

# Data Warehouse Report – Part 1

M32083

UP2067015

---

## Question 1

Write and run five SQL queries. You must submit a screenshot of the queries running and their results as well as the short description of the business rationale, in no more than 200 words per query.

1. To help the business **identify the top selling merchandises online**, we first bring in merchandise ID and the sum of merchandise sold, then we group them by merchandise ID so that we have the total units sold for each merchandise ID. Since we want to view the top merchandises, we order them in descending order.

```
select MerchandiseID, sum(MerchandiseSold) as TotalMerchandiseSold from OnlineSalesFact  
GROUP BY MerchandiseID ORDER BY TotalMerchandiseSold DESC
```



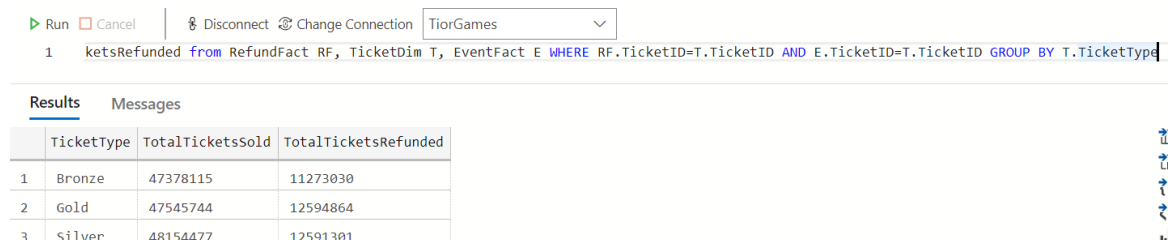
The screenshot shows a SQL query running in a database client. The query is: `select MerchandiseID, sum(MerchandiseSold) as TotalMerchandiseSold from OnlineSalesFact GROUP BY MerchandiseID ORDER BY TotalMerchandiseSold DESC`. The results are displayed in a table with two columns: MerchandiseID and TotalMerchandiseSold. The table shows the top 10 selling merchandise items.

	MerchandiseID	TotalMerchandiseSold
1	27	77515
2	4	65113
3	26	61022
4	5	58532
5	24	58200
6	18	57224
7	2	55640
8	16	55537
9	12	54255
10	14	53837

2. To help the business **compare the ticket refund rate by ticket types**, we first get the ticket type, the sum of tickets sold, and the sum of tickets refunded. Then we group them by ticket types. Now we can calculate the refund rate as below:

$$\text{TotalTicketsRefunded} / \text{TotalTicketsSold}$$

```
select T.TicketType, sum(E.TicketsSold) as TotalTicketsSold, sum(RF.TicketsRefunded) as  
TotalTicketsRefunded from RefundFact RF, TicketDim T, EventFact E WHERE RF.TicketID=T.T  
icketID AND E.TicketID=T.TicketID GROUP BY T.TicketType
```



The screenshot shows a SQL query running in a database client. The query is: `select T.TicketType, sum(E.TicketsSold) as TotalTicketsSold, sum(RF.TicketsRefunded) as TotalTicketsRefunded from RefundFact RF, TicketDim T, EventFact E WHERE RF.TicketID=T.TicketID AND E.TicketID=T.TicketID GROUP BY T.TicketType`. The results are displayed in a table with three columns: TicketType, TotalTicketsSold, and TotalTicketsRefunded. The table shows the refund rate for three ticket types: Bronze, Gold, and Silver.

	TicketType	TotalTicketsSold	TotalTicketsRefunded
1	Bronze	47378115	11273030
2	Gold	47545744	12594864
3	Silver	48154477	12591301

- To help the business **understand the relationship between different promotion types and attendance rate**, we need the promotion type. We also bring in the sum of stadium capacity (assume it equals to the sum of tickets on sale for all events) and sum of spectators number (we use this rather than *TicketsSold* because it excludes the number of tickets refunded). Now we can calculate the refund rate as below:

$$\text{TotalStadiumCapacity} / \text{TotalSpectatorsNumber}$$

```
SELECT P.PromotionType, SUM(S.StadiumCapacity) as TotalStadiumCapacity, SUM(E.Spectator
sNumber) as TotalSpectatorsNumber from ((StadiumDim S inner join GameFact G on S.Stadiu
mID=G.StadiumID) inner join EventFact E on G.EventID=E.EventID) inner join PromotionDim
P on e.PromotionID=p.promotionID GROUP BY P.PromotionType
```

