

BUSINESS ANALYTICS (BA) COURSEWORK REPORT

Do not write your name on your work unless your lecturer has explicitly told you to do so.

Student ID number	Title of degree studying	Level/Year
2200918	Bachelor of Science (Honours) Data Science and Analytics	6

Short unit name:	M32367 - BA				Due date: 10 JUNE 24	Deadline: 10 JUN
Full unit name:	Business Analytics					
Unit lecturer name:	Mr. Kon Hon Sing				Group: (if applicable)	
Additional items e.g. CD/disk/USB:	Yes		No	✓	Details:	

All additional items should be clearly labelled with ID number and unit name and securely attached to your work.

Candidates are reminded that the following are defined as Assessment Offences and will be dealt with in accordance with the University's Code of Student Discipline:

- a) Any attempt to complete an assessment by means considered to be unfair.
- b) Plagiarism, which the University defines as the incorporation by a student in work for assessment of material which is not their own, in the sense that all or a substantial part of the work has been copied without any adequate attempt at attribution or has been incorporated as if it were the student's own work when in fact it is wholly or substantially the work of another person or persons.

Please note: **Group coursework** will be filed under the first Student ID number at the top of the list. Ensure you know all group member's ID numbers.

NB: Coursework not collected will be disposed of six months **after** the hand-in date.

FOR OFFICIAL USE ONLY

Date received/Office stamp	Provisional mark % / Comments

Academic Staff Member

Table of Contents

1. Introduction	3
2. Query Analysis (Q1)	4
2.1 Query 1: Event Revenue and Promotional Analysis	4
2.2 Query 2: Refund Analysis by Ticket and Merchandise Sales	5
2.3 Query 3: Player Performance and Engagement Metrics	6
2.4 Query 4: Event Profitability and Resource Utilization Analysis	7
2.5 Query 5: Coach and Club Performance Evaluation	8
3. Dimension Analysis (Q2)	10
3.1 Proposed Dimension 1: SponsorDim	10
3.2 Proposed Dimension 2: PlayerPerformanceDim	11
4. Issue Response (Q3)	13
5. Join Explanation and Solution (Q4)	13
6. Database Structure Issues (Q5)	15

1. Introduction

In the contemporary era where strategic decision-making is increasingly reliant on data analytics, the role of data warehousing has become crucial. This report delves into the advanced capabilities of data warehousing, emphasizing the use of Online Analytical Processing (OLAP) functions. These functions are critical for effectively managing complex datasets, extracting valuable insights, and providing actionable information that guides strategic planning and operational efficiency.

The report is structured to guide the reader through a comprehensive exploration of data warehousing functionalities, highlighting practical applications and strategic benefits:

1. **Query Analysis (Q1):** This section elaborates on various advanced SQL queries using OLAP functions, displaying their utility in generating meaningful insights for informed decision-making.
 - **Event Revenue and Promotional Analysis**
 - **Refund Analysis by Ticket and Merchandise Sales**
 - **Player Performance and Engagement Metrics**
 - **Event Profitability and Resource Utilization Analysis**
 - **Coach and Club Performance Evaluation**
2. **Dimension Analysis (Q2):** Discusses the integration of new dimensions—SponsorDim and PlayerPerformanceDim—into the data warehouse, enhancing its analytical capabilities with detailed data dictionaries.
3. **Issue Response (Q3):** Addresses specific challenges encountered in data management within the data warehouse, offering code solutions and practical advice for optimizing data integrity and accessibility.
4. **Join Explanation and Solution (Q4):** Explains the occurrence of null values in join operations and provides strategies for managing these values effectively, enhancing the reliability of data retrieval processes.
5. **Database Structure Issues (Q5):** Critically examines the current database structure, identifying potential issues and proposing remedies to improve data coherence and reduce redundancy.

