BI MODULE ASSIGNMENT 1 ANSWERS

Question 1

a. Query 1: Profit Margins by Event Promotion Type and Marketeer.

Figure 1

Query 1

```sql
-- q1a
  SELECT
        CONCAT(UPPER(ed.EventName), '-', ed.EventYear) AS [Event Name],
        pd.PromotionType,
        md.MarketeerName,
        ROUND(SUM(COALESCE(ef.PromotionRevenue, 0) - COALESCE(ef.PromotionCost, 0)), 0) AS [Total Promotion Profit],
--      Ranks by TPP within each grouping set
    RANK() OVER (
        PARTITION BY
            CASE
                    WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NULL AND md.MarketeerName IS NULL THEN 'Event'
                    WHEN pd.PromotionType IS NOT NULL AND ed.EventName IS NULL AND md.MarketeerName IS NULL THEN 'PromotionType'
                    WHEN md.MarketeerName IS NOT NULL AND ed.EventName IS NULL AND pd.PromotionType IS NULL THEN 'Marketeer'
                    WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NOT NULL AND md.MarketeerName IS NULL
                    THEN 'Event + Promotion'
                    WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND md.MarketeerName IS NOT NULL AND pd.PromotionType IS NULL
                    THEN 'Event + Marketeer'
                    WHEN pd.PromotionType IS NOT NULL AND md.MarketeerName IS NOT NULL AND ed.EventName IS NULL THEN 'PromotionType + Marketeer'
                    WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NOT NULL AND md.MarketeerName IS NOT NULL
                    THEN 'Event + Promotion + Marketeer'
            END
        ORDER BY SUM(COALESCE(ef.PromotionRevenue, 0) - COALESCE(ef.PromotionCost, 0)) DESC
                ) AS [TPP RankWithinGroup],
        ROUND(AVG(COALESCE(ef.PromotionRevenue, 0) - COALESCE(ef.PromotionCost, 0)), 0) AS [AVG Promotion Profit],
-- Ranks by APP within each grouping set
    RANK() OVER (
        PARTITION BY
            CASE
                    WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NULL AND md.MarketeerName IS NULL THEN 'Event'
                    WHEN pd.PromotionType IS NOT NULL AND ed.EventName IS NULL AND md.MarketeerName IS NULL THEN 'PromotionType'
                    WHEN md.MarketeerName IS NOT NULL AND ed.EventName IS NULL AND pd.PromotionType IS NULL THEN 'Marketeer'
                    WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NOT NULL AND md.MarketeerName IS NULL
                    THEN 'Event + Promotion'
                    WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND md.MarketeerName IS NOT NULL AND pd.PromotionType IS NULL
                    THEN 'Event + Marketeer'
                    WHEN pd.PromotionType IS NOT NULL AND md.MarketeerName IS NOT NULL AND ed.EventName IS NULL THEN 'PromotionType + Marketeer'
                    WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NOT NULL AND md.MarketeerName IS NOT NULL
                    THEN 'Event + Promotion + Marketeer'
            END
        ORDER BY AVG(COALESCE(ef.PromotionRevenue, 0) - COALESCE(ef.PromotionCost, 0)) DESC
                ) AS [APP RankWithinGroup]
  FROM
      EventDim ed
      JOIN EventFact ef ON ed.EventID = ef.EventID
      JOIN PromotionDim pd ON ef.PromotionID = pd.PromotionID
      JOIN MarketeerDim md ON pd.MarketeerID = md.MarketeerID
  -- Grouping sets for flexible grouping
  GROUP BY GROUPING SETS (
                          (ed.EventName, ed.EventYear),
                          (pd.PromotionType),
                          (md.MarketeerName),
                          (ed.EventName, ed.EventYear, pd.PromotionType),
                          (ed.EventName, ed.EventYear, md.MarketeerName),
                          (pd.PromotionType, md.MarketeerName),
                          (ed.EventName, ed.EventYear, pd.PromotionType, md.MarketeerName)
                          )
  -- Enforcing custom grouping order for clarity
  --
  ORDER BY
      CASE
          WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NULL AND md.MarketeerName IS NULL THEN 1 -- Event Name
          WHEN pd.PromotionType IS NOT NULL AND ed.EventName IS NULL AND md.MarketeerName IS NULL THEN 2 -- PromotionType
          WHEN md.MarketeerName IS NOT NULL AND ed.EventName IS NULL AND pd.PromotionType IS NULL THEN 3 -- Marketeer
          WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NOT NULL AND md.MarketeerName IS NULL THEN 4 -- Event + Promotion
          WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND md.MarketeerName IS NOT NULL AND pd.PromotionType IS NULL THEN 5 -- Event + Marketeer
          WHEN pd.PromotionType IS NOT NULL AND md.MarketeerName IS NOT NULL AND ed.EventName IS NULL THEN 6 -- PromotionType + Marketeer
          WHEN ed.EventName IS NOT NULL AND ed.EventYear IS NOT NULL AND pd.PromotionType IS NOT NULL AND md.MarketeerName IS NOT NULL
          THEN 7 -- Event + Promotion + Marketeer
      END, [TPP RankWithinGroup]
  ;
```

The query above highlights the profitability of promotions from multiple dimensions to help key decision makers optimize marketing spend, identify successful campaigns, and focus on high-performing promotions, promotion types, marketers, and combinations of both when planning upcoming events.
Promotion profit is calculated by subtracting PromotionCost from PromotionRevenue in the EventFact table. The COALESCE function ensures that the query effectively handles null values in both columns. The profit is then aggregated and ranked by various grouping dimensions. The use of GROUPING SETS provides detailed and

flexible insight into the total and average profit by Event Name, Promotion Type, Marketeer Name, and all possible combinations between the three groups. Unlike simple RANK() functions, the PARTITION BY clause had to be defined using the CASE WHEN statement to ensure that each record is ranked in the appropriate group, in descending order of both Total Promotion Profit (TPP) and Average Promotion Profit (APP). However, the row orders are displayed using the TPP. Figure 1 above shows the query, while Figure 2 and Figure 3 below show snippets of the results.