Run Cancel Disconnect Change Connection TiorGames

1 umID) inner join EventFact E on G.EventID=E.EventID) inner join PromotionDim P on e.PromotionID=p.promotionID GROUP BY P.PromotionType

Results Messages

	PromotionType	TotalStadiumCapacity	TotalSpectatorsNumber
1	Direct Marketing	21883000	3823253
2	Sponsorships	16743000	2791398
3	Sales Promotion	18707000	3146927
4	Public Relations	24300500	4141201
5	Digital Promotions	18966000	3064630
6	General Advertising	16492000	2309584

- Let's suppose the business wants to **reward a pay raise to the referees with at least 3 years of experience and appeared in at least 1 game per year**. First we need the referee ID and their years of experience, then we append a column of their total game appearance by counting the occurrence of their ID in the *GameFact* table. At last, we filter years of experience to at least 3. Now we can calculate as below:

$$\text{If } (\text{TotalGameAppearance} / \text{RefereeYearsOfExperience}) \geq 1,$$

we advise the business to reward that corresponding referee

```
select R.RefereeID, R.RefereeYearsOfExperience, COUNT(G.RefereeID) as TotalGameAppearan
ce from RefereeDim R, GameFact G where R.RefereeID=G.RefereeID GROUP BY R. RefereeID, R
.RefereeYearsOfExperience HAVING R.RefereeYearsOfExperience>=3
```

Run

Cancel

Disconnect

Change Connection

TiorGames

1

Dim R, GameFact G where R.RefereeID=G.RefereeID GROUP BY R. RefereeID, R.RefereeYearsOfExperience HAVING R.RefereeYearsOfExperience>=3

Results

Messages

	RefereeID	RefereeYearsOfExperience	TotalGameAppearance
1	3	4	5
2	6	3	5
3	8	5	3
4	9	3	1
5	12	4	3
6	13	3	2
7	14	5	3
8	15	3	3
9	17	3	4
1...	19	4	2
1...	23	5	7
1...	28	5	3

5. Let’s suppose the business is **looking for new opportunities in countries with less professional clubs**. First we bring in the countries, then we count the number of clubs in each country and at the end order the result by ascending order. The top countries have the least number of professional clubs (at least 1).

```
SELECT L.Country, count(C.ClubID) as NumberOfClubs from LocationDim L, ClubDim C where L.LocationID=C.ClubLocation group by L.Country order by NumberOfClubs asc
```

Run

Cancel

Disconnect

Change Connection

TiorGames

1

ClubID) as NumberOfClubs

from LocationDim L, ClubDim C

where L.LocationID=C.ClubLocation

group by L.Country

order by NumberOfClubs

asc

Results

Messages

	Country	NumberOfClubs
1	Albania	1
2	Argentina	1
3	Australia	1
4	Brazil	1
5	Burma	1
6	Cameroon	1
7	Croatia	1
8	Czechia	1
9	Ecuador	1
10	Hungary	1
11	India	1
12	Indonesia	1
13	Latvia	1
14	Libya	1
15	Mongolia	1
16	Mozambiq..	1
17	Poland	1
18	Puerto R...	1
19	Serbia	1
20	South Af	1

Results grid

## Question 2

Modify the given schema and suggest at least two more dimensions that would provide you with insights that you wish were there. You must submit the two dimensions, the data dictionary for them and the rationale report.

First, I would like to help the business identify **purchase orders (PO) for their merchandises**. This will be a table that provides characteristics of what a PO requires, such as shipping cost, freight rate, arrival date, etc. It contains 2 foreign keys of *MerchandiseID* and *ProviderID*.