2. Query Analysis (Q1)

2.1 Query 1: Event Revenue and Promotional Analysis

```
SELECT
    ed.EventName,
    ed.EventYear,
    SUM(ef.TicketsSold) AS TotalTicketsSold,
    FORMAT(SUM(ef.TicketsSoldPND), 'N2') AS TotalRevenue, -- Formatted as a string with two decimal
places
    MAX(ef.VIPSpectatorsNumber) AS MaxVIPAttendees,
    ROUND(AVG(ef.PromotionCost), 2) AS AveragePromotionCost,
    FORMAT(SUM(ef.PromotionRevenue), 'N2') AS TotalPromotionRevenue, -- Consistent currency
formatting
    SUM(ef.MerchandiseSold) AS TotalMerchandiseSold,
    FORMAT(SUM(ef.MerchandiseSoldPND), 'N2') AS MerchandiseRevenue, -- Consistent currency formatting
    ROUND(100.0 * SUM(ef.MerchandiseSoldPND) / SUM(ef.TicketsSoldPND), 2) AS
MerchRevenuePercentageOfTotal, -- New metric
    RANK() OVER (PARTITION BY ed.EventYear ORDER BY SUM(ef.TicketsSoldPND) DESC) AS RevenueRank
FROM EventFact ef
JOIN EventDim ed ON ef.EventID = ed.EventID
LEFT JOIN PromotionDim pd ON ef.PromotionID = pd.PromotionID
GROUP BY ed.EventName, ed.EventYear
ORDER BY ed.EventYear DESC, TotalRevenue DESC;
```

Run

Cancel

Disconnect

Change

Database: TiorGames

Estimated Plan

Enable Actual Plan

Parse

Enable SQLCMD

To Notebook

1

-- Query 1: Event Revenue and Promotional Analysis

2

-- Provides a comprehensive financial overview of each event, detailing ticket and merchandise sales, promotional costs, and VIP attendance metrics.

3

SELECT

4

ed.EventName,

5

ed.EventYear,

6

SUM(ef.TicketsSold) AS TotalTicketsSold,

7

FORMAT(SUM(ef.TicketsSoldPND), 'N2') AS TotalRevenue, -- Formatted as a string with two decimal places

8

MAX(ef.VIPspectatorsNumber) AS MaxVIPAttendees,

9

ROUND(AVG(ef.PromotionCost), 2) AS AveragePromotionCost,

10

FORMAT(SUM(ef.PromotionRevenue), 'N2') AS TotalPromotionRevenue, -- Consistent currency formatting

11

SUM(ef.MerchandiseSold) AS TotalMerchandiseSold,

12

FORMAT(SUM(ef.MerchandiseSoldPND), 'N2') AS MerchandiseRevenue, -- Consistent currency formatting

13

ROUND(100.0 * SUM(ef.MerchandiseSoldPND) / SUM(ef.TicketsSoldPND), 2) AS MerchRevenuePercentageOfTotal, -- New metric

14

RANK() OVER (PARTITION BY ed.EventYear ORDER BY SUM(ef.TicketsSoldPND) DESC) AS RevenueRank

15

FROM EventFact ef

16

JOIN EventDim ed ON ef.EventID = ed.EventID

17

LEFT JOIN PromotionDim pd ON ef.PromotionID = pd.PromotionID

18

GROUP BY ed.EventName, ed.EventYear

19

ORDER BY ed.EventYear DESC, TotalRevenue DESC;

20

Results

Messages

	Ev...	Ev...	Tota...	TotalReve...	MaxV...	AverageP...	TotalProm...	TotalMe...	Merchandis...	MerchReve...	RevenueRank
1	msi	2021	200596	740,304.00	141	6028.140000	2,267,906.00	263926	1,007,844.00	136.1400000000...	1
2	Worlds	2020	147574	629,804.00	140	6217.370000	1,950,482.00	239984	831,463.00	132.0200000000...	1
3	Worlds	2019	191008	677,834.00	135	5669.570000	2,542,215.00	272801	887,297.00	130.9000000000...	1
4	msi	2019	158004	627,144.00	144	6363.690000	2,150,630.00	235581	828,901.00	132.1700000000...	2
5	msi	2018	183600	820,300.00	143	6035.180000	2,746,435.00	303572	1,035,165.00	126.1900000000...	1
6	Worlds	2018	140880	667,120.00	135	6778.180000	1,782,414.00	223620	761,712.00	114.1800000000...	2
7	Worlds	2017	172557	754,578.00	144	5788.450000	2,403,387.00	274212	973,483.00	129.0100000000...	1
8	msi	2017	172900	682,765.00	141	6019.740000	2,303,196.00	275946	937,296.00	137.2800000000...	2
9	msi	2016	206613	756,063.00	130	6396.460000	2,425,161.00	269819	963,376.00	127.4200000000...	1

Business Rationale: This query strategically analyses the financial metrics of events to optimize profit generation and assess the effectiveness of marketing strategies. By detailing ticket sales, VIP attendance, and the financial outcomes of promotional activities, it offers a granular view of revenue streams. Event managers are equipped to pinpoint high-yield events, optimize resource allocation, and refine promotional tactics to maximize return on investment. This comprehensive financial analysis is crucial for predictive budgeting and strategic planning, ensuring that events are both financially viable and aligned with broader business goals.

