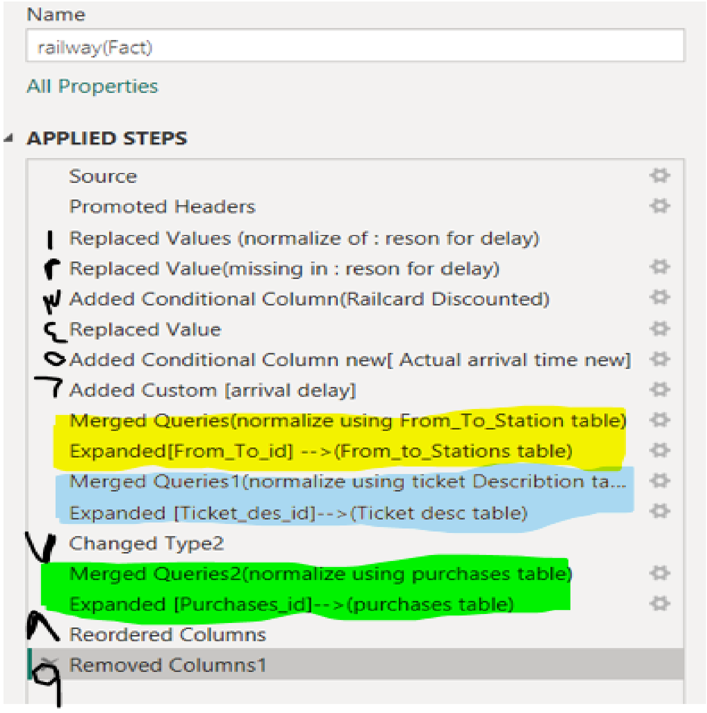
Applied Steps on Railway Data

Project number	6
Project name	(Outstanding): UK Train Rides



Name

railway(Fact)

All Properties

APPLIED STEPS

Source
Promoted Headers
١ Replaced Values (normalize of : reson for delay)
٢ Replaced Value(missing in : reson for delay)
٣ Added Conditional Column(Railcard Discounted)
٤ Replaced Value
٥ Added Conditional Column new[Actual arrival time new]
٦ Added Custom [arrival delay]
Merged Queries(normalize using From_To_Station table)
Expanded[From_To_id] -->(From_to_Stations table)
Merged Queries1(normalize using ticket Describtion ta...
Expanded [Ticket_des_id]-->(Ticket desc table)
٧ Changed Type2
Merged Queries2(normalize using purchases table)
Expanded [Purchases_id]-->(purchases table)
٨ Reordered Columns
٩ Removed Columns1

- first , we cleaning a denormalized table called (Railway Fact)

1-In the [Reason for delay] column we doing this replacement :

- Weather Condition ➔ Weather
- Signal failure ➔ Signal Failure
- Staff Shortage ➔ Staffing

2-Missing value ➔No Delaying

3- Added Conditional Column(Railcard Discounted) Adding a conditional column based on [Railcard] column:

- If [Railcard] column = Adult OR Disabled OR Senior➔True

  Else ➔ Fales

4- in [actual arrival time]column , we replaced " " ➔ null

5- Added Conditional Column new[Actual arrival Time new]

- Add new [actual arrival time new] column to processed the null values exist
- the null values is assigned because the journey status is Canceled
- Null value is replaced with [arrival time]

6- Added Custom(Arrival_Delay)

- To calculate Delay time , the difference between [arrival time] column &[Actual arrival time]column

7- Check for every data type in this table

8- Reorders columns to be readable

Now the (Railway Fact table) is Cleaned , and Ready to apply normalization (Dimensional tables)

we have a denormalized table called (Railway fact) , this table contain a lot of information (columns) about many Entity in the business combined into one place (Railway Fact) table

- information in this table about

➢ Frist merge

1- Stations (From…….To……..) source station ➔ Destination station (are exchange with each ticket has been reserved)

- But when we look there , we find only 65 unique values related to stations [(From …… To…….) source station➔Destination station]

Based on it

- we isolated information about stations (Departure Station & Arrival Destination) From (Railway Fact) table
- and create a new table called (From_to_Stations) ➔select two columns➔remove duplicates
- after that we assign a index from 1 ➔ work as unique identifier for this table (PK)

Relation between Fact & Dimensional(From_To_Stations)

- in the (Railway Fact table) we apply a merge query based on two column (Departure Station & Arrival Destination)
- then, we expanded only the (PK) related to (From_To_Stations Table)
- and remove the (Departure Station & Arrival Destination) from (Railway Fact Table)

➢ Second merge

2- information about Ticket Description (Ticket Class & Ticket Type) a characteristics of tickets are exchange with each other for each reserved ticket

- But when we look there , we find only 6 unique values related to Ticket Description (Ticket Class & Ticket Type) exchange with each other

Based on it

- we isolated information about Ticket Description (Ticket Class & Ticket Type)From (Railway Fact) table
- and create a new table called (Ticket Description) ➔select two columns➔remove duplicates
- after that we assign a index from 1 ➔ work as unique identifier for this table (PK) called (Ticket_des_id)

Create the relationship as it was created above between Fact &(Ticket Description).

➢ Third merge

3- information about Purchases (Purchase Type & Payment Method) a characteristics of Purchases are exchange with each other for each reserved ticket

- But when we look there , we find only 6 unique values related to Purchases (Purchase Type & Payment Method) exchange with each other

Based on it

- we isolated information about Purchase (Purchase Type & Payment Method) From (Railway Fact) table
- and create a new table called (Purchase) ➔select two columns➔remove duplicates
- after that we assign an index from 1 ➔ work as unique identifier for this table (PK) called (Purchases_id)

Create the relationship as it was created above between Fact &(Purchase).

- now we closed and apply all this steps

Then, we create a calendar using DAX In Power bi Desktop and make a Date of this calendar   based on the

[Date of Purchase column] in (Railway Fact Table) and customize this calendar by adding [Month] &[Day]&[Year]

Team member
Amr Osama Elsayed Ahmed
Abdelhamed Ebrahim Abdelhamed
Mousa Mohamed Elsayed Gaballah
Ziad Abdelaziz Shoaib
Ahmed Hamada