PurchaseOrderDim
POID (PK)
MerchandiseID (FK)
ProviderID (FK)
PODate
ArrivalDate
CountryOfOrigin
ProductSize
ShippingMethod
TariffRate
MerchandiseCost
FreightRate

Dimension	Attributes	Data type	Identifier	notes
PurchaseOrderDim	POID	INT	PK	
	MerchandiseID	INT	FK	
	ProviderID	INT	FK	
	PODate	DATE		
	ArrivalDate	DATE		
	CountryOfOrigin	VARCHAR(75)		
	ProductSize	VARCHAR(75)		
	ShippingMethod	VARCHAR(75)		
	TariffRate	INT		
	MerchandiseCost	INT		
	FreightRate	INT		

Second, I would like to help the business identify **fouls that given during games**. This will be a table that provides characteristics of foul, including foul reasons, foul types, foul penalty, etc. It contains 2 foreign keys *PlayerID* of *RefereeID* and at the same time is a foreign key to the *GameFact* table.

FoulDim
FoulID (PK)
PlayerID (FK)
RefereeID (FK)
FoulReason
FoulType
FoulPenalty

Dimension	Attributes	Data type	Identifier	notes
FoulDim	FoulID PlayerID RefereeID FoulReason FoulType FoulPenalty	INT INT INT VARCHAR(75) VARCHAR(75) VARCHAR(75)	PK FK FK	

### Question 3

A common way of introducing data warehousing is to refer to its fundamental characteristics. Identify and describe three characteristics of data warehousing. Use no more than 300 words in total.

Data warehouse is **integrated**, meaning it connects data from multiple data sources, including internal databases and RDMS, in a common and universally acceptable manner. This ensures the consistency of codes, attribute measures, naming conventions, and formats (Naeem, 2020).

Data warehouse is **time-variant**, meaning the data it holds is retrieved periodically within a specific time period and does not change over time. Therefore, the information is provided from a historical standpoint (Naeem, 2020).

Data warehouse is also **non-volatile**, meaning entry of new data does not remove or overwrite the previous data. The only moves that we can do to the data are reading and loading (Naeem, 2020).

## Question 4

A junior member of your team complained fiercely to the data warehouse administrators that the access that you have is too restricted, and they would like to be able to view the spectators of each game of each event. In other words, the people that purchase tickets and attend the event. They argued that they would like to know when these people purchased the tickets, how many they bought and why they returned them if they did so.

Your response is that a dimension such as this would require a new fact table. The junior member seemed confused and asked you why, since there are already the eventFact, Tickets, TimeDim, DateDim and Refund tables.

To answer this you need to simply state one sentence. What would you say?

Use no more than 100 words.

We do not have the right to access those kinds of data.



## Question 5

When joining two tables in any type of DBMS system, including a data warehouse, you have multiple join options. Typically, when two tables are joined together with inner, left or right join they may or may not have the same number of rows.

A junior member of your team came to you saying that we have incomplete data and that our audience will be very upset if we present them with null data values. You are very confused and ask the member of your team to show you the query they are running:

```
select * from PlayerInGameDim left join ChampionInGameSpecDim on
PlayerInGameDim.PlayerInGameID = ChampionInGameSpecDim.PlayerInGameID
```

After a bit of investigation, you identify that there are indeed null values and some of the values do not adhere to foreign key constraints. Your task is:

- Explain why there are null values
  - Explain why some of the values do not match foreign key constraints
  - Suggest a way to solve it. You have the choice to present a code solution or a narrative. Remember that you will not be able to execute code that alters the structure of the database.
- 
- When we use the left join, it returns all records from the left table (*PlayerInGameDim*), including the records that do not match the right table (*ChampionInGameSpecDim*). As a result, null values are found in each column of the right table in the records that have no matches between two tables.
  - Some of the values do not match foreign key constraints may possibly because *PlayerInGameID* in the *PlayerInGameDim* table has a NOT NULL constraint while *PlayerInGameID* in the *ChampionInGameSpecDim* table does not.
  - I will use the inner join to bring only the records that both tables match to avoid NULL values.

## Reference

Naeem, T. (2020, February 3). Data Warehouse Concepts: Kimball vs. Inmon Approach | Astera. Astera. <https://www.astera.com/type/blog/data-warehouse-concepts/>