2.2 Query 2: Refund Analysis by Ticket and Merchandise Sales

-- Analyzes refund trends to understand customer satisfaction and financial impacts over time, highlighting changes and anomalies.

```
SELECT
    YEAR(dd.DateValue) AS Year,
    COUNT(rf.RefundID) AS TotalRefundsIssued,
    SUM(rf.TicketsRefundedPND) AS TotalRefundAmountTickets,
    SUM(rf.MerchandiseRefunded) AS TotalMerchandiseRefunded,
    SUM(rf.MerchandiseRefundedPND) AS TotalRefundAmountMerchandise,
    LAG(SUM(rf.TicketsRefundedPND), 1) OVER (ORDER BY YEAR(dd.DateValue)) AS PreviousYearTicketRefunds
FROM RefundFact rf
JOIN DateDim dd ON rf.DateID = dd.DateID
GROUP BY YEAR(dd.DateValue)
ORDER BY Year DESC;
```

21	-- Query 2: Refund Analysis by Ticket and Merchandise Sales
22	-- Analyzes refund trends to understand customer satisfaction and financial impacts over time, highlighting changes and anomalies.
23	SELECT
24	YEAR(dd.DateValue) AS Year,
25	COUNT(rf.RefundID) AS TotalRefundsIssued,
26	SUM(rf.TicketsRefundedPND) AS TotalRefundAmountTickets,
27	SUM(rf.MerchandiseRefunded) AS TotalMerchandiseRefunded,
28	SUM(rf.MerchandiseRefundedPND) AS TotalRefundAmountMerchandise,
29	LAG(SUM(rf.TicketsRefundedPND), 1) OVER (ORDER BY YEAR(dd.DateValue)) AS PreviousYearTicketRefunds
30	FROM RefundFact rf
31	JOIN DateDim dd ON rf.DateID = dd.DateID
32	GROUP BY YEAR(dd.DateValue)
33	ORDER BY Year DESC;
34	

Results Messages

Year	TotalRefundsIssued	TotalRefundAmountTickets	TotalMerchandiseRefunded	TotalRefundAmountMerchandise	PreviousYearTicketRefunds
2021	124	229317	80936	363763	218024
2020	120	218024	74117	341732	456524
2019	248	456524	161440	760708	456176
2018	248	456176	167031	720131	439885
2017	240	439885	147103	697721	447329
2016	244	447329	152491	759864	NULL

Business Rationale:

This query evaluates the financial impact of refunds on the organization, serving as a barometer for customer satisfaction and operational success. By tracking the volume and value of ticket and merchandise refunds year-over-year, it allows financial analysts to detect emerging trends, identify areas of concern, and adjust policies to enhance consumer satisfaction. This proactive approach not only mitigates financial risk but also supports efforts to enhance brand loyalty and customer retention through improved product offerings and event experiences.

2.3 Query 3: Player Performance and Engagement Metrics

-- Provides detailed performance metrics for players and benchmarks against average performances.
SELECT

```
    p.PlayerGameName,  
    COUNT(DISTINCT g.GameID) AS GamesPlayed,  
    SUM(pr.PRAssists) AS TotalAssists,  
    SUM(pr.PRDeaths) AS TotalDeaths,  
    SUM(pr.PRKills) AS TotalKills,  
    AVG(pr.PRDeaths) AS AvgDeathsPerGame,  
    ROUND(PERCENT_RANK() OVER (ORDER BY SUM(pr.PRAssists) DESC),2) AS AssistPercentileRank  
FROM PlayerInGameDim pg  
JOIN PlayerDim p ON pg.PlayerID = p.PlayerID  
JOIN PersonalRecordDim pr ON pg.PRID = pr.PRID  
JOIN GameDim g ON pg.GameID = g.GameID  
GROUP BY p.PlayerGameName  
ORDER BY TotalKills DESC, AvgDeathsPerGame;
```

```

36 -- Query 3: Player Performance and Engagement Metrics
37 -- Provides detailed performance metrics for players and benchmarks against average performances.
38 SELECT
39     p.PlayerGameName,
40     COUNT(DISTINCT g.GameID) AS GamesPlayed,
41     SUM(pr.PRAssists) AS TotalAssists,
42     SUM(pr.PRDeaths) AS TotalDeaths,
43     SUM(pr.PRKills) AS TotalKills,
44     AVG(pr.PRDeaths) AS AvgDeathsPerGame,
45     ROUND(PERCENT_RANK() OVER (ORDER BY SUM(pr.PRAssists) DESC),2) AS AssistPercentileRank
46 FROM PlayerInGameDim pg
47 JOIN PlayerDim p ON pg.PlayerID = p.PlayerID
48 JOIN PersonalRecordDim pr ON pg.PRID = pr.PRID
49 JOIN GameDim g ON pg.GameID = g.GameID
50 GROUP BY p.PlayerGameName
51 ORDER BY TotalKills DESC, AvgDeathsPerGame;
52

