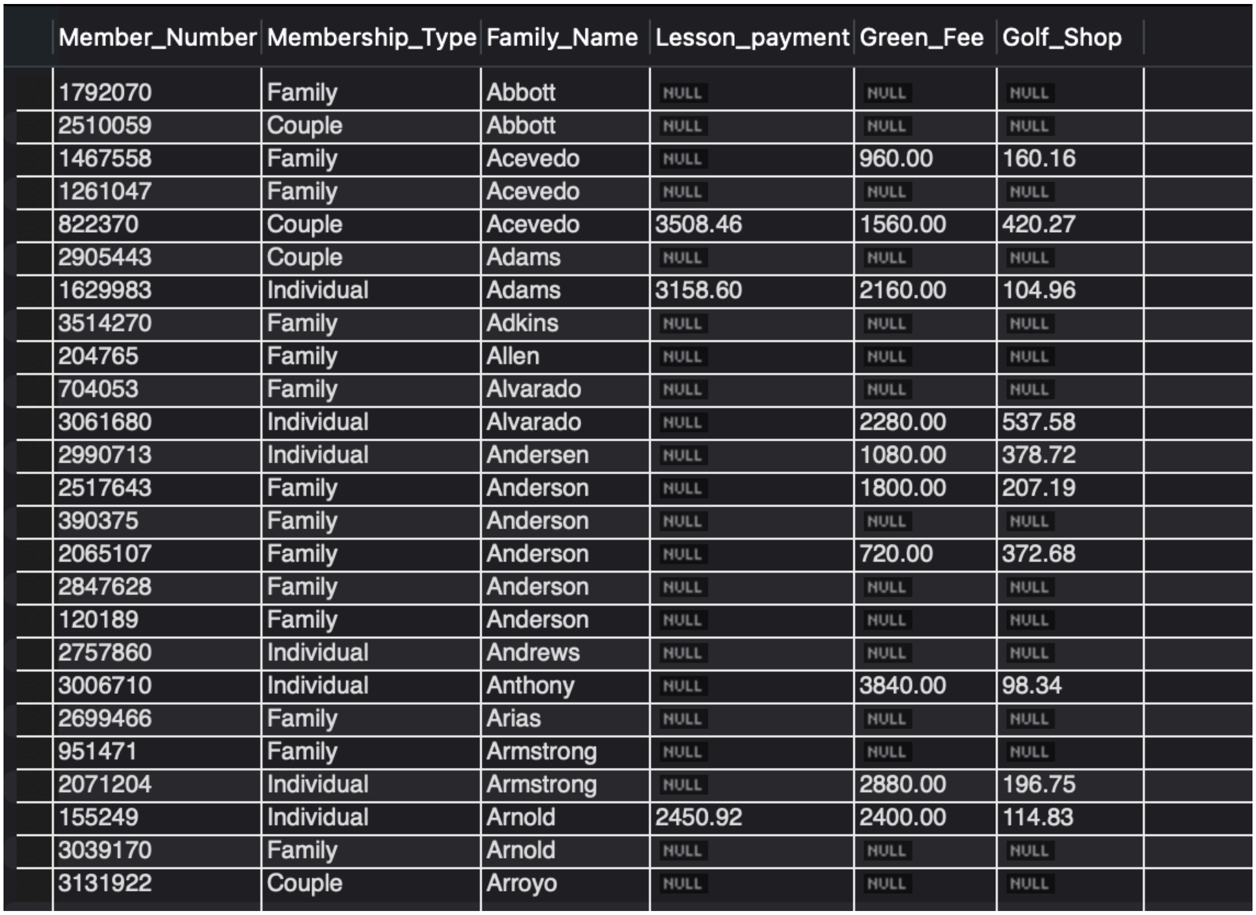
```
CREATE OR REPLACE VIEW members_activities AS
SELECT memberships.Member_Number, memberships.Membership_Type, memberships.Family_Name, memberships.Year_Joined,
                    Thanksgiving, `Easter Brunch`, `4th of July`, `Private Function`,
    CASE WHEN Promo_1_date IS NOT NULL THEN 1 ELSE 0 END AS Promo_1,
    CASE WHEN promotwo.Member_Number IS NOT NULL THEN 1 ELSE 0 END AS Promo_2
FROM memberships
LEFT JOIN ( SELECT special.Member_Number, special.Thanksgiving
                                    FROM special
                                    GROUP BY special.Member_Number
                                    ) AS Thanksgiving
                    ON Thanksgiving.Member_Number = memberships.Member_Number
LEFT JOIN ( SELECT special.Member_Number, special.`Easter Brunch`
                                    FROM special
                                    GROUP BY special.Member_Number
                                    ) AS EB
                    ON EB.Member_Number = memberships.Member_Number
LEFT JOIN ( SELECT special.Member_Number, special.`4th of July`
                                    FROM special
                                    GROUP BY special.Member_Number
                                    ) AS 4th_of_July
                    ON 4th_of_July.Member_Number = memberships.Member_Number
LEFT JOIN ( SELECT special.Member_Number, special.`Private Function`
                                    FROM special
                                    GROUP BY special.Member_Number
                                    ) AS PF
                    ON PF.Member_Number = memberships.Member_Number
```

```
LEFT JOIN (SELECT promoone.Member_Number, Date AS Promo_1_date
   FROM promoone) AS pone ON pone.Member_Number = memberships.Member_Number
              LEFT JOIN promotwo ON promotwo.Member_Number = memberships.Member_Number
ORDER BY memberships.Member_Number;

SELECT *
FROM members_activities
LIMIT 25;
```

