Dimensional Modeling

Odato,CJ

2025-03-08

Setting Directory

```
setwd("C:/Users/Admin/Desktop/R/DW101_Dimensional_Modeling")
```

Loading Packages

```
library(dplyr)
library(DBI)
library(RSQLite)
library(lubridate)
library(readxl)
```

Importing Dataset

```
northwind_data <- read_excel("northwind.xlsx", sheet = "orders")
```

Creating a Fact Table

```
northwind_data$order_date <- as.Date(northwind_data$order_date)
northwind_data$date_day <- day(northwind_data$order_date)
northwind_data$date_month <- month(northwind_data$order_date,label=TRUE)
northwind_data$date_year <- year(northwind_data$order_date)
northwind_data$date_quarter <- quarter(northwind_data$order_date)
northwind_data$time_id <- seq(1,length(northwind_data$order_date),by=1)
part_1 <- cbind(northwind_data[1:3],northwind_data[19])
northwind_details <- read_excel("northwind.xlsx",sheet = "order_details")
part_2 <- cbind(northwind_details[1:5])
northwind_supplier <- read_excel("northwind.xlsx",sheet = "products")
part_3 <- cbind(northwind_supplier[1],northwind_supplier[3])
```

Combining 3 parts

```
order_fact <- part_1 %>%
  left_join(part_2, by = "order_id") %>%
  left_join(part_3, by = "product_id")
```

Computing Total Sales

```
order_fact$total_sales <- (order_fact$unit_price * (1 - order_fact$discount)) *
  order_fact$quantity
```

Arranging orders

```
order_fact <- order_fact[,c("order_id","customer_id","product_id","employee_id",
                            "time_id","unit_price","quantity","discount","total_sales")]
```

Dimensions

```
dim_time <- cbind(northwind_data[19],northwind_data[15:18])
northwind_employee <- read_excel("northwind.xlsx", sheet = "employees")
dim_employee <- cbind(northwind_employee[1:2],northwind_employee[4],northwind_employee[17])
northwind_customers <- read_excel("northwind.xlsx", sheet = "customers")
dim_customers <- cbind(northwind_customers[1:2],northwind_customers[6],northwind_customers[9])
northwind_p <- read_excel("northwind.xlsx", sheet = "products")
dim_product <- cbind(northwind_p[1:4])
```

Join Fact Table with Dimensions

```
star_schema <- order_fact %>%
  left_join(dim_customers, by = "customer_id") %>%
  left_join(dim_product, by = "product_id") %>%
  left_join(dim_time, by = "time_id") %>%
  left_join(dim_employee, by = "employee_id")
```

Store in Database

```
conn <- dbConnect(SQLite(), "northwind_1.db")

dbWriteTable(conn, "order_fact", order_fact, overwrite = TRUE)
dbWriteTable(conn, "dim_customers", dim_customers, overwrite = TRUE)
dbWriteTable(conn, "dim_product", dim_product, overwrite = TRUE)
dbWriteTable(conn, "dim_time", dim_time, overwrite = TRUE)
dbWriteTable(conn, "dim_employee", dim_employee, overwrite = TRUE)
dbListTables(conn)
```

```
## [1] "dim_customers" "dim_employee"  "dim_product"   "dim_time"
## [5] "order_fact"
```

```
dbDisconnect(conn)
```

Open in R

```r
con <- dbConnect(RSQLite::SQLite(), "northwind_1.db")
```

ANSWERS

1. Granularity Statement

Each record in the Sales Fact table represents a single line item of a sales transaction, capturing the sale of a specific product by a specific customer, handled by a specific employee, at a specific time. The transaction includes the unit price of the product, the quantity purchased, any discount applied, and the computed total sales amount.

2. What were Northwind's top-selling products of all time?

```r
top_selling <- "SELECT p.product_name, SUM(o.quantity) AS total_quantity_sold
FROM order_fact o
JOIN dim_product p ON o.product_id = p.product_id
GROUP BY p.product_name
ORDER BY total_quantity_sold DESC
LIMIT 10"

df1 <- dbGetQuery(con, top_selling)
print("Top Selling Products ")
```

```
## [1] "Top Selling Products "
```

```r
print(df1)
```

```
##              product_name total_quantity_sold
## 1       Camembert Pierrot                1577
## 2    Raclette Courdavault                1496
## 3       Gorgonzola Telino                1397
## 4  Gnocchi di nonna Alice                1263
## 5                 Pavlova                1158
## 6    Rhönbräu Klosterbier                1155
## 7      Guaraná Fantástica                1125
## 8        Boston Crab Meat                1103
## 9          Tarte au sucre                1083
## 10            Flotemysost                1057
```