Figure 2

Query1 Result Snippet a

	Event Name	PromotionType	MarketeerName	Total Promotion Profit	TPP RankWithinGroup	AVG Promotion Profit	APP RankWithinGroup
1	MSI-2018	[NULL]	[NULL]	2142917	1	21429	1
2	WORLDS-2019	[NULL]	[NULL]	2009275	2	21375	2
3	WORLDS-2017	[NULL]	[NULL]	1830330	3	18488	5
4	MSI-2016	[NULL]	[NULL]	1791911	4	18100	7
5	MSI-2017	[NULL]	[NULL]	1731321	5	18224	6
6	MSI-2021	[NULL]	[NULL]	1683176	6	17352	8
7	WORLDS-2016	[NULL]	[NULL]	1625431	7	18683	4
8	MSI-2019	[NULL]	[NULL]	1616080	8	19239	3
9	WORLDS-2020	[NULL]	[NULL]	1434440	9	17282	9
10	WORLDS-2018	[NULL]	[NULL]	1240160	10	15502	10
11	-	Public Relations	[NULL]	3825828	1	19721	2
12	-	Direct Marketing	[NULL]	3279267	2	17822	5
13	-	Digital Promotions	[NULL]	2649464	3	18399	4
14	-	Sponsorships	[NULL]	2498378	4	19072	3
15	-	Sales Promotion	[NULL]	2448610	5	16771	6
16	-	General Advertising	[NULL]	2403494	6	20197	1
17	-	[NULL]	Wikizz	1724946	1	19166	15
18	-	[NULL]	Shufflebeat	1076430	2	17362	24
19	-	[NULL]	Mynte	840267	3	21007	7
20	-	[NULL]	Dabfeed	798013	4	19464	12
21	-	[NULL]	Linkbridge	726784	5	20765	8
22	-	[NULL]	Divape	641599	6	18871	16
23	-	[NULL]	Bluezoom	633898	7	19809	11
24	-	[NULL]	Trilia	615570	8	19857	10
25	-	[NULL]	Leexo	599335	9	19333	14
26	-	[NULL]	Feednation	591449	10	21123	6
27	-	[NULL]	Zooxo	566419	11	18272	20

Figure 2 shows the different groups ranked in order of total profit. We can also see the difference in the ranks in terms of TPP and APP, for example, in row 16 we see that General Advertising has the lowest total promotion profit, but performs best on average.

Figure 3

Query1 *Result Snippet b*

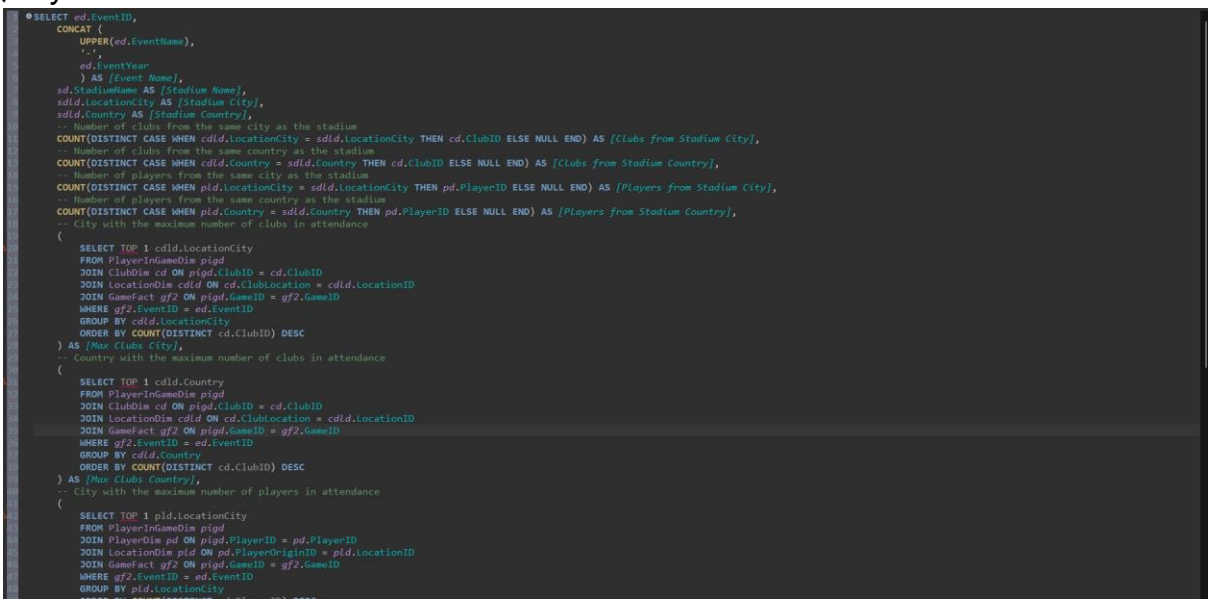
	Event Name	PromotionType	MarketeerName	Total Promotion Profit	TPP RankWithinGroup	AVG Promotion Profit	APP RankWithinGroup
103	WORLDS-2016	General Advertising	[NULL]	184965	52	26424	2
104	WORLDS-2020	Public Relations	[NULL]	181351	53	13950	53
105	MSI-2019	Sponsorships	[NULL]	147015	54	12251	59
106	MSI-2017	General Advertising	[NULL]	139879	55	23313	7
107	WORLDS-2018	General Advertising	[NULL]	138960	56	19851	23
108	MSI-2021	Digital Promotions	[NULL]	134987	57	12272	58
109	MSI-2019	General Advertising	[NULL]	131172	58	18739	31
110	WORLDS-2018	Digital Promotions	[NULL]	88961	59	14827	49
111	WORLDS-2020	Sales Promotion	[NULL]	73710	60	7371	60
112	MSI-2021	[NULL]	Wikizz	256496	1	23318	88
113	MSI-2018	[NULL]	Mynte	238025	2	26447	50
114	MSI-2018	[NULL]	Wikizz	225556	3	22556	93
115	WORLDS-2018	[NULL]	Wikizz	225139	4	20467	117
116	MSI-2016	[NULL]	Wikizz	215545	5	15396	196
117	WORLDS-2019	[NULL]	Wikizz	213990	6	21399	108
118	MSI-2017	[NULL]	Shufflebeat	170750	7	18972	140
119	MSI-2019	[NULL]	Mynte	156212	8	26035	52
120	WORLDS-2017	[NULL]	Wikizz	155883	9	15588	193
121	MSI-2019	[NULL]	Wikizz	152399	10	19050	139
122	MSI-2017	[NULL]	Wikizz	152241	11	16916	176
123	MSI-2018	[NULL]	Linkbridge	150144	12	25024	62

Figure 3 highlights the least performing promotion types and the best the Marketeers across the events.

b. Query 2: Club and Player Attendance by Location.