```

Results

Messages

	PlayerGameName	GamesPlayed	TotalAssists	TotalDeaths	TotalKills	AvgDeathsPerGame	AssistPercentileRank
1	jickovici3j	7	230	67	141	9	0.01
2	hmcgarrie7b	6	184	111	133	18	0.03
3	cstonebanks7s	8	167	161	130	20	0.06
4	aforde8y	7	174	62	128	8	0.04
5	zscolland4h	7	271	122	128	15	0
6	rfillingham39	4	54	84	118	21	0.64
7	arucklessek	4	146	79	117	19	0.12
8	dbolle9a	5	222	108	114	18	0.02
9	mchatin5x	3	153	89	109	17	0.09
10	nstorck6l	5	141	65	108	13	0.14

Business Rationale:

This query leverages detailed performance data to empower coaches and sports analysts with actionable insights for player development and team strategy optimization. By examining metrics such as games played, assists, kills, and deaths per game, it facilitates the identification of strengths and areas for improvement. The inclusion of percentile ranks for assists enables benchmarking against peer performances, guiding targeted training programs and strategic game play adjustments that aim to elevate overall team performance and competitive positioning.

2.4 Query 4: Event Profitability and Resource Utilization Analysis

```
-- Evaluates event profitability and calculates break-even points to optimize financial planning.
SELECT
    e.EventName,
    e.EventYear,
    COUNT(DISTINCT ef.TicketID) AS TotalTicketsSold,
    SUM(ef.TicketsSoldPND) AS TicketRevenue,
    SUM(ef.PromotionCost) AS TotalPromotionCost,
    SUM(ef.PromotionRevenue) AS PromotionRevenue,
    (SUM(ef.TicketsSoldPND) + SUM(ef.PromotionRevenue) - SUM(ef.PromotionCost)) AS NetEventProfit,
    ROUND((SUM(ef.PromotionCost) / COUNT(DISTINCT ef.TicketID)),2) AS AvgCostPerTicket
FROM EventFact ef
JOIN EventDim e ON ef.EventID = e.EventID
GROUP BY e.EventName, e.EventYear
ORDER BY e.EventYear DESC, NetEventProfit DESC;
```

```

56 -- Query 4: Event Profitability and Resource Utilization Analysis
57 -- Evaluates event profitability and calculates break-even points to optimize financial planning.
58 SELECT
59     e.EventName,
60     e.EventYear,
61     COUNT(DISTINCT ef.TicketID) AS TotalTicketsSold,
62     SUM(ef.TicketsSoldPND) AS TicketRevenue,
63     SUM(ef.PromotionCost) AS TotalPromotionCost,
64     SUM(ef.PromotionRevenue) AS PromotionRevenue,
65     (SUM(ef.TicketsSoldPND) + SUM(ef.PromotionRevenue) - SUM(ef.PromotionCost)) AS NetEventProfit,
66     ROUND((SUM(ef.PromotionCost) / COUNT(DISTINCT ef.TicketID)),2) AS AvgCostPerTicket
67 FROM EventFact ef
68 JOIN EventDim e ON ef.EventID = e.EventID
69 GROUP BY e.EventName, e.EventYear
70 ORDER BY e.EventYear DESC, NetEventProfit DESC;
71