3. What was Northwind's top-selling product(s) per month? per quarter?

Per Month

```r
# Per Month
query_top_products_month <- "
SELECT p.product_name, t.date_month, SUM(o.quantity) AS total_quantity_sold
FROM order_fact o
JOIN dim_product p ON o.product_id = p.product_id
JOIN dim_time t ON o.time_id = t.time_id
GROUP BY t.date_month, p.product_name
```

```
HAVING SUM(o.quantity) = (
    SELECT MAX(total_sales)
    FROM (
        SELECT t2.date_month, SUM(o2.quantity) AS total_sales
        FROM order_fact o2
        JOIN dim_time t2 ON o2.time_id = t2.time_id
        GROUP BY t2.date_month, o2.product_id
    ) sub
    WHERE sub.date_month = t.date_month
)
ORDER BY t.date_month;
"
top_products_month <- dbGetQuery(con, query_top_products_month)
print("Top Products per Month ")
```

```
## [1] "Top Products per Month "
```

```
print(top_products_month)
```

```
##            product_name date_month total_quantity_sold
## 1  Raclette Courdavault        Apr                 326
## 2      Boston Crab Meat        Aug                 160
## 3     Gorgonzola Telino        Dec                 313
## 4          Pâté chinois        Feb                 220
## 5  Raclette Courdavault        Jan                 262
## 6  Raclette Courdavault        Jul                 185
## 7     Gorgonzola Telino        Jun                 171
## 8    Guaraná Fantástica        Mar                 333
## 9     Gorgonzola Telino        May                 155
## 10    Camembert Pierrot        Nov                 208
## 11          Flotemysost        Oct                 188
## 12     Boston Crab Meat        Sep                 194
```

Per Quarter

```
# Per Quarter
query_top_products_quarter <- "
SELECT p.product_name, t.date_quarter, SUM(o.quantity) AS total_quantity_sold
FROM order_fact o
JOIN dim_product p ON o.product_id = p.product_id
JOIN dim_time t ON o.time_id = t.time_id
GROUP BY t.date_quarter, p.product_name
HAVING SUM(o.quantity) = (
    SELECT MAX(total_sales)
    FROM (
        SELECT t2.date_quarter, SUM(o2.quantity) AS total_sales
        FROM order_fact o2
        JOIN dim_time t2 ON o2.time_id = t2.time_id
        GROUP BY t2.date_quarter, o2.product_id
    ) sub
    WHERE sub.date_quarter = t.date_quarter
)
ORDER BY t.date_quarter;
"
top_products_quarter <- dbGetQuery(con, query_top_products_quarter) %>% as_tibble()
print("Top Products per Quarter ")
```

```
## [1] "Top Products per Quarter "
```

```
print(top_products_quarter)
```

```
## # A tibble: 4 x 3
##   product_name         date_quarter total_quantity_sold
##   <chr>                       <int>               <dbl>
## 1 Guaraná Fantástica              1                 539
## 2 Raclette Courdavault            2                 395
## 3 Boston Crab Meat                3                 446
## 4 Gorgonzola Telino               4                 599
```

4. Who are the best customers in terms of sales of all time?

```
query_best_customers <- "
SELECT c.company_name, SUM(o.unit_price * o.quantity) AS total_sales
FROM order_fact o
JOIN dim_customers c ON o.customer_id = c.customer_id
GROUP BY c.company_name
ORDER BY total_sales DESC
LIMIT 10;
"
best_customers <- dbGetQuery(con, query_best_customers)
print("The Best Customers ")
```

```
## [1] "The Best Customers "
```

```
print(best_customers)
```

```
##                    company_name total_sales
## 1                    QUICK-Stop   117483.39
## 2            Save-a-lot Markets   115673.39
## 3                  Ernst Handel   113236.68
## 4  Hungry Owl All-Night Grocers    57317.39
## 5    Rattlesnake Canyon Grocery    52245.90
## 6                 Hanari Carnes    34101.15
## 7                Folk och fä HB    32555.55
## 8                Mère Paillarde    32203.90
## 9               Königlich Essen    31745.75
## 10                Queen Cozinha    30226.10
```

5. Who are the best customers in terms of sales per month? per quarter?

Per Month

```
# Per Month
query_best_customers_month <- "
SELECT c.company_name, t.date_month, SUM(o.unit_price * o.quantity) AS total_sales
FROM order_fact o
JOIN dim_customers c ON o.customer_id = c.customer_id
JOIN dim_time t ON o.time_id = t.time_id
GROUP BY c.company_name, t.date_month
HAVING SUM(o.unit_price * o.quantity) = (
```

```
    SELECT MAX(total_sales)
    FROM (
        SELECT t2.date_month, c2.company_name, SUM(o2.unit_price * o2.quantity) AS total_sales
        FROM order_fact o2
        JOIN dim_customers c2 ON o2.customer_id = c2.customer_id
        JOIN dim_time t2 ON o2.time_id = t2.time_id
        GROUP BY t2.date_month, c2.company_name
    ) sub
    WHERE sub.date_month = t.date_month
)
ORDER BY t.date_month;
"
best_customers_month <- dbGetQuery(con, query_best_customers_month) %>% as_tibble()
print("Best Customers per Month ")
```

```
## [1] "Best Customers per Month "
```

```
print(best_customers_month)
```

```
## # A tibble: 12 x 3
##    company_name                date_month total_sales
##    <chr>                       <chr>            <dbl>
##  1 Save-a-lot Markets          Apr             25353.
##  2 QUICK-Stop                  Aug              7338.
##  3 Ernst Handel                Dec             19759.
##  4 QUICK-Stop                  Feb             23147.
##  5 Ernst Handel                Jan             17711.
##  6 Save-a-lot Markets          Jul             14358.
##  7 Save-a-lot Markets          Jun              3680.
##  8 Hanari Carnes               Mar             17112.
##  9 QUICK-Stop                  May             16043.
## 10 Piccolo und mehr            Nov             12654.
## 11 Save-a-lot Markets          Oct             19994.
## 12 Hungry Owl All-Night Grocers Sep            11794.
```