DW golf_payment:

I have created dedicated data warehouses for members involved in golf-related activities, covering areas such as golf lessons and purchases from the golf shop. These warehouses provide a detailed breakdown of payments for each activity.



Member_Number	Membership_Type	Family_Name	Lesson_payment	Green_Fee	Golf_Shop
1792070	Family	Abbott	NULL	NULL	NULL
2510059	Couple	Abbott	NULL	NULL	NULL
1467558	Family	Acevedo	NULL	960.00	160.16
1261047	Family	Acevedo	NULL	NULL	NULL
822370	Couple	Acevedo	3508.46	1560.00	420.27
2905443	Couple	Adams	NULL	NULL	NULL
1629983	Individual	Adams	3158.60	2160.00	104.96
3514270	Family	Adkins	NULL	NULL	NULL
204765	Family	Allen	NULL	NULL	NULL
704053	Family	Alvarado	NULL	NULL	NULL
3061680	Individual	Alvarado	NULL	2280.00	537.58
2990713	Individual	Andersen	NULL	1080.00	378.72
2517643	Family	Anderson	NULL	1800.00	207.19
390375	Family	Anderson	NULL	NULL	NULL
2065107	Family	Anderson	NULL	720.00	372.68
2847628	Family	Anderson	NULL	NULL	NULL
120189	Family	Anderson	NULL	NULL	NULL
2757860	Individual	Andrews	NULL	NULL	NULL
3006710	Individual	Anthony	NULL	3840.00	98.34
2699466	Family	Arias	NULL	NULL	NULL
951471	Family	Armstrong	NULL	NULL	NULL
2071204	Individual	Armstrong	NULL	2880.00	196.75
155249	Individual	Arnold	2450.92	2400.00	114.83
3039170	Family	Arnold	NULL	NULL	NULL
3131922	Couple	Arroyo	NULL	NULL	NULL

SQL CODE:

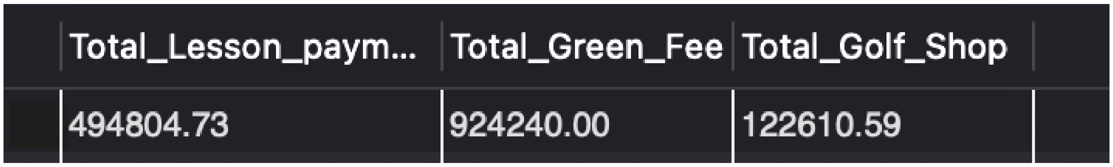
```
CREATE OR REPLACE VIEW golf_payment AS
SELECT memberships.Member_Number, Membership_Type, Family_Name, Lesson_payment, Green_Fee, Golf_Shop
FROM memberships
LEFT JOIN ( SELECT golf.Member_Number, SUM(Amount) AS Lesson_payment
                                        FROM golf
          WHERE Description = "Golf_Lessons"
                                        GROUP BY golf.Member_Number
                                        ) AS golf_lesson
                    ON golf_lesson.Member_Number = memberships.Member_Number
LEFT JOIN ( SELECT golf.Member_Number, SUM(Amount) AS Green_Fee
                                        FROM golf
          WHERE Description = "Green_Fee"
                                        GROUP BY golf.Member_Number
                                        ) AS golf_green
                    ON golf_green.Member_Number = memberships.Member_Number
LEFT JOIN ( SELECT golf.Member_Number, SUM(Amount) AS Golf_Shop
                                        FROM golf
          WHERE Description = "Golf_Shop"
                                        GROUP BY golf.Member_Number
                                        ) AS golf_shop
                    ON golf_shop.Member_Number = memberships.Member_Number
```

```
ORDER BY memberships.Member_Number;

SELECT *
FROM golf_payment
LIMIT 25;
```

2. Create two SQL queries on your data warehouse.

   a. To identify the category where golf players spend the most, I analyzed the club's revenue from golf lessons, green fees, and the golf shop. The results clearly indicate that revenue from golf lessons exceeds that of both the golf shop and green fees. Thus, for members who engage in golf, golf lessons emerge as the club's primary source of income.

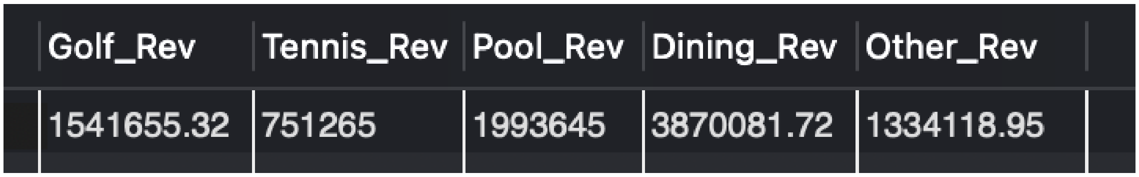


Total_Lesson_paym...	Total_Green_Fee	Total_Golf_Shop
494804.73	924240.00	122610.59

SQL CODE:

```
SELECT SUM(Lesson_payment) AS Total_Lesson_payment,
SUM(Green_Fee) AS Total_Green_Fee, SUM(Golf_Shop) AS Total_Golf_Shop
FROM golf_payment;
```

   b. To determine which activity generates the most revenue, I analyzed the total revenue from golf, tennis, pool, dining, and other activities. The analysis revealed that dining contributes the highest revenue to the club, while tennis generates the least. This suggests that the club could further explore tennis-related data to identify opportunities for increasing revenue from tennis activities.