Figure 4

Query 2

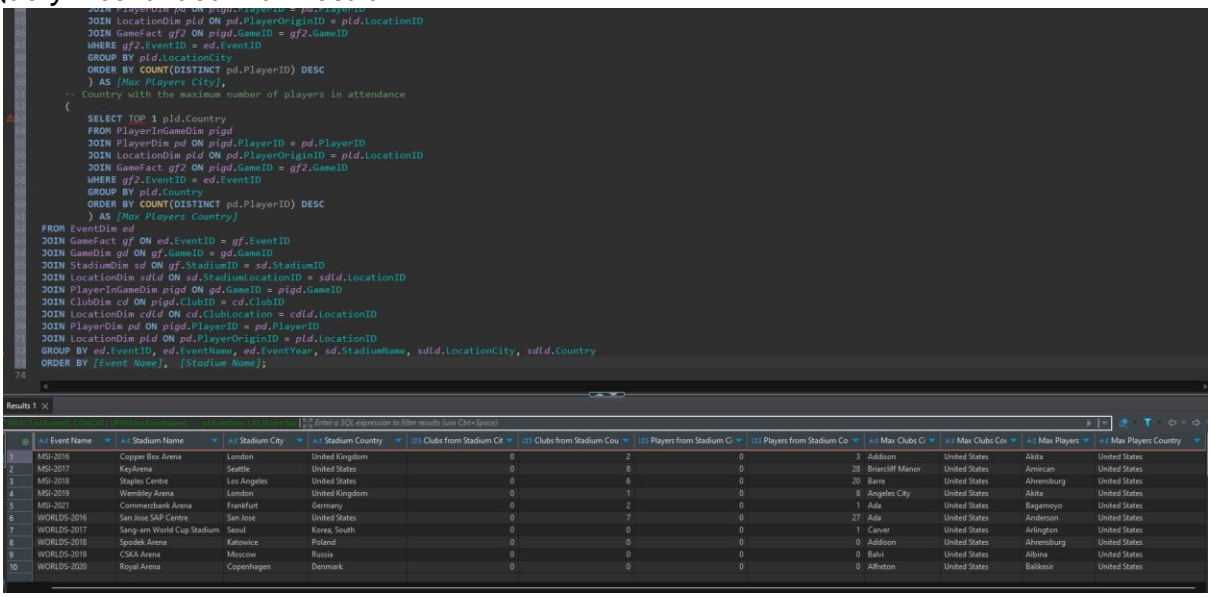


```
SELECT ed.EventID,
    CONCAT (
        UPPER(ed.EventName),
        '-',
        ed.EventYear
        ) AS [Event Name],
    sd.StadiumName AS [Stadium Name],
    sdld.LocationCity AS [Stadium City],
    sdld.Country AS [Stadium Country],
    -- Number of clubs from the same city as the stadium
    COUNT(DISTINCT CASE WHEN cdld.LocationCity = sdld.LocationCity THEN cd.ClubID ELSE NULL END) AS [Clubs from Stadium City],
    -- Number of clubs from the same country as the stadium
    COUNT(DISTINCT CASE WHEN cdld.Country = sdld.Country THEN cd.ClubID ELSE NULL END) AS [Clubs from Stadium Country],
    -- Number of players from the same city as the stadium
    COUNT(DISTINCT CASE WHEN pld.LocationCity = sdld.LocationCity THEN pd.PlayerID ELSE NULL END) AS [Players from Stadium City],
    -- Number of players from the same country as the stadium
    COUNT(DISTINCT CASE WHEN pld.Country = sdld.Country THEN pd.PlayerID ELSE NULL END) AS [Players from Stadium Country],
    -- City with the maximum number of clubs in attendance
    (
        SELECT TOP 1 cdld.LocationCity
        FROM PlayerInGameDim pigd
        JOIN ClubDim cd ON pigd.ClubID = cd.ClubID
        JOIN LocationDim cdld ON cd.ClubLocation = cdld.LocationID
        JOIN GameFact gf2 ON pigd.GameID = gf2.GameID
        WHERE gf2.EventID = ed.EventID
        GROUP BY cdld.LocationCity
        ORDER BY COUNT(DISTINCT cd.ClubID) DESC
    ) AS [Max Clubs City],
    -- Country with the maximum number of clubs in attendance
    (
        SELECT TOP 1 cdld.Country
        FROM PlayerInGameDim pigd
        JOIN ClubDim cd ON pigd.ClubID = cd.ClubID
        JOIN LocationDim cdld ON cd.ClubLocation = cdld.LocationID
        JOIN GameFact gf2 ON pigd.GameID = gf2.GameID
        WHERE gf2.EventID = ed.EventID
        GROUP BY cdld.Country
        ORDER BY COUNT(DISTINCT cd.ClubID) DESC
    ) AS [Max Clubs Country],
    -- City with the maximum number of players in attendance
    (
        SELECT TOP 1 pld.LocationCity
        FROM PlayerInGameDim pigd
        JOIN PlayerDim pd ON pigd.PlayerID = pd.PlayerID
        JOIN LocationDim pld ON pd.PlayerOriginID = pld.LocationID
        JOIN GameFact gf2 ON pigd.GameID = gf2.GameID
        WHERE gf2.EventID = ed.EventID
        GROUP BY pld.LocationCity
```

This query provides insights into the geographic distribution of clubs and players attending World Championships, assisting Tior Games in optimising event planning and marketing strategies. It calculates the number of clubs and players from the same city and country as the stadium, offering insights into local representation. To identify the top contributing regions, the query uses subqueries to accurately determine the city and country with the highest club and player attendance. The results are then grouped by event, stadium, and location, thus offering a clear view of attendance patterns. The idea behind this analysis is to assist Tior Games in targeting future events in high-performing regions, focusing recruitment efforts on talent-rich areas, and optimising marketing campaigns. The query's flexible aggregation and ranking logic ensures accurate insights, empowering Tior Games to make data-driven decisions that enhance the championship's popularity, player engagement, and overall success. Figures 4 and 5 show the query and the results, respectively.

Figure 5

Query 2 continued with Result



```
        JOIN LocationDim pld ON pd.PlayerOriginID = pld.LocationID
        JOIN GameFact gf2 ON pigd.GameID = gf2.GameID
        WHERE gf2.EventID = ed.EventID
        GROUP BY pld.LocationCity
        ORDER BY COUNT(DISTINCT pd.PlayerID) DESC
        ) AS [Max Players City],
    -- Country with the maximum number of players in attendance
    (
        SELECT TOP 1 pld.Country
        FROM PlayerInGameDim pigd
        JOIN PlayerDim pd ON pigd.PlayerID = pd.PlayerID
        JOIN LocationDim pld ON pd.PlayerOriginID = pld.LocationID
        JOIN GameFact gf2 ON pigd.GameID = gf2.GameID
        WHERE gf2.EventID = ed.EventID
        GROUP BY pld.Country
        ORDER BY COUNT(DISTINCT pd.PlayerID) DESC
        ) AS [Max Players Country]
FROM EventDim ed
JOIN GameFact gf ON ed.EventID = gf.EventID
JOIN GameDim gd ON gf.GameID = gd.GameID
JOIN StadiumDim sd ON gf.StadiumID = sd.StadiumID
JOIN LocationDim sdld ON sd.StadiumLocationID = sdld.LocationID
JOIN PlayerInGameDim pigd ON gd.GameID = pigd.GameID
JOIN ClubDim cd ON pigd.ClubID = cd.ClubID
JOIN LocationDim cdld ON cd.ClubLocation = cdld.LocationID
JOIN PlayerDim pd ON pigd.PlayerID = pd.PlayerID
JOIN LocationDim pld ON pd.PlayerOriginID = pld.LocationID
GROUP BY ed.EventID, ed.EventName, ed.EventYear, sd.StadiumName, sdld.LocationCity, sdld.Country
ORDER BY [Event Name], [Stadium Name];
```

