

Analyze and model data warehouse for airline system

Analyzing and Data Warehouse Modeling Project



**MADE BY:**

Dina Hosny

Fatma Nabil

kareem beltagy

mohamed labib

[Introduction: 2](#_Toc123256875)

[Dimensional Modeling Process: 2](#_Toc123256876)

[1- Business process: 2](#_Toc123256877)

[2- Granularity: 2](#_Toc123256878)

[3- Dimensions Tables: 2](#_Toc123256879)

[4- Facts Tables: 2](#_Toc123256880)

[Project Processes: 3](#_Toc123256881)

[Schema Type: 3](#_Toc123256882)

[Process Diagram: 3](#_Toc123256883)

[Logical Model: 3](#_Toc123256884)

[Tables Identifications: 5](#_Toc123256885)

[1- Passenger Dimension. 5](#_Toc123256886)

[2- Flight Dimension. 5](#_Toc123256887)

[3- Date Dimension. 5](#_Toc123256888)

[4- Class Dimension. 5](#_Toc123256889)

[5- Reservation Channel Dimension. 5](#_Toc123256890)

[6- Fare Base Dimension. 5](#_Toc123256891)

[7- Hotel Dimension. 5](#_Toc123256892)

[8- Customer Services Dimension. 6](#_Toc123256893)

[9- Interaction Dimension. 6](#_Toc123256894)

[10- Reservation Fact. 6](#_Toc123256895)

[11- Transit Hotel Fact. 6](#_Toc123256896)

[12- Customer Care Fact. 6](#_Toc123256897)

[Assumptions: 7](#_Toc123256898)

[Physical Model: 7](#_Toc123256899)

[Insert Dummy Data into The Physical Tables: 13](#_Toc123256900)

[Insert The Measured Data Into Fact Tables 19](#_Toc123256901)

[SQL Queries to Answer Business Questions: 23](#_Toc123256902)

Introduction:

The project’s purpose is to analyze the flight activities of some airline companies and their frequent flyers and model its data warehouse schema.

Dimensional Modeling Process:

1. Business process:

Analyze the Airlines flights activities.

1. Granularity:

Per Reservation.  
  
**NOTE:** Customer Care contains transactions that can be committed before any reservation processes like making an interaction before the flight reservation. In this case and for this business process, the granularity will be Per Each Interaction.

1. Dimensions Tables:
2. Passenger Dimension.
3. Flight Dimension.
4. Date Dimension.
5. Class Dimension.
6. Reservation Channel Dimension.
7. Fare Base Dimension.
8. Hotel Dimension.
9. Interaction Dimension.
10. Customer Services Dimension.
11. Facts Tables:
12. Reservation Fact.
13. Transit Hotel Fact.
14. Customer Care Fact.

Project Processes:

Schema Type:

**Galaxy Schema:** The business process required more than one fact table to represent different processes.

Process Diagram:

