TERM DEPOSIT DW Documentation

Table of Contents

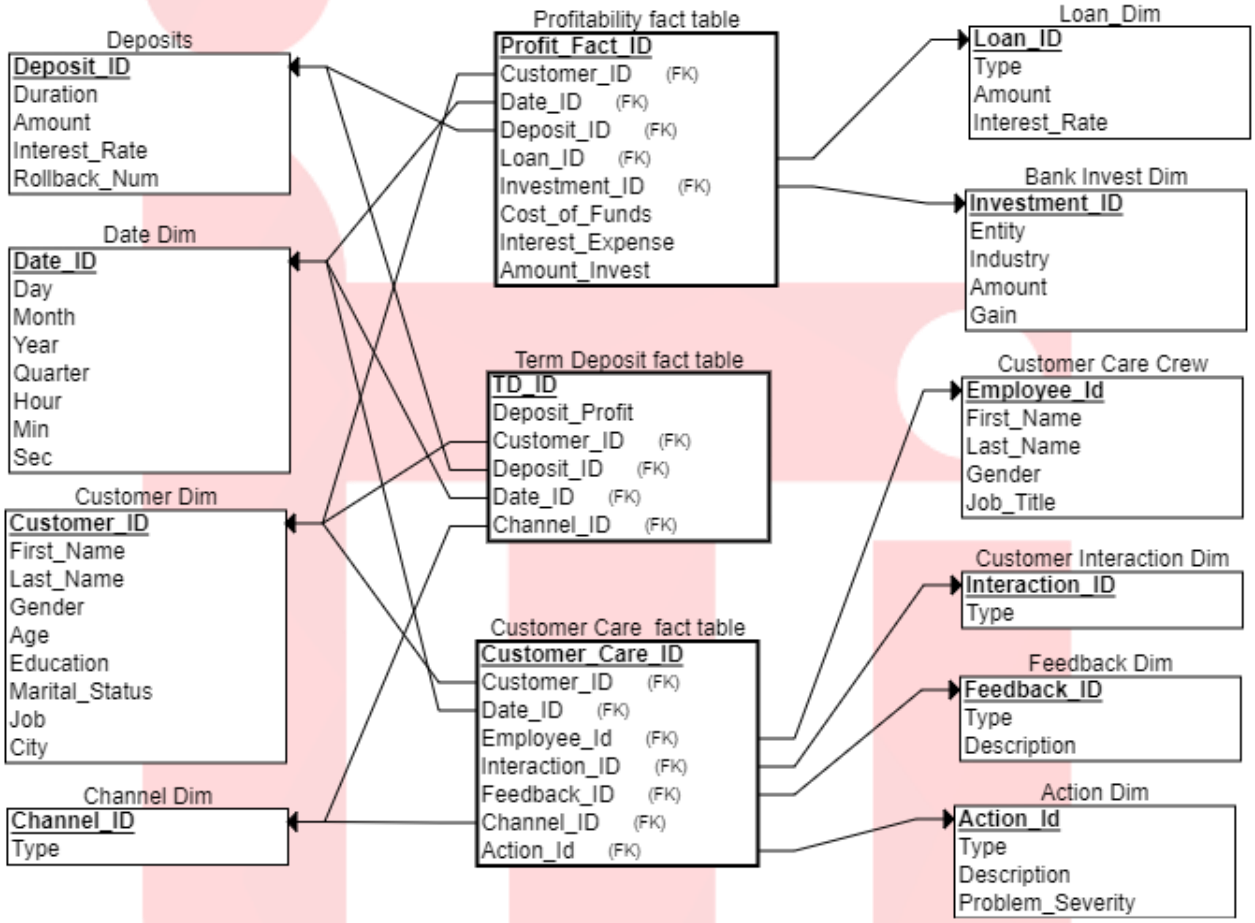
Bus Matrix

	Term Deposit	Profitability	Customer Care
Action			√
Bank Invest		√	
Channel	√	√	√
Crew			√
Customer	√	√	√
Interaction			√
Date	√	√	√
Deposits	√	√	
Feedback			√
Loan		√	

Business Statement

The bank wants to analyze and improve its term deposit system to ensure ongoing business processes are successful and expand its operations by discovering new opportunities. The bank is interested in understanding the behavior of customers who invest in term deposits, the **profitability** of the system, and customer interactions. The marketing department wants to understand the **demographic profile of customers**, their investment amount, the duration of their investment, the interest rate they are offered, and how often they roll over their term deposits. The finance team wants to understand the cost of funds, the interest expense, and the net interest margin, as well as the impact of changes in interest rates on the system. The customer interaction process includes opening a term deposit account, depositing funds, and withdrawing funds, and the bank's customer care team interacts with customers to **handle inquiries**, complaints, and feedback. The bank wants to analyze the interaction type, problem severity, and channel of interaction for each customer interaction.

Logical Model



Why we use Star Schema?

We find that the star schema is suitable for our case study and performs better in implementation, after searching also we find that the star schema is better for many reasons:

- Streamlined query execution: Star schema modeling enables us to streamline and improve query execution. Queries may be run effectively and rapidly without the use of intricate joins or subqueries since the fact table serves as the foundation of the schema and is linked to the dimension tables by foreign keys.

- Star schema modeling is scalable and has a great data handling capacity. The schema can handle massive volumes of data without affecting query performance or data analysis by dividing the data into smaller, more manageable tables.

- Better data analysis: Star schema modeling offers a clear and simple method for analyzing data from several aspects. Analysts can quickly drill down into the data by traversing through the dimension tables, enabling more intricate analysis and insights.

What does the data represent?

- **Customer_Dim:** Contains the demographic information of the bank's customers, such as their first name, last name, gender, age, education, marital status, job, and cities located in Egypt.