```

Results

Messages

	EventName	EventYear	TotalTicketsSold	TicketRevenue	TotalPromotionCost	PromotionRevenue	NetEventProfit	AvgCostPerTicket
1	msi	2021	15	740304	584730.00	2267906.00	2423480.00	38982.000000
2	Worlds	2020	15	629804	516042.00	1950482.00	2064244.00	34402.800000
3	Worlds	2019	15	677834	532940.00	2542215.00	2687109.00	35529.330000
4	msi	2019	15	627144	534550.00	2150630.00	2243224.00	35636.670000
5	msi	2018	15	820300	603518.00	2746435.00	2963217.00	40234.530000
6	Worlds	2018	15	667120	542254.00	1782414.00	1907280.00	36150.270000
7	Worlds	2017	15	754578	573057.00	2403387.00	2584908.00	38203.800000
8	msi	2017	15	682765	571875.00	2303196.00	2414086.00	38125.000000
9	msi	2016	15	756063	633250.00	2425161.00	2547974.00	42216.670000
10	Worlds	2016	15	632577	493855.00	2119286.00	2258008.00	32923.670000

Business Rationale:

This query conducts a deep dive into the profitability of events, focusing on optimizing financial outcomes by analyzing revenues against promotional and operational costs. It provides event planners with a clear breakdown of financial efficiency, highlighting profitable ventures and cost-intensive activities. By calculating average costs per ticket and net event profits, it aids in fine-tuning pricing strategies and promotional spending, ensuring that each event is not only a cultural success but also a financial boon. This strategic financial oversight is essential for sustaining long-term profitability and growth in the competitive event landscape.

2.5 Query 5: Coach and Club Performance Evaluation

```

SELECT
    CoachName,
    ClubName,
    AvgAssists,
    AvgDeaths,
    TimesCoached,
    DENSE_RANK() OVER (
        PARTITION BY ClubName
        ORDER BY AvgAssists DESC, AvgDeaths ASC, TimesCoached DESC
    ) AS RankInClub
FROM (
    SELECT
        c.CoachName,
        cl.ClubName,
        AVG(pr.PRAssists) AS AvgAssists,
        AVG(pr.PRDeaths) AS AvgDeaths,
        COUNT(cc.ClubCoachID) AS TimesCoached
    FROM ClubCoachDim cc
    JOIN CoachDim c ON cc.CoachID = c.CoachID
    JOIN ClubDim cl ON cc.ClubID = cl.ClubID
    JOIN PlayerInGameDim pg ON cl.ClubID = pg.ClubID
    JOIN PersonalRecordDim pr ON pg.PRID = pr.PRID
    GROUP BY c.CoachName, cl.ClubName
) AS DerivedTable
ORDER BY ClubName, RankInClub;

```

```

74 -- Query 5: Coach and Club Performance Evaluation
75 -- Analyzes coaching effectiveness and compares performance across different clubs with handling ties in
76 SELECT
77     CoachName,ClubName,AvgAssists,AvgDeaths,TimesCoached,
78     DENSE_RANK() OVER (PARTITION BY ClubName ORDER BY AvgAssists DESC, AvgDeaths ASC, TimesCoached DESC)
79 FROM (SELECT c.CoachName,cl.ClubName,
80         AVG(pr.PRAssists) AS AvgAssists,
81         AVG(pr.PRDeaths) AS AvgDeaths,
82         COUNT(cc.ClubCoachID) AS TimesCoached
83     FROM ClubCoachDim cc
84     JOIN CoachDim c ON cc.CoachID = c.CoachID
85     JOIN ClubDim cl ON cc.ClubID = cl.ClubID
86     JOIN PlayerInGameDim pg ON cl.ClubID = pg.ClubID
87     JOIN PersonalRecordDim pr ON pg.PRID = pr.PRID
88     GROUP BY c.CoachName, cl.ClubName
89 ) AS DerivedTable
90 ORDER BY ClubName, RankInClub;
91

```

Results

Messages

	CoachName	ClubName	AvgAssists	AvgDeaths	TimesCoached	RankInClub
155	Marissa Steventon	Voltsillam	31	16	19	1
156	Claire Di Frisco	Voltsillam	31	16	19	1
157	Raeann Cuss	Voltsillam	31	16	19	1
158	Rebekkah Elnor	Voyatouch	26	16	20	1
159	Patsy Turl	Voyatouch	24	15	25	2
160	Tadeo Stuther	Voyatouch	23	14	15	3
161	Alberik O'Fielly	Voyatouch	23	14	15	3
162	Billye Hair	Voyatouch	15	12	5	4
163	Linda Elby	Voyatouch	15	12	5	4
164	Dov Livingston	Voyatouch	15	12	5	4
165	Florella Dedden	Voyatouch	15	12	5	4