Results 1

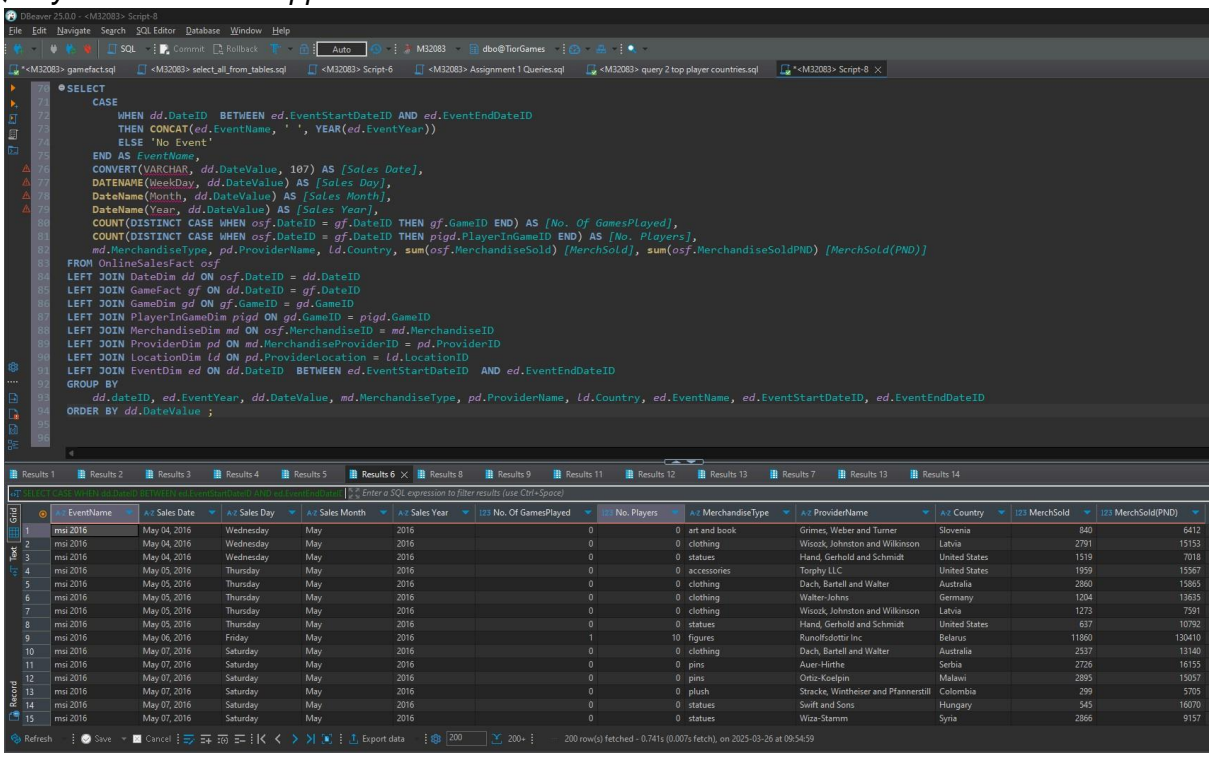
	Event Name	Stadium Name	Stadium City	Stadium Country	Clubs from Stadium Cit	Clubs from Stadium Cou	Players from Stadium Ci	Players from Stadium Co	Max Clubs Ci	Max Clubs Cou	Max Players	Max Players Country
1	MSI-2016	Copper Box Arena	London	United Kingdom	0	2	0	3	Addison	United States	Akita	United States
2	MSI-2017	KeyArena	Seattle	United States	0	8	0	28	Briarcliff Manor	United States	Amircan	United States
3	MSI-2018	Staples Centre	Los Angeles	United States	0	6	0	20	Barre	United States	Ahrensburg	United States
4	MSI-2019	Wembley Arena	London	United Kingdom	0	1	0	8	Angeles City	United States	Akita	United States
5	MSI-2021	Commerzbank Arena	Frankfurt	Germany	0	2	0	1	Ada	United States	Bagamoyo	United States
6	WORLDS-2016	San Jose SAP Centre	San Jose	United States	0	7	0	27	Ada	United States	Anderson	United States
7	WORLDS-2017	Sang-am World Cup Stadium	Seoul	Korea, South	0	0	0	1	Carver	United States	Arlington	United States
8	WORLDS-2018	Spodek Arena	Katowice	Poland	0	0	0	0	Addison	United States	Ahrensburg	United States
9	WORLDS-2019	CSKA Arena	Moscow	Russia	0	0	0	0	Balvi	United States	Albina	United States
10	WORLDS-2020	Royal Arena	Copenhagen	Denmark	0	0	0	0	Alfreton	United States	Balikesir	United States

However, the results show that irrespective of the event country, the US consistently exhibits the highest level of participation. In some rows, we have no attendance from the club Country and City.

c. Query 3: Merchandise Sales Performance Analysis.

Figure 6

Query 3 and Result Snippet



```
SELECT
    CASE
        WHEN dd.DateID  BETWEEN ed.EventStartDateID AND ed.EventEndDateID
        THEN CONCAT(ed.EventName, ' ', YEAR(ed.EventYear))
        ELSE 'No Event'
    END AS EventName,
    CONVERT(VARCHAR, dd.DateValue, 107) AS [Sales Date],
    DATENAME(WeekDay, dd.DateValue) AS [Sales Day],
    DateName(Month, dd.DateValue) AS [Sales Month],
    DateName(Year, dd.DateValue) AS [Sales Year],
    COUNT(DISTINCT CASE WHEN osf.DateID = gf.DateID THEN gf.GameID END) AS [No. Of GamesPlayed],
    COUNT(DISTINCT CASE WHEN osf.DateID = gf.DateID THEN pigd.PlayerInGameID END) AS [No. Players],
    md.MerchandiseType, pd.ProviderName, ld.Country, sum(osf.MerchandiseSold) [MerchSold], sum(osf.MerchandiseSoldPND) [MerchSold(PND)]
FROM OnlineSalesFact osf
LEFT JOIN DateDim dd ON osf.DateID = dd.DateID
LEFT JOIN GameFact gf ON dd.DateID = gf.DateID
LEFT JOIN GameDim gd ON gf.GameID = gd.GameID
LEFT JOIN PlayerInGameDim pigd ON gd.GameID = pigd.GameID
LEFT JOIN MerchandiseDim md ON osf.MerchandiseID = md.MerchandiseID
LEFT JOIN ProviderDim pd ON md.MerchandiseProviderID = pd.ProviderID
LEFT JOIN LocationDim ld ON pd.ProviderLocation = ld.LocationID
LEFT JOIN EventDim ed ON dd.DateID  BETWEEN ed.EventStartDateID  AND ed.EventEndDateID
GROUP BY
    dd.dateID, ed.EventYear, dd.DateValue, md.MerchandiseType, pd.ProviderName, ld.Country, ed.EventName, ed.EventStartDateID, ed.EventEndDateID
ORDER BY dd.DateValue ;
```