- **Date_Dim:** This table contains information about dates, including the day, month, year, quarter, hour, minute, and second.

- **Deposits_Dim:** Contains information about the term deposits, including the deposit ID, duration “in moths”, amount, interest rate, and rollback number.

- **Loan_Dim:** Contains information about loans, including the loan ID, type, amount, and interest rate.

- **Bank_Invest_Dim:** This table contains information about the bank's investments, including the investment ID, entity, industry, amount, and gain.

- **Customer_Interaction_Dim:** This table contains information about customer interactions, including the interaction ID and type ,” Open New Account,Deposit Cash and Withdraw Money”

- **Feedback_Dim:** This table contains information about customer feedback, including the feedback ID, type, and description.

- **Channel_Dim:** Contains information about the channels of interaction, including the channel ID and type “Telephone, Internet and In person”

- **Action_Dim:** This table contains information about customer care actions, including the action ID, type, description, and problem severity.

- **Customer_Care_Crew_Dim:** This table contains information about the bank's customer care employees, including their employee ID, first name, last name, gender, and job title.

Queries

- **How many term deposits does each customer have?**

```
SELECT Customer_ID, First_Name, Last_Name, Gender, Age, Education,
Marital_Status, Job, City
FROM Customer_Dim
WHERE Customer_ID IN (SELECT Customer_ID FROM Term_Deposit_Fact);
```

- **How many customers with term deposits does the bank have, grouped by gender?**

```
SELECT Gender, COUNT(*) AS Total_Customers
FROM Customer_Dim
WHERE Customer_ID IN (SELECT Customer_ID FROM Term_Deposit_Fact)
GROUP BY Gender;
```

- **How many customers with term deposits does the bank have, grouped by education level and job?**

```
SELECT Education, Job, COUNT(*) AS Total_Customers
FROM Customer_Dim
WHERE Customer_ID IN (SELECT Customer_ID FROM Term_Deposit_Fact)
GROUP BY Education, Job;
```

- **How many customers with term deposits does the bank have, grouped by marital status?**

```
SELECT Marital_Status, COUNT(*) AS Total_Customers
FROM Customer_Dim
WHERE Customer_ID IN (SELECT Customer_ID FROM Term_Deposit_Fact)
GROUP BY Marital_Status;
```

- **How many customers with term deposits does the bank have, grouped by age?**

```
SELECT Age, COUNT(*) AS Total_Customers
FROM Customer_Dim
WHERE Customer_ID IN (SELECT Customer_ID FROM Term_Deposit_Fact)
GROUP BY Age;
```

- **How many customers with term deposits does the bank have who are graduates?**

```
SELECT COUNT(*) AS Total_Customers
FROM Customer_Dim
WHERE Customer_ID IN (SELECT Customer_ID FROM Term_Deposit_Fact)
AND Education = 'graduate';
```

- **How much profit did the bank make from term deposits in each year, and which year had the highest profit?**

```
SELECT d.Year, SUM(t.Deposit_Profit) AS Total_Deposit_Profit
FROM Date_Dim d
JOIN Term_Deposit_Fact t ON d.Date_ID = t.Date_ID
GROUP BY d.Year
ORDER BY Total_Deposit_Profit DESC;
```

- **How much profit did the bank make from term deposits for customers in each age group, and which age group contributed the most profit?**

```
SELECT d.Year, SUM(t.Deposit_Profit) AS Total_Deposit_Profit
FROM Date_Dim d
JOIN Term_Deposit_Fact t ON d.Date_ID = t.Date_ID
GROUP BY d.Year
ORDER BY Total_Deposit_Profit DESC;
```

- **Which customers contributed the most profit to the bank from their term deposits, and how much profit did they contribute?**

```
SELECT Customer_ID, SUM(Deposit_Profit) AS Total_Deposit_Profit
FROM Term_Deposit_Fact
GROUP BY Customer_ID
ORDER BY Total_Deposit_Profit DESC;
```

- **Which jobs contributed the most profit to the bank from term deposits, and how much profit did they contribute?**

```
SELECT c.Job, SUM(t.Deposit_Profit) AS Total_Deposit_Profit
FROM Customer_Dim c
JOIN Term_Deposit_Fact t ON c.Customer_ID = t.Customer_ID
GROUP BY c.Job
ORDER BY Total_Deposit_Profit DESC;
```

- **What is the total amount and average gain of each bank entity's investments?**

```
SELECT Entity, SUM(Amount) AS Total_Amount, AVG(Gain) AS Avg_Gain
FROM Bank_Invest_Dim
GROUP BY Entity
ORDER BY Avg_Gain DESC;
```

- **What is the highest loan amount?**

```
SELECT MAX(Amount) AS Highest_Loan FROM Loan_Dim;
```

Indexes

For Indexes, We used only:

Cluster index on each PK columns " By default on Toad".

Oracle defines two types of indexes: the B-Tree (Balanced Tree) Index and the Bitmap Index.

B-Tree	Bitmap
• Index-organized tables An index-organized table differs from a heap-organized because the data is itself the index.	• Function-based indexes This type of index includes columns that are either transformed by a function, such as the `UPPER` function, or included in an expression. B-tree or bitmap indexes can be function-based.
• Reverse key indexes In this type of index, the bytes of the index key are reversed, for example, 103 is stored as 301. The reversal of bytes spreads out inserts into the index over many blocks.	• Application domain indexes A user creates this type of index for data in an application-specific domain. The physical index need not use a traditional index structure and can be stored either in the Oracle database as tables or externally as a file.
• UNIQUE Descending indexes This type of index stores data on a particular column or columns in descending order.	
• B-tree cluster indexes	