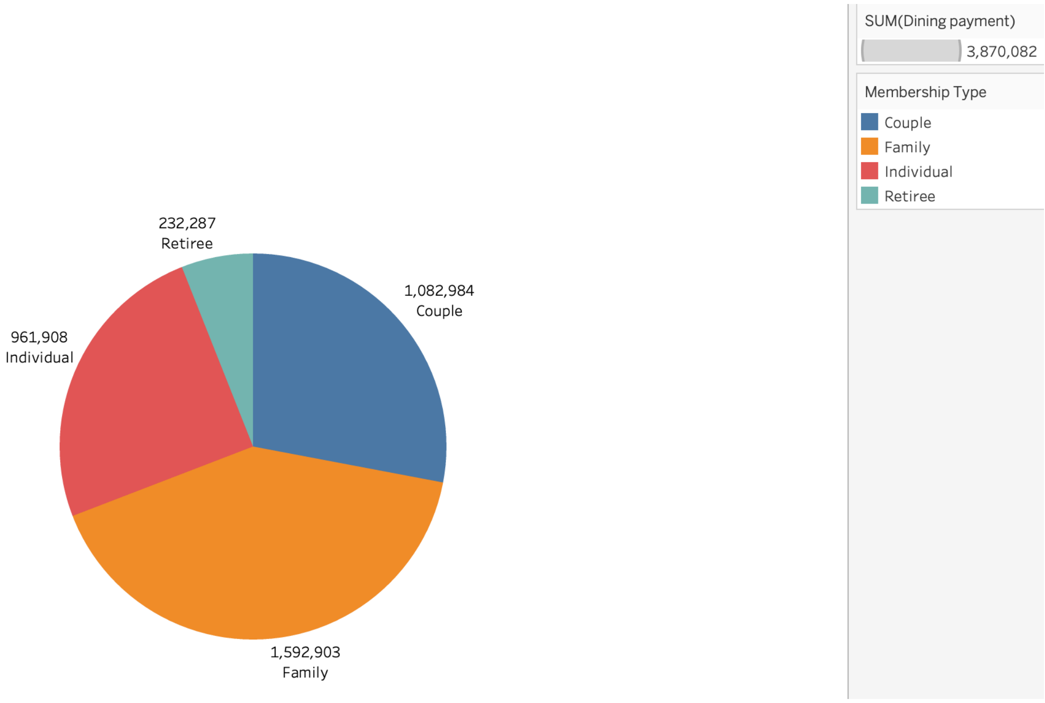


Golf_Rev	Tennis_Rev	Pool_Rev	Dining_Rev	Other_Rev
1541655.32	751265	1993645	3870081.72	1334118.95

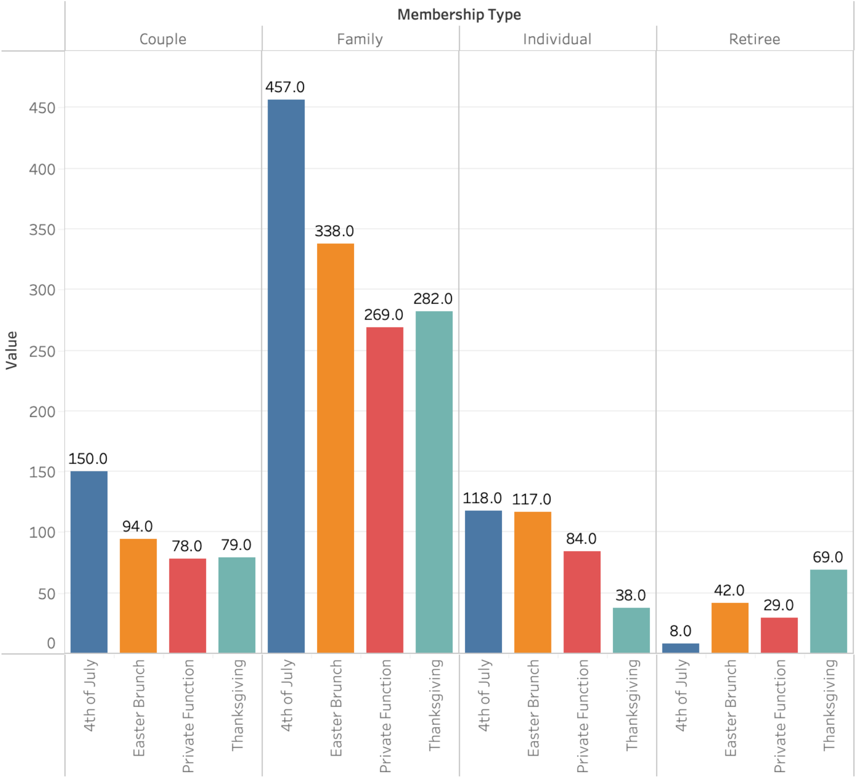
SQL CODE:

```
SELECT SUM(Golf_payment) AS Golf_Rev,
            SUM(Tennis_payment) AS Tennis_Rev,
  SUM(Pool_payment) AS Pool_Rev,
  SUM(Dining_payment) AS Dining_Rev,
  SUM(Other_payment) AS Other_Rev
FROM revenue;
```