	EventName	Sales Date	Sales Day	Sales Month	Sales Year	No. Of GamesPlayed	No. Players	MerchandiseType	ProviderName	Country	MerchSold	MerchSold(PND)
1	msi 2016	May 04, 2016	Wednesday	May	2016	0	0	art and book	Grimes, Weber and Turner	Slovenia	840	6412
2	msi 2016	May 04, 2016	Wednesday	May	2016	0	0	clothing	Wisozk, Johnston and Wilkinson	Latvia	2791	15153
3	msi 2016	May 04, 2016	Wednesday	May	2016	0	0	statues	Hand, Gerhold and Schmidt	United States	1519	7018
4	msi 2016	May 05, 2016	Thursday	May	2016	0	0	accessories	Torphy LLC	United States	1959	15567
5	msi 2016	May 05, 2016	Thursday	May	2016	0	0	clothing	Dach, Bartell and Walter	Australia	2860	15865
6	msi 2016	May 05, 2016	Thursday	May	2016	0	0	clothing	Walter-Johns	Germany	1204	13635
7	msi 2016	May 05, 2016	Thursday	May	2016	0	0	clothing	Wisozk, Johnston and Wilkinson	Latvia	1273	7591
8	msi 2016	May 05, 2016	Thursday	May	2016	0	0	statues	Hand, Gerhold and Schmidt	United States	637	10792
9	msi 2016	May 06, 2016	Friday	May	2016	1	10	figures	Runolfsdottir Inc	Belarus	11860	130410
10	msi 2016	May 07, 2016	Saturday	May	2016	0	0	clothing	Dach, Bartell and Walter	Australia	2537	13140
11	msi 2016	May 07, 2016	Saturday	May	2016	0	0	pins	Auer-Hirthe	Serbia	2726	16155
12	msi 2016	May 07, 2016	Saturday	May	2016	0	0	pins	Ortiz-Koelpin	Malawi	2895	15057
13	msi 2016	May 07, 2016	Saturday	May	2016	0	0	plush	Stracke, Wintheiser and Pfannerstill	Colombia	299	5705
14	msi 2016	May 07, 2016	Saturday	May	2016	0	0	statues	Swift and Sons	Hungary	545	16070
15	msi 2016	May 07, 2016	Saturday	May	2016	0	0	statues	Wiza-Stamm	Syria	2866	9157

The query above is designed to provide insights into merchandise sales performance based on the date of sales. It allows us to analyse how merchandise sales vary across different events, months, years, merchandise types, and merchandise providers. One of the key objectives of the query is to determine whether sales occur on dates outside of events. To achieve this, LEFT JOINs are used, ensuring that the query returns sales data even if there are no matching game facts or event dates. However, after reviewing the 952 records in the result, it appears that all online sales occurred during events, although not always on game days. The query also counts the number of players and games to explore potential correlations with merchandise performance — for instance, whether more games or players lead to higher sales. Additionally, it enables us to compare merchandise types and their providers, helping identify which products and suppliers are performing well.

d. Query 4: Comprehensive Event Sales and Refund Analysis.

Query 4 provides insight into the number of refunds for tickets and merchandise for each event. This offers Tior Games key insights into refund trends. The query calculates the refund rate in quantity and pounds. In addition, the query incorporates 'netsales' columns, showing the calculation of the total revenue after sales and refunds. This provides a foundation for event profitability analysis, facilitating the identification of tickets, merchandise, or events that are particularly susceptible to refunds by examining the refund rates.

Figure 7

Query 4 and Result

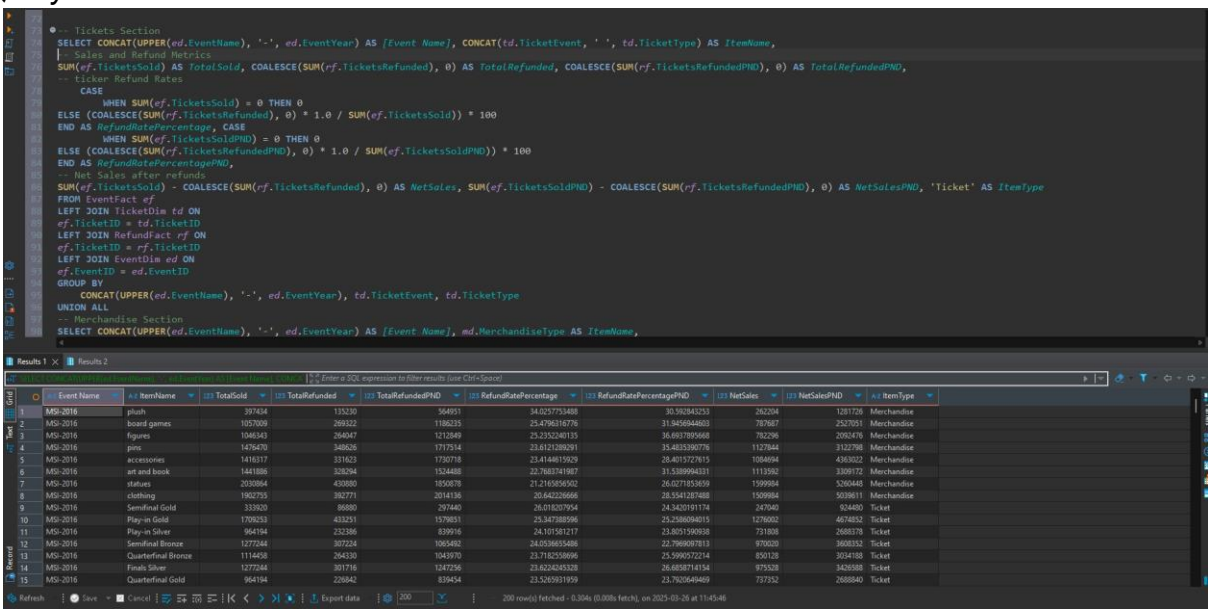
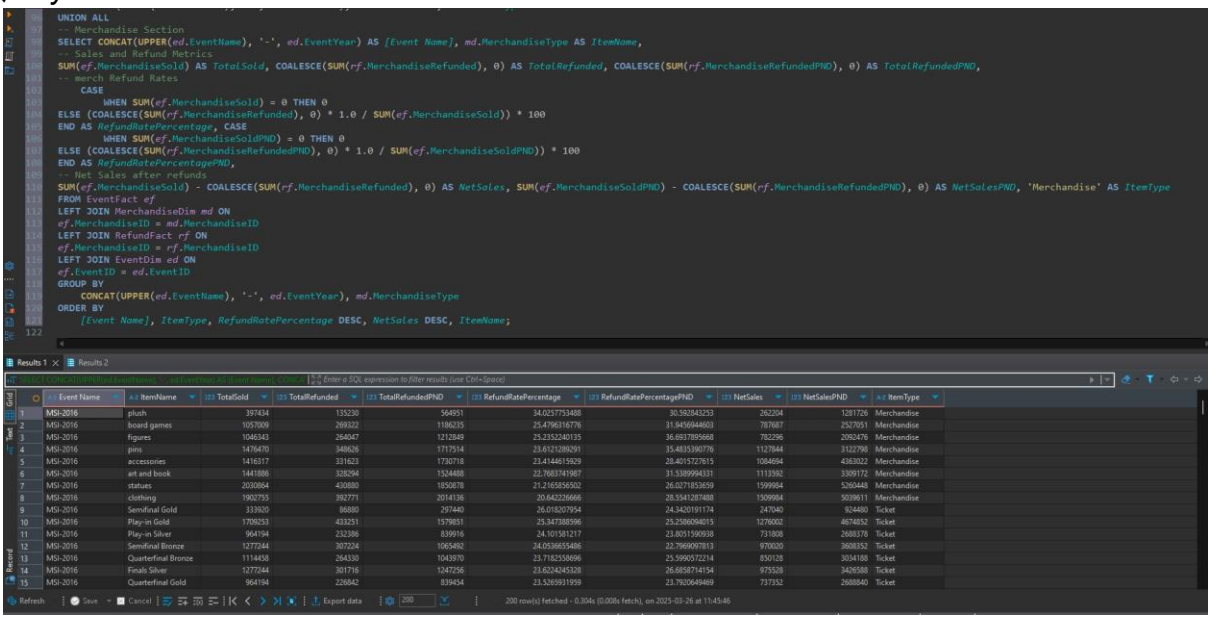