Business Rationale:

This query critically assesses coaching effectiveness across different clubs, offering a data-driven foundation for human resource decisions in sports management. By ranking coaches within clubs based on their impact on player performance metrics like assists and deaths, it highlights coaching excellence and areas needing improvement. This ranking system not only fosters a competitive spirit among the coaching staff but also aids in strategic decision-making regarding coach training, recruitment, and retention. Ultimately, this contributes to enhancing club performance, nurturing talent, and maintaining a high standard of professional sportsmanship.

3. Dimension Analysis (Q2)

3.1 Proposed Dimension 1: SponsorDim

Dimension Details:

The SponsorDim is designed to systematically track and analyze data related to sponsors who financially or materially support events and players. It is integrated with the EventDim via the EventID attribute, facilitating the assessment of a sponsor's influence and financial impact on specific events. This dimension can also potentially integrate with financial and marketing modules to evaluate the broader implications of sponsorships.

Data Dictionary:

SponsorDim

Field Name	Data Type	Description
SponsorID	int (PK)	A unique identifier for each sponsor.
SponsorName	varchar(50)	The name of the sponsoring entity, be it a company or an individual.
SponsorType	varchar(50)	Classification of the sponsor, such as Financial, Technical, or Service.
SponsorStartDate	date	The date when the sponsorship agreement begins.
SponsorEndDate	date	The termination date of the sponsorship agreement if it ends.
SponsorshipValuePND	int	The financial contribution of the sponsorship, measured in pounds.
EventID	int (FK)	A foreign key that links to the EventDim to associate sponsors with specific events.

Business Rationale:

The SponsorDim enhances strategic financial planning and marketing decision-making by allowing the organization to track and analyze the duration, type, and value of each sponsorship. It supports advanced analytics for assessing sponsorship trends over time and predicting future sponsorship contributions. This dimension is vital for optimizing sponsorship strategies to maximize revenue and enhance brand visibility through targeted marketing campaigns, making it an invaluable asset for comprehensive business analysis.

3.2 Proposed Dimension 2: PlayerPerformanceDim

Dimension Details:

The PlayerPerformanceDim captures intricate performance metrics of players across different games, linking directly with the GameDim through the GameID and PlayerDim via PlayerID. This dimension facilitates detailed analysis of player performance trends, effectiveness in various game stages, and overall impact on team success.

Data Dictionary:

PlayerPerformanceDim

Field Name	Data Type	Description
PerformanceID	int (PK)	Unique identifier for each performance record.
PlayerID	int (FK)	Connects to PlayerDim, identifying individual players.
GameID	int (FK)	Links to GameDim, specifying the exact match.
PerformanceScore	int	A quantitative score derived from various performance metrics.
PerformanceRank	int	The rank of a player among peers in the same game based on performance.
ImprovementScore	decimal(10, 2)	A measure indicating the player's performance improvement over time.

Business Rationale:

The PlayerPerformanceDim is crucial for enabling detailed insights into player development and game strategy effectiveness. By providing a quantitative analysis of performance metrics linked to specific games, this dimension aids coaches, team analysts, and sports strategists in identifying high-performing talents and areas needing improvement. It also enhances player valuation for contract negotiations and sponsorship deals. Additionally, this dimension enriches fan engagement by highlighting player achievements and trends during broadcasts and in digital content.

4. Issue Response (Q3)

A new fact table is essential to centralize and integrate specific transactional data such as the exact times of ticket purchases, the number of tickets bought, and detailed reasons for ticket returns across various dimensions including TimeDim, DateDim, and Refund. This integration will enable comprehensive analysis of spectator behavior across all events, something that our current disjointed data structure does not support.

5. Join Explanation and Solution (Q4)

- Null Values: Why are null values appearing in the join operation?

When conducting a LEFT JOIN between **PlayerInGameDim** and **ChampionInGameSpecDim** using **PlayerInGameID** as the join key, null values appearing in the output are a common occurrence. This happens when there are records in **PlayerInGameDim** that do not have corresponding entries in **ChampionInGameSpecDim**. Reasons for these nulls include:

- **Data Integrity Issues:**

If foreign key constraints are not rigorously enforced, there may be **PlayerInGameIDs** without valid references in **ChampionInGameSpecDim**. These inconsistencies can arise from delayed data syncing or discrepancies between data sources.