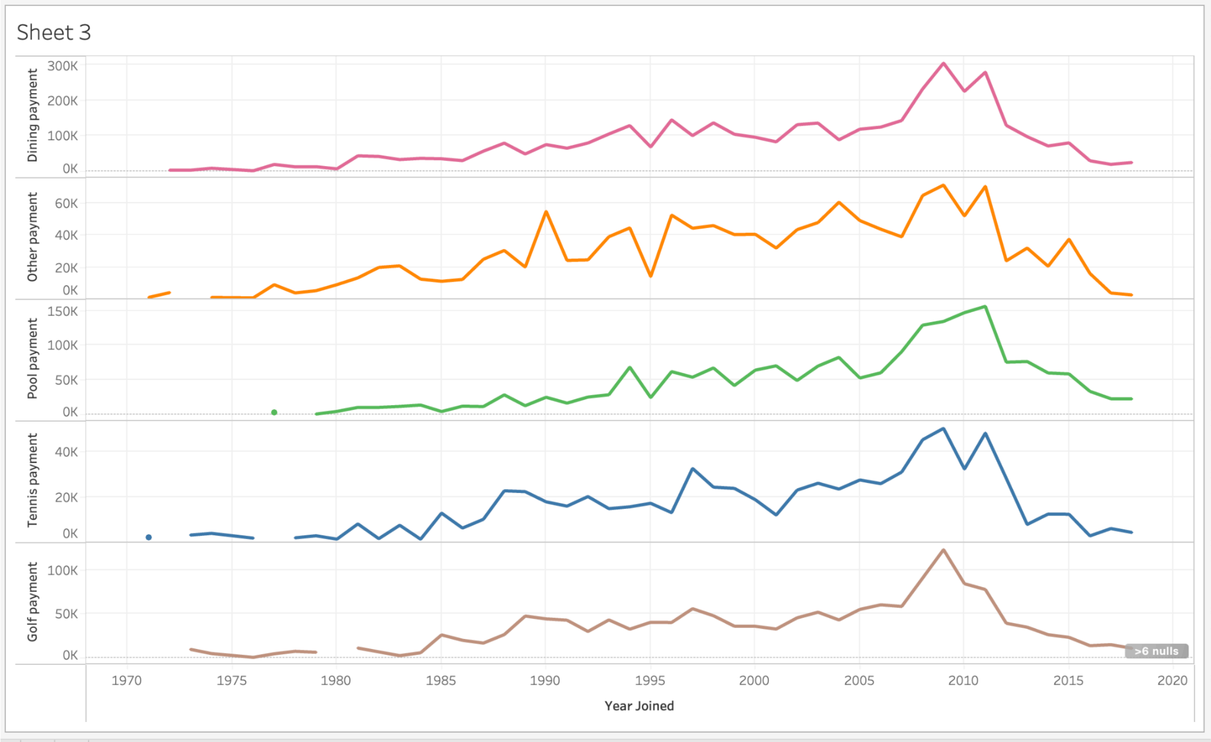
3. Create three visualizations in Tableau using your data warehouse.
    a. I am using **the revenue data warehouse**. This pie chart illustrates the distribution of dining payments across different membership types, revealing that Family memberships contribute the largest share (1,592,903), followed by Couple memberships (1,082,984), Individual memberships (961,908), and Retiree memberships, which contribute the least (232,287). The total dining revenue amounts to 3,870,082. These insights highlight the significant role Family memberships play in driving dining revenue, suggesting an opportunity to maintain or enhance offerings tailored to this group. Conversely, the low contribution from retirees presents a potential area for growth through targeted campaigns or promotions to increase engagement.



    b. I am using **the member_activity data warehouse**. This bar chart visualizes the participation of different membership types (Couple, Family, Individual, and Retiree) across four special events: 4th of July, Easter Brunch, Private Function, and Thanksgiving. It shows that Family memberships have the highest participation across all events, particularly for the 4th of July and Easter Brunch. Couple memberships also exhibit relatively strong engagement on the 4th of July compared with the last two groups. Individual memberships participate moderately, while Retiree memberships show the least participation in most events, with Thanksgiving being their most attended event. This visualization highlights the popularity of events among different membership types, suggesting opportunities to tailor event offerings to less engaged groups, such as retirees and individuals, to increase participation.