Figure 8

Query 4 and Results b



e. Query 5: Player Performance And Demographic Summary

Figure 9

Query 5 and Results

```
WITH PPaDS AS (
    SELECT
        p.PlayerRealName,
        SUM(pr.PRKills) AS TotalKills,
        SUM(pr.PRAssists) AS TotalAssists,
        SUM(pr.PRDeaths) AS TotalDeaths,
        DENSE_RANK() OVER (ORDER BY SUM(pr.PRKills) DESC) AS KillsRank,
        DENSE_RANK() OVER (ORDER BY SUM(pr.PRAssists) DESC) AS AssistsRank,
        DENSE_RANK() OVER (ORDER BY SUM(pr.PRDeaths) ) AS DeathsRank,
        l.Country,
        CASE
            WHEN DATEDIFF(YEAR, p.PlayerDoB, GETDATE()) < 18 THEN 'Under 18'
            WHEN DATEDIFF(YEAR, p.PlayerDoB, GETDATE()) BETWEEN 18 AND 24 THEN '18-24'
            WHEN DATEDIFF(YEAR, p.PlayerDoB, GETDATE()) BETWEEN 25 AND 34 THEN '25-34'
            WHEN DATEDIFF(YEAR, p.PlayerDoB, GETDATE()) BETWEEN 35 AND 44 THEN '35-44'
            ELSE '45+'
        END AS AgeGroup
    FROM
        PlayerDim p
    JOIN PlayerInGameDim pig ON p.PlayerID = pig.PlayerID
    JOIN PersonalRecordDim pr ON pig.PRID = pr.PRID
    JOIN LocationDim l ON p.PlayerOriginID = l.LocationID
    GROUP BY
        p.PlayerRealName, l.Country, p.PlayerDoB
)
SELECT
    PlayerRealName,TotalKills,KillsRank, TotalAssists, AssistsRank, TotalDeaths, DeathsRank, Country, AgeGroup
FROM
    PPaDS
ORDER BY
    KillsRank;
```

Results 1 | Results 2 | Results 3

	PlayerRealName	TotalKills	KillsRank	TotalAssists	AssistsRank	TotalDeaths	DeathsRank	Country	AgeGroup
1	Jemie Ickovici	141	1	230	3	67	68	United Kingdom	18-24
2	Hurley McGarrie	133	2	184	11	111	90	France	18-24
3	Cami Stonebanks	130	3	167	19	161	96	Slovakia	18-24
4	Zoiine Scolland	128	4	271	1	122	94	Mongolia	25-34
5	Aili Forde	128	4	174	15	62	63	Ecuador	25-34
6	Rosa Fillingham	118	5	54	104	84	82	Ukraine	18-24
7	Agnes Rucklesse	117	6	146	29	79	78	Bulgaria	18-24
8	Dieter Bolle	114	7	222	6	108	89	Colombia	25-34
9	Mia Chatin	109	8	153	24	89	85	United States	18-24
10	Baudoin Abram	108	9	132	41	83	81	Australia	18-24
11	Neilla Storck	108	9	141	34	65	66	United Kingdom	18-24

Refresh | Save | Cancel | Export data | 200 | 200+ | 200 row(s) fetched - 0.688s (0.001s fetch), on 2025-03-26 at 13:45:34

The query aggregates player performance data alongside demographic information such as country and age group, providing essential insights into player behavior. The key insights include metrics like total kills, assists, and deaths, which help identify top performers. The TotalKills and Total Assists are ranked in descending order while the TotalDeaths are ranked in ascending order. Demographic analysis reveals regional trends and preferences based on age, aiding in targeted marketing and game design decisions. Linking performance metrics to demographics allows businesses to tailor features and campaigns to specific audiences. In the results, we can observe that the dominant age groups are 18-24 and 25-34.

The query uses a comprehensive approach by combining performance and demographic data for deeper engagement insights. This information enables Tior Games to craft effective marketing campaigns, enhance gameplay based on player preferences, and improve community engagement. Additionally, it aids in identifying high-performing players for retention strategies and optimising resource allocation.

Question 2

Below are the two dimension tables I would suggest for more insights.

1. ProductDim: A dimension table that gives all the information needed about the products Tior Games has ever sold. Below is its data dictionary.

Table 1

ProductDim

Column Name	Data Type	Description
ProductID	INT (PK)	Unique ID for each product
ProductName	VARCHAR(50)	Name of the product
ProductDescription	VARCHAR(255)	Specific details about Merch(White Tior Games T-Shirt with game character printed on the back)
Category	VARCHAR(30)	Product category (loot box, skin, jersey, etc.)
Price	FLOAT	Base price of the product
Currency	VARCHAR(10)	Currency used for the product sale (USD, NGN)
ReleaseDate	DATE	Date when the product was introduced
IsLimitedEdition	BOOLEAN	Flag indicating if the product is limited edition
MerchandiseID	INT (FK)	Foreign key linking to DimMerchandise

Using ProductDim as a bridge dimension table between OnlineSalesFact/RefundFact and MerchandiseDim improves Tior's data warehouse architecture. By separating detailed product information from broader merchandise categories, the schema becomes more normalized and redundancies are eliminated. This additional database allows Tior Games to store product-specific attributes-such as price, currency, and release date, in ProductDim while using MerchandiseDim to represent higher-level categories, such as apparel or figurines, improving data integrity and making the schema more scalable, allowing new products or categories to be added seamlessly without changing the fact table.

From a business perspective, we can also gain deeper insights and more flexible reporting. Tior Games can easily analyse product performance by merchandise category and/or vendor, while tracking sales of limited-edition items and identifying pricing trends across different currencies. Such multi-level aggregation capabilities support more granular trend analysis, such as evaluating the popularity of specific product types (e.g., shirts vs. facecaps) or identifying high revenue merchandise categories and/or vendors. Adding the ProductDim to the db will help Tior Games executives and decision makers gain greater visibility into product sales trends and improve data-driven decisions that can enhance marketing strategies, optimise pricing models, and ultimately increase profitability. Another helpful dimensional table related to sales will be a GameItemDim, which will provide details on perks, skins, gems, etc., that players can use to improve their chances of winning, but this will require a fact table.