Per Quarter

```
# Per Quarter
query_best_customers_quarter <- "
SELECT c.company_name, t.date_quarter, SUM(o.unit_price * o.quantity) AS total_sales
FROM order_fact o
JOIN dim_customers c ON o.customer_id = c.customer_id
JOIN dim_time t ON o.time_id = t.time_id
GROUP BY c.company_name, t.date_quarter
HAVING SUM(o.unit_price * o.quantity) = (
    SELECT MAX(total_sales)
    FROM (
        SELECT t2.date_quarter, c2.company_name, SUM(o2.unit_price * o2.quantity) AS total_sales
        FROM order_fact o2
        JOIN dim_customers c2 ON o2.customer_id = c2.customer_id
        JOIN dim_time t2 ON o2.time_id = t2.time_id
        GROUP BY t2.date_quarter, c2.company_name
    ) sub
    WHERE sub.date_quarter = t.date_quarter
)
```

```
ORDER BY t.date_quarter;
"
best_customers_quarter <- dbGetQuery(con, query_best_customers_quarter) %>% as_tibble()
print("Best Customers per Quarter ")
```

```
## [1] "Best Customers per Quarter "
```

```
print(best_customers_quarter)
```

```
## # A tibble: 4 x 3
##   company_name       date_quarter total_sales
##   <chr>                     <int>       <dbl>
## 1 QUICK-Stop                    1      36248.
## 2 QUICK-Stop                    2      37003.
## 3 Save-a-lot Markets            3      25251
## 4 Save-a-lot Markets            4      31311.
```

6. How much did Northwind sell by each product category per month?

```
query_sales_category_month <- "
SELECT p.category_id, t.date_month, SUM(o.unit_price * o.quantity) AS total_sales
FROM order_fact o
JOIN dim_product p ON o.product_id = p.product_id
JOIN dim_time t ON o.time_id = t.time_id
GROUP BY p.category_id, t.date_month
ORDER BY t.date_month, p.category_id;
"
sales_category_month <- dbGetQuery(con, query_sales_category_month) %>% as_tibble()
print("Sales per Month")
```

```
## [1] "Sales per Month"
```

```
print(sales_category_month)
```

```
## # A tibble: 96 x 3
##    category_id date_month total_sales
##          <dbl> <chr>            <dbl>
##  1           1 Apr             31848.
##  2           2 Apr             16640.
##  3           3 Apr             20898.
##  4           4 Apr             43099.
##  5           5 Apr             12345.
##  6           6 Apr             29459.
##  7           7 Apr             21210.
##  8           8 Apr             14831.
##  9           1 Aug             11190.
## 10           2 Aug              6836.
## # i 86 more rows
```

7. How much did Northwind sell by each product category per quarter?

```
query_sales_category_quarter <- "
SELECT p.category_id, t.date_quarter, SUM(o.unit_price * o.quantity) AS total_sales
FROM order_fact o
JOIN dim_product p ON o.product_id = p.product_id
JOIN dim_time t ON o.time_id = t.time_id
GROUP BY p.category_id, t.date_quarter
ORDER BY t.date_quarter, p.category_id;
"
sales_category_quarter <- dbGetQuery(con, query_sales_category_quarter) %>% as_tibble()
print(sales_category_quarter)
```

```
## # A tibble: 32 x 3
##    category_id date_quarter total_sales
##          <dbl>        <int>       <dbl>
##  1           1            1     131499.
##  2           2            1      36969.
##  3           3            1      67037.
##  4           4            1      71048.
##  5           5            1      33176.
##  6           6            1      53037.
##  7           7            1      25280.
##  8           8            1      45076.
##  9           1            2      55700.
## 10           2            2      25330.
## # i 22 more rows
```

8. Which employee (with her/his supervisor) sold the most orders?

```
query_top_employee <- "
SELECT e.last_name, e.reports_to, COUNT(o.order_id) AS total_orders
FROM order_fact o
JOIN dim_employee e ON o.employee_id = e.employee_id
GROUP BY e.last_name, e.reports_to
ORDER BY total_orders DESC
LIMIT 1;
"
top_employee <- dbGetQuery(con, query_top_employee)
print(top_employee)
```

```
##   last_name reports_to total_orders
## 1   Peacock     Fuller          420
```

Close Database

```
dbDisconnect(con)
```