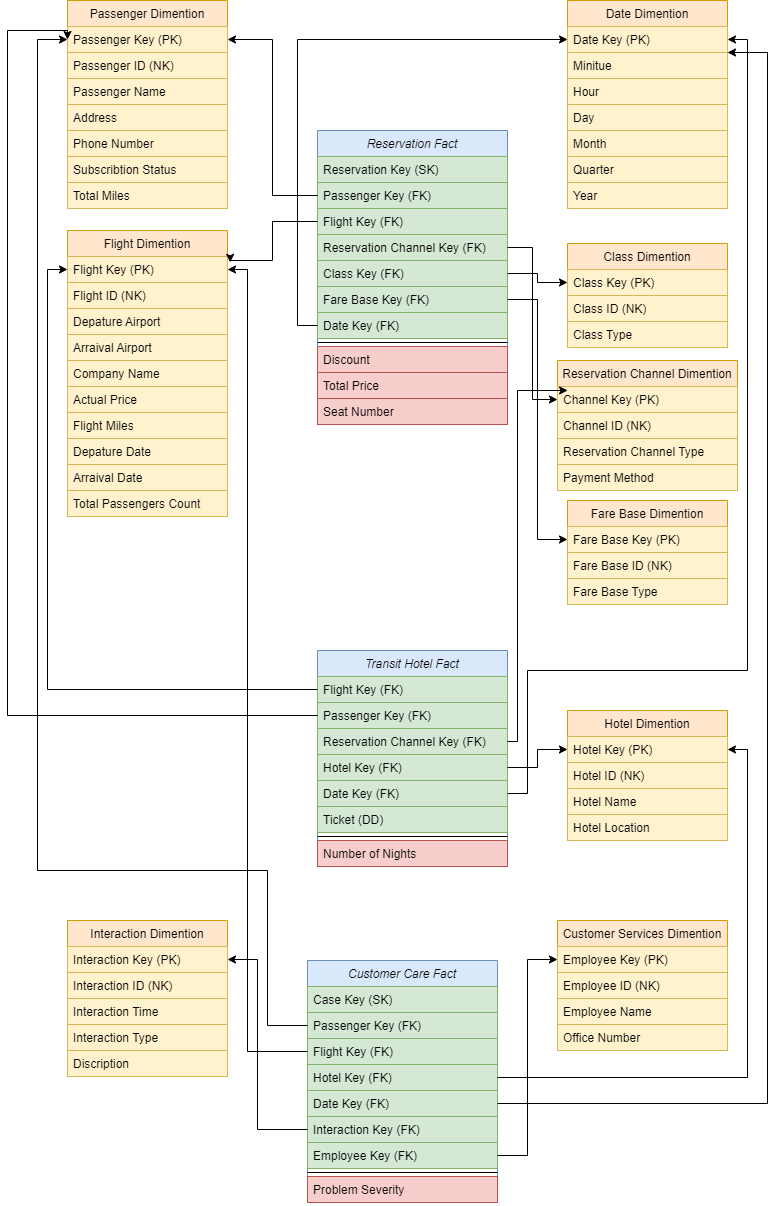
Using **Kimball’s** Philosophy.



There will be one row in each fact table for each different transaction collected from passengers. The dimensionality associated with this data is quite extensive. The **Role-Playing** technique is used as the date dimension table was used more than one time in the facts table as different dates.

Logical Model:

Represents the facts and dimensions tables, and the relations between them.



Tables Identifications:

1. Passenger Dimension.

Represents data about passengers, like passenger **Name**, **ID**, **Address**, and **Phone Number**. In addition to their **Subscription Type**: **GOLD**, **PLATINUM**, or **TITANUM** provide different benefits, and the **Total Miles** for each passenger.

1. Flight Dimension.

Represents data about the flight, like flight **ID**, **Departure Airport**, **Arrival Airport**, **Airline Company Name**, the **Miles of the Flight**, the **Departure Date**, the **Arrival Date**, and the **Total Number of Passengers** on this flight.

1. Date Dimension.

Represents the **Minute**, **Hour**, **Day**, **Month**, **Quarter**, and **Year** for each date stored for different business needs.

1. Class Dimension.

Represents the passengers’ reservation class for each flight, it can be **First Class**, **Business Class**, or **Economy Class**.

1. Reservation Channel Dimension.

Represents the channel that the ticket was purchased from. It can be **Website**, **Office**, or in the **Airport**. It also includes the payment method if it was via **Visa**, **Bank**, or **Cash**.

1. Fare Base Dimension.

Represents if the passenger fully paid the ticket price, or with a discount.  
this represents using code numbers which:

**111 means Full Fare First Class.  
112 means Discount Fare First Class.  
221 means Full Fare Business Class.  
222 means Discount Fare Business Class.  
331 means Full Fare Economy Class.  
332 means Discount Fare Economy Class.**

1. Hotel Dimension.

Represents data about the hotels that the passengers stayed on during the transit flights, like hotel **ID**, **Name,** and its **Location**.

1. Customer Services Dimension.

Represents data about the customer services employees who are responsible for customer care interactions, like employee **ID**, **Name**, and his **Office** **Number**.

1. Interaction Dimension.

Represents the passenger’s different interactions with customer services, it contains the **Interaction Time** if it was **Before**, **Within**, or **After** the flight, and the **Interaction Type** if it was **Feedback**, **Inquiry**, or **Complaint.**

1. Reservation Fact.

Represents the reservation processes and the data needed for it, it measures the **Discount** for each passenger based on his subscription:

**10% discount for GOLD subscribers.  
15% discount for PLATINUM subscribers.  
25% discount for TITANIUM subscribers.**

It also measures the **Total Price** after the discount and generates the passenger’s **Seat Number** after the reservation process is done.

1. Transit Hotel Fact.

Represents the hotel reservation process and the **Number of Nights** that each passenger stayed based on the **Arrival Time** to the hotel, and the **Departure Time** from it. Each passenger is assigned to a hotel based on the **Ticket Number** he owned.

1. Customer Care Fact.

Represents the passenger’s different interactions and the customer service’s employee who is responsible for it. It measures the **Problem Severity** based on its **Type** and **Time** to work on different interactions based on its priorities. It represents like:

**1 for Complaint Within the flight.  
2 for Complaint Before the flight.  
3 for Complaint After the flight.  
4 for Inquiry Within the flight.  
5 for Feedback Within the flight.  
6 for Inquiry Before the flight.  
7 for Feedback Before the flight.  
8 for Feedback After the flight.  
9 for Inquiry After the flight.**

Assumptions:

1. The passengers are considered as frequent flyers based on the **Number of Miles** that are assigned to their profiles.
2. The model assume that each passenger has a different subscription: **GOLD**, **PLATINUM**, or **TITANUM**. Each passenger can subscribe to one subscription that provides different benefits to him and can change it at any time.
3. The passenger table can apply the **Slowly Changed Dimension** concept on the **Subscription** attribute, that it may change its value over the time.
4. To detect if the passenger paid the ticket using a discount or not, we used the **Fare Base** dimension that stored this information for some business needs
5. The discount is computed for each passenger based on his subscription, that the **GOLD** subscribers can have a **10%** discount on their ticket price, the **PLATINUM** subscribers can have a **15%** discount on their ticket price, and the **TITANUM** subscribers can have a **25%** on their ticket price.
6. The **Total Price = Actual Price – (Discount \* Actual Price)**;that the actual price is the ticket price stored in the Flight dimension, and the discount is the value that computed based on the passengers’ subscription.
7. The **Seat Number** is a unique number generated for each passenger with each reservation process.
8. The **Number of Nights = Arrival Date – Departure date** from the hotel.
9. The **Problem Severity** computes based on the **Interaction Time** and **Interaction Type**. It assumes that **Complaints** always have the highest priority even if it is before or after the flight.

Physical Model:

The creation of the tables using SQL queries. In this project, PL SQL is used for all queries.

1. Create Passenger Dimension:

CREATE TABLE passenger\_dim (

passenger\_key NUMBER CONSTRAINT pk\_cons PRIMARY KEY,

passenger\_id NUMBER CONSTRAINT unq\_cons UNIQUE,

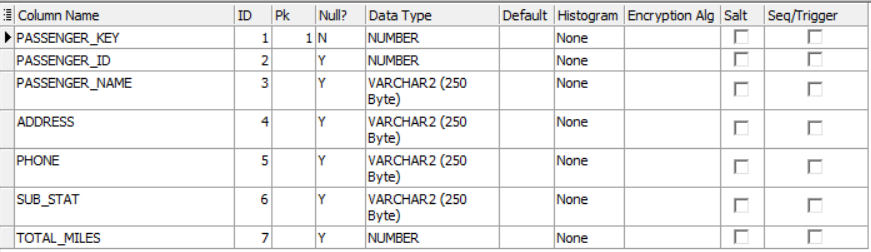
passenger\_name VARCHAR(250),

address VARCHAR(250),

phone VARCHAR(250),

sub\_stat VARCHAR(250),

total\_miles NUMBER );



1. Create Flight Dimension:

CREATE TABLE flight\_dim (

flight\_key NUMBER CONSTRAINT flight\_pk\_cons PRIMARY KEY,

flight\_id NUMBER CONSTRAINT flight\_unq\_cons UNIQUE,

depature\_airport VARCHAR(250),

arraival\_airport VARCHAR(250),

comany\_name VARCHAR(250),

actual\_price NUMBER,

flight\_miles NUMBER,

depature\_date DATE,

arraival\_date DATE,

total\_passenger\_count NUMBER );

A picture containing table

Description automatically generated

1. Create Date Dimension:

CREATE TABLE date\_dim (

date\_key NUMBER CONSTRAINT date\_pk\_cons PRIMARY KEY,

minitue NUMBER,

hours NUMBER,

days NUMBER,

months NUMBER,

quarter NUMBER,

years NUMBER );

Table

Description automatically generated

1. Create Class Dimension:

CREATE TABLE class\_dim (

class\_key NUMBER CONSTRAINT class\_pk\_cons PRIMARY KEY,

class\_id NUMBER CONSTRAINT class\_unq\_cons UNIQUE,

class\_type VARCHAR(250) );

Graphical user interface, application

Description automatically generated

1. Create Reservation Channel Dimension:

CREATE TABLE reservation\_channel\_dim (

channel\_key NUMBER CONSTRAINT channel\_pk\_con PRIMARY KEY,

channel\_id NUMBER CONSTRAINT channel\_unq\_cond UNIQUE,

reservation\_channel\_type VARCHAR(250),

payment\_method VARCHAR(250) );

Graphical user interface, application

Description automatically generated

1. Create Fare Base Dimension:

CREATE TABLE fare\_base\_dim (

fare\_base\_key NUMBER CONSTRAINT fare\_pk\_cons PRIMARY KEY,

fare\_base\_id NUMBER CONSTRAINT fare\_unq\_cons UNIQUE,

fare\_base\_type VARCHAR(250) );

Graphical user interface, application

Description automatically generated

1. Create Hotel Dimension:

CREATE TABLE hotel\_dim (

hotel\_key NUMBER CONSTRAINT hotel\_pk\_cons PRIMARY KEY,

hotel\_id NUMBER CONSTRAINT hotel\_unq\_cons UNIQUE,

hotel\_name VARCHAR(250),

hotel\_location VARCHAR(250) );

Graphical user interface, table

Description automatically generated

1. Create Customer Services Dimension:

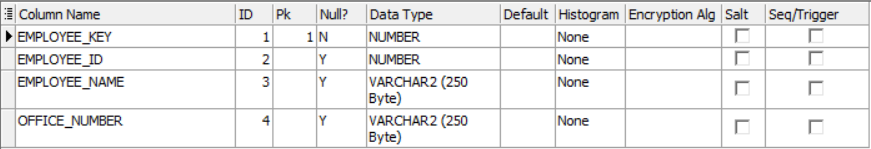
CREATE TABLE cutomer\_services\_dim (

employee\_key NUMBER CONSTRAINT emp\_pk\_cons PRIMARY KEY,

employee\_id NUMBER CONSTRAINT emp\_unq\_cons UNIQUE,

employee\_name VARCHAR(250),

office\_number VARCHAR(250) );



1. Create Interaction Dimension:

CREATE TABLE interation\_dim (

interaction\_key NUMBER CONSTRAINT inter\_pk\_cons PRIMARY KEY,

interaction\_id NUMBER CONSTRAINT inter\_unq\_cons UNIQUE,

interaction\_time VARCHAR(250),

interaction\_type VARCHAR(250),

discription VARCHAR(250) );

Table

Description automatically generated

1. Create Reservation Fact:

CREATE TABLE reservation\_fact (

reservation\_key NUMBER CONSTRAINT res\_pk\_cons PRIMARY KEY,

passenger\_key NUMBER,

flight\_key NUMBER,

reservation\_channel\_key NUMBER,

class\_key NUMBER,

fare\_base\_key NUMBER,

date\_key NUMBER,

discount NUMBER,

total\_price NUMBER,

seat\_number NUMBER );

ALTER TABLE reservation\_fact

ADD (

CONSTRAINT pass\_fk\_cons FOREIGN KEY(passenger\_key) REFERENCES passenger\_dim(passenger\_key),

CONSTRAINT flight\_fk\_con FOREIGN KEY(flight\_key) REFERENCES flight\_dim(flight\_key),

CONSTRAINT channel\_fk\_cons FOREIGN KEY(reservation\_channel\_key) REFERENCES reservation\_channel\_dim(channel\_key),

CONSTRAINT class\_fk\_cons FOREIGN KEY(class\_key) REFERENCES class\_dim(class\_key),

CONSTRAINT fare\_fk\_cons FOREIGN KEY(fare\_base\_key) REFERENCES fare\_base\