2. GameCharacterDim: Another Important dimension table is the GameCharacterDim. As a computer-based Multiplayer Online Battle Arena competition, League of Fun needs to keep track of all its game characters from inception to date. The addition of this dimension table improves the depth of game-related insights. By linking this dimension to PlayerInGameDim, Tior Games can track player performance by character, revealing which characters lead to a higher percentage of kills, assists, or deaths. It also enables detailed player profiling, such as identifying preferred characters, win rates by champion, and playstyle tendencies. The CharacterType and

Special attributes also support meta-analysis, allowing Tior Games to monitor the effectiveness of specific roles (e.g., Tanks vs. Assassins) and assess the impact of character abilities on match outcomes.

Table 2

GameCharacterDim

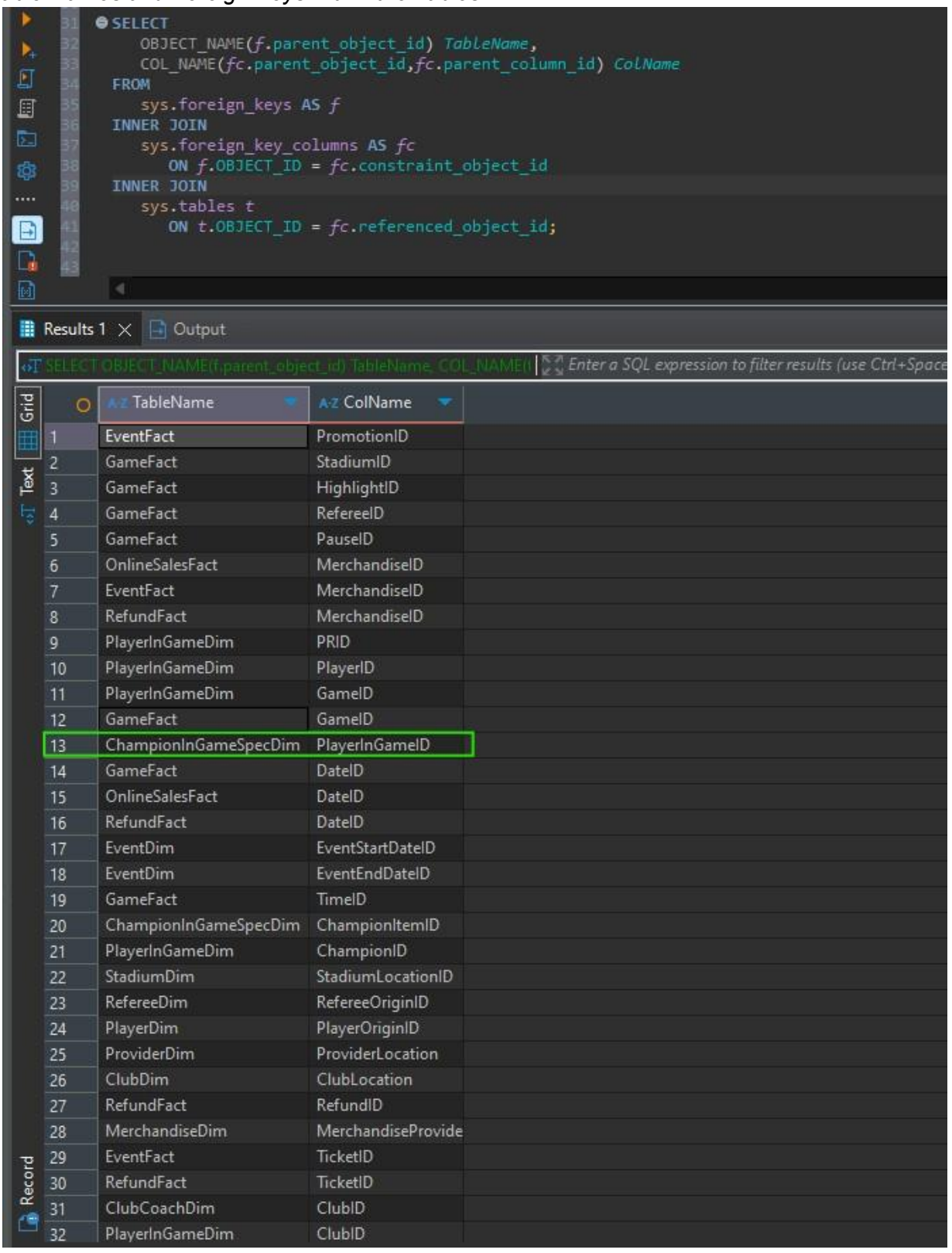
Column Name	Data Type	Description
CharacterID	INT (PK)	Unique ID for each in-game character
CharacterName	VARCHAR(55)	Name of the character
CharacterType	VARCHAR(75)	Character class or role (e.g., Tank, Support, Assassin)
ReleaseDate	DATE	Date the character was introduced in the game
Special Abilities	VARCHAR(255)	Description of unique abilities or powers of the character
BaseHealth	FLOAT	Initial health points of the character
BaseDamage	FLOAT	Initial attack damage of the character

This also aids in gaining insights for game balancing and marketing. For instance, if a particular character is used by a larger percentage of players, it becomes easier to investigate the reasons why using the GameCharacterDim table. This facilitates in-depth analysis and character popularity trends by tracking pick, win, and ban rates, which helps developers fine-tune overpowered or underutilised characters. If character attributes are changed frequently, an additional attribute such as drop-date can be added to monitor modifications over time, allowing for a slowly changing dimensions Type 2 approach, and ensuring that we understand character trends among players.

Figure

10

Table Names and Foreign Keys within the Tables



```
SELECT
    OBJECT_NAME(f.parent_object_id) TableName,
    COL_NAME(fc.parent_object_id,fc.parent_column_id) ColName
FROM
    sys.foreign_keys AS f
INNER JOIN
    sys.foreign_key_columns AS fc
       ON f.OBJECT_ID = fc.constraint_object_id
INNER JOIN
    sys.tables t
       ON t.OBJECT_ID = fc.referenced_object_id;
```

Results 1 | Output

Enter a SQL expression to filter results (use Ctrl+Space

	TableName	ColName
1	EventFact	PromotionID
2	GameFact	StadiumID
3	GameFact	HighlightID
4	GameFact	RefereeID
5	GameFact	PauseID
6	OnlineSalesFact	MerchandiseID
7	EventFact	MerchandiseID
8	RefundFact	MerchandiseID
9	PlayerInGameDim	PRID
10	PlayerInGameDim	PlayerID
11	PlayerInGameDim	GameID
12	GameFact	GameID
13	ChampionInGameSpecDim	PlayerInGameID
14	GameFact	DateID
15	OnlineSalesFact	DateID
16	RefundFact	DateID
17	EventDim	EventStartDateID
18	EventDim	EventEndDateID
19	GameFact	TimeID
20	ChampionInGameSpecDim	ChampionItemID
21	PlayerInGameDim	ChampionID
22	StadiumDim	StadiumLocationID
23	RefereeDim	RefereeOriginID
24	PlayerDim	PlayerOriginID
25	ProviderDim	ProviderLocation
26	ClubDim	ClubLocation
27	RefundFact	RefundID
28	MerchandiseDim	MerchandiseProvide
29	EventFact	TicketID
30	RefundFact	TicketID
31	ClubCoachDim	ClubID
32	PlayerInGameDim	ClubID