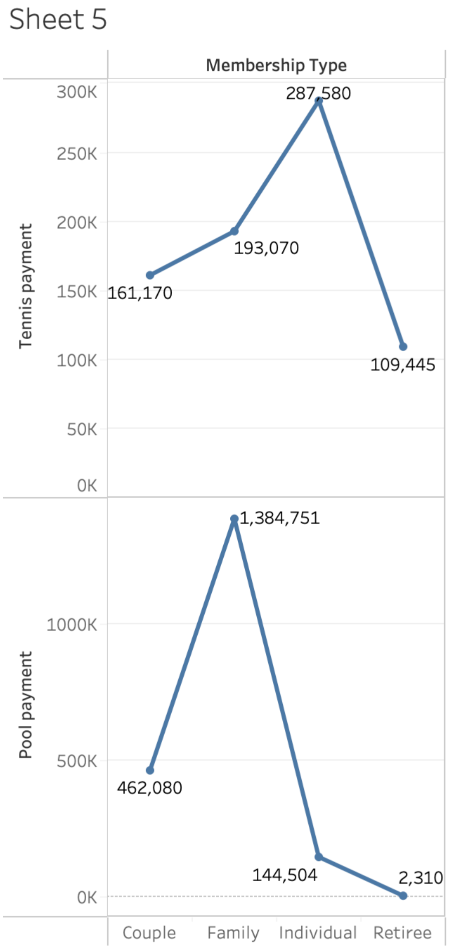
c. I am using **the revenue data warehouse.** This line chart visualizes the trends in revenue from various activities (Dining, Other, Pool, Tennis, and Golf) over time based on the year members joined. It shows that Other payments are more likely to generate the local highest revenue, with a steady upward trend until a peak around 2010, followed by a slight decline. Revenue across all activities fluctuated significantly, peaking between 2005 and 2015. This visualization highlights that the club can track and imitate the promotion or any activity they held between 2005 and 2015 to draw more new members or make more incomes.



Sheet 3

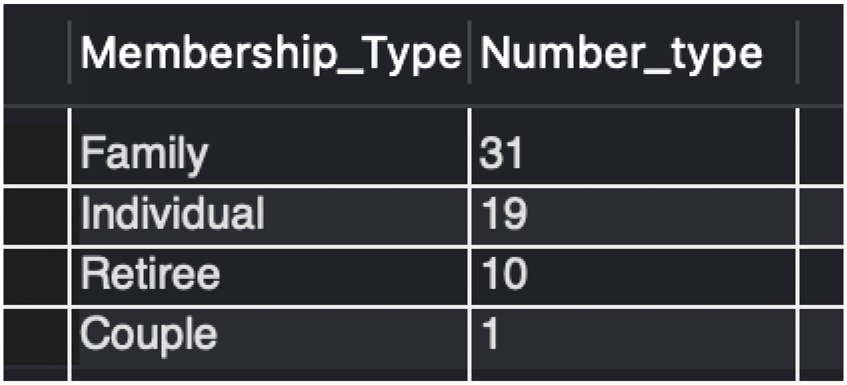
Dining payment
300K
200K
100K
0K
Other payment
60K
40K
20K
0K
Pool payment
150K
100K
50K
0K
Tennis payment
40K
20K
0K
Golf payment
100K
50K
0K
>6 nulls
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
Year Joined

4. In particular, Blue Hill CC would like to use the data warehouse to investigate cross-selling and other marketing opportunities. Make two recommendations for potential marketing targets. Examples: “members who spend a lot of money on XXX are most likely to also spend money on XXX”, “members with more children are most likely to purchase XXX”.
    a. This visualization highlights the trends in Tennis and Pool payments across different membership types, showing that Family memberships are the highest contributors to pool activities, and Individual memberships are the highest contributors to Tennis activities. Families contribute the most to Pool payments, likely due to greater utilization by children or multiple family members. Pool programs targeting families (e.g., family discounts, bundle offers) could further enhance revenue. Tennis programs targeting individuals could also further enhance revenue. Retirees contribute the least to these activities, suggesting room for improvement. The club could explore tailored offerings, such as lower-cost recreational packages or specialized events, to attract this group.



The trends of sum of Tennis payment and sum of Pool payment for Membership Type.

    b. This query identifies members who spend exclusively on dining and groups them by membership type. The results show that Family memberships dominate this category, with 31 families prioritizing dining over other activities, likely reflecting their preference for family-oriented meals or gatherings. Individuals are the second-largest group (19), followed by Retirees (10), who likely view dining as a key social activity. Couples, with only 1 member in this category, are more likely to diversify their spending across other activities like golf or tennis. These insights highlight that dining is a primary focus for families, individuals, and retirees, while couples tend to engage in a broader mix of activities. The club can consider to provide some discounted family or children's packages to earn more.



Membership_Type	Number_type
Family	31
Individual	19
Retiree	10
Couple	1

SQL CODE:

```
SELECT Membership_Type, COUNT(Membership_Type) AS Number_type
FROM revenue
WHERE Dining_payment IS NOT NULL
            AND Golf_payment IS NULL
            AND Tennis_payment IS NULL
   AND Other_payment IS NULL
   AND Pool_payment IS NULL
GROUP BY Membership_Type;
```

5. What can you learn from the data about the nature of Promo1 and Promo2?
The code and its results provide insights into the spending behavior of members based on their participation in Promo1, Promo2, both promotions, or neither.

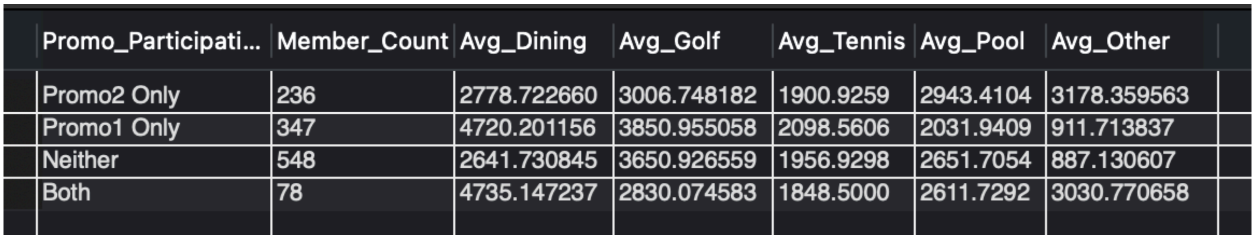
Promo1:
Members participating in Promo1 Only have the highest average spending on Dining (4,720.20) and Golf (3,850.96), indicating that Promo1 targets or attracts members with higher overall spending habits. Promo1 participants also spend significantly on Tennis (2,098.56), which is higher than Promo2 participants.

Promo2:
Members participating in Promo2 Only spend more on Pool (2,943.41) and Other activities (3,178.36). Promo2 participants seem to prioritize pool and miscellaneous club services rather than sports-oriented activities like Golf and Tennis.

Promo1 vs. Promo2:
Promo1 attracts high-spending members with a focus on sports and premium activities, while Promo2 appeals to members with a preference for pool and miscellaneous services.

Both Promo1 and Promo2:
Members participating in Both Promo1 and Promo2 show a relatively high spending pattern in dining and other activities, suggesting that members engaging in both promotions are likely targeted with dining and other activities.

Neither promotion:
Members who participate in neither promotion ("Neither") have the lowest average spending across dining and other activities, highlighting the potential value of engaging non-participants with tailored promotions in dining and other activities to increase their spending.



Promo_Participati...	Member_Count	Avg_Dining	Avg_Golf	Avg_Tennis	Avg_Pool	Avg_Other
Promo2 Only	236	2778.722660	3006.748182	1900.9259	2943.4104	3178.359563
Promo1 Only	347	4720.201156	3850.955058	2098.5606	2031.9409	911.713837
Neither	548	2641.730845	3650.926559	1956.9298	2651.7054	887.130607
Both	78	4735.147237	2830.074583	1848.5000	2611.7292	3030.770658

SQL CODE:

```
SELECT
  CASE
    WHEN Promo_1 = 1 AND Promo_2 = 1 THEN 'Both'
    WHEN Promo_1 = 1 THEN 'Promo1 Only'
    WHEN Promo_2 = 1 THEN 'Promo2 Only'
    ELSE 'Neither'
  END AS Promo_Participation,
  COUNT(*) AS Member_Count,
  AVG(Dining_payment) AS Avg_Dining,
  AVG(Golf_payment) AS Avg_Golf,
    AVG(Tennis_payment) AS Avg_Tennis,
    AVG(Pool_payment) AS Avg_Pool,
    AVG(Other_payment) AS Avg_Other
FROM revenue
GROUP BY Promo_Participation;
```