- **Data Entry Errors:**

Errors in data entry or processing can result in unmatched records, where player entries do not correctly link to item data.

- **Business Logic:**

Some players, especially newcomers or those in specific game modes (like tutorials or special events), may not require items, leading to intentionally absent data in **ChampionInGameSpecDim**.

Code Solution:

```
-- Query to fetch player in-game details with item information, handling possible null values
SELECT
    p.PlayerInGameID,
    p.PRID,
    p.PlayerID,
    p.ChampionID,
    p.ClubID,
    p.GameID,
    p.PlayerInGamePositionInGame,
    -- Handle null items by substituting default values
    COALESCE(ci.ChampionItemName, 'No Item') AS ChampionItemName,
-- Displays 'No Item' if null
    COALESCE(ci.ChampionItem1Perk, 'None') AS ChampionItem1Perk,
-- Displays 'None' for missing perks
    COALESCE(ci.ChampionItem2Perk, 'None') AS ChampionItem2Perk,
    COALESCE(ci.ChampionItemPrice, 0) AS ChampionItemPrice
-- Assumes 0 as default price if no item price is provided
FROM
    PlayerInGameDim p
LEFT JOIN
    ChampionInGameSpecDim c ON p.PlayerInGameID = c.PlayerInGameID
LEFT JOIN
    ChampionItemDim ci ON c.ChampionItemID = ci.ChampionItemID;
```

111	SELECT
112	p.PlayerInGameID,
113	p.PRID,
114	p.PlayerID,
115	p.ChampionID,
116	p.ClubID,
117	p.GameID,
118	p.PlayerInGamePositionInGame,
119	-- Handle null items by substituting default values
120	COALESCE(ci.ChampionItemName, 'No Item') AS ChampionItemName,
121	-- Displays 'No Item' if null
122	COALESCE(ci.ChampionItem1Perk, 'None') AS ChampionItem1Perk,
123	-- Displays 'None' for missing perks
124	COALESCE(ci.ChampionItem2Perk, 'None') AS ChampionItem2Perk,
125	COALESCE(ci.ChampionItemPrice, 0) AS ChampionItemPrice
126	-- Assumes 0 as default price if no item price is provided
127	FROM
128	PlayerInGameDim p
129	LEFT JOIN
130	ChampionInGameSpecDim c ON p.PlayerInGameID = c.PlayerInGameID
131	LEFT JOIN
132	ChampionItemDim ci ON c.ChampionItemID = ci.ChampionItemID;
133	

PlayerID	ChampionID	ClubID	GameID	PlayerInGamePositionInGame	ChampionItemName	ChampionItem1Perk	ChampionItem2Perk	ChampionItemPrice
241	55	55	64	Bot Laner	No Item	None	None	0
259	19	8	86	Mid Laner	No Item	None	None	0
172	23	17	75	Support	No Item	None	None	0
226	49	29	81	Top Laner	No Item	None	None	0
202	77	39	19	Support	No Item	None	None	0
10	41	12	72	Mid Laner	No Item	None	None	0
50	5	23	46	Jungler	No Item	None	None	0

6. Database Structure Issues (Q5)

Issue 1: Lack of Foreign Key Constraints

- **Problem:**

The absence of enforced foreign key constraints in tables like **PlayerInGameDim** and **ChampionInGameSpecDim** leads to data integrity issues, including orphan records and mismatches that result in null values and inaccurate data aggregation.

- **Solution:**

Implement and strictly enforce foreign key constraints across all relevant tables. This enforcement should include defining foreign keys in the database schema that link child tables to their respective parent tables via primary keys, ensuring all data points are valid and existent.

Issue 2: Redundant Data Storage

- **Problem:**

Overlapping data storage in dimensions such as **EventDim** and **TicketDim** causes database bloat and complicates data management, impacting performance and update efficiency.

- **Solution:**

Normalize the database schema to reduce redundancy. Conduct a thorough review of all dimensions to identify and consolidate overlapping data, integrating related dimensions such as **EventDim** and **TicketDim** into a single unified dimension. This process should include dependency analysis to ensure data integrity and minimize disruption to existing applications.