Figure

11

ERD Snippet Highlighting PlayerInGameDim and ChampinInGameSpecDim

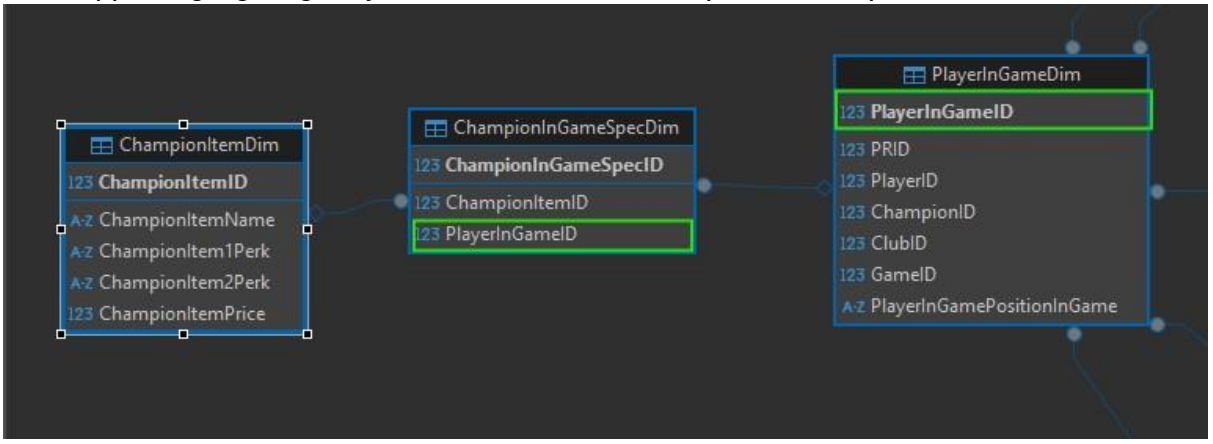
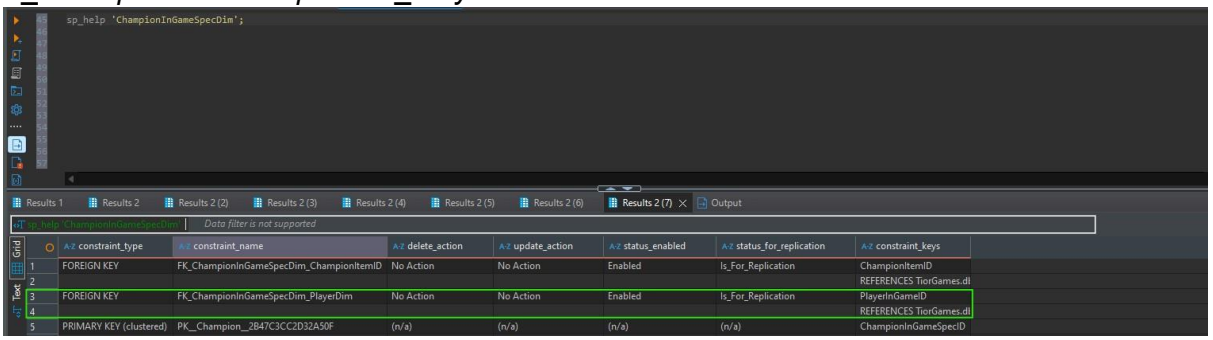


Figure 12

FK_ChampionInGameSpecDim_PlayerDim Constraint

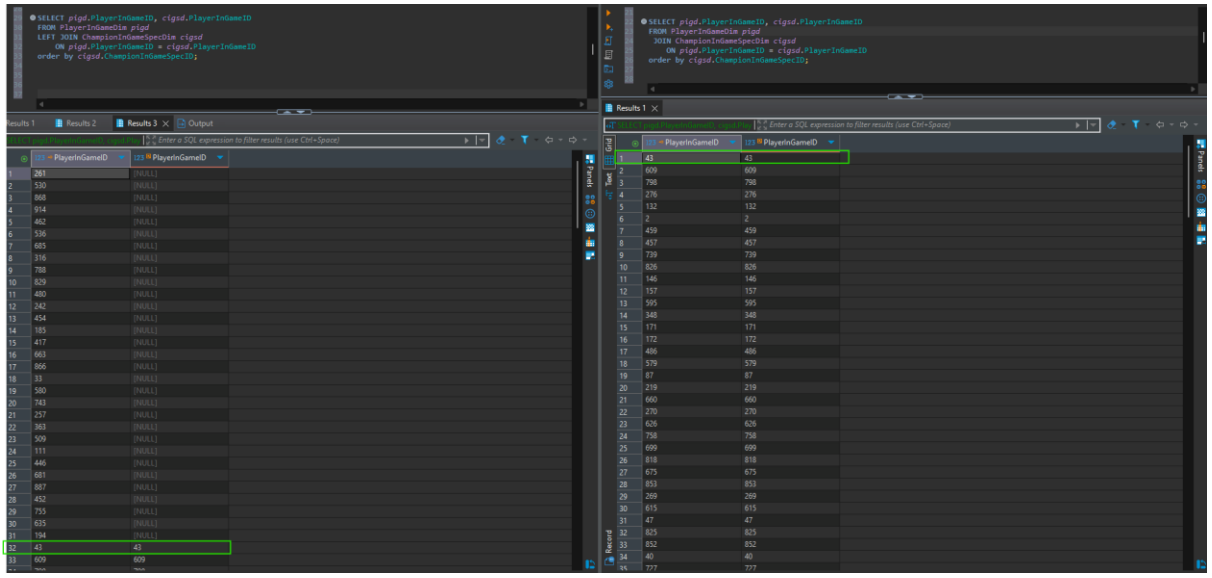


c. To resolve this issue, one option is to modify the code to use an inner join instead of a left join. This would exclude any records that are not present in both tables. Figure 13 illustrates the difference between using a left join and an inner join.

Figure

13

Difference between Left Join and Inner Join



However, a more permanent solution would be to enforce referential integrity, which prevents null values from appearing in a child table like ChampionInGameSpecDim. Additionally, it would be important to define the actions to be taken when insert, update, or delete operations are performed on the parent table, PlayerInGameDim. This could be achieved using triggers or specifying CASCADE, SET DEFAULT, or NO ACTION when creating the foreign key constraint.

References

Sherman, R. (2014). *Business intelligence guidebook: From data integration to analytics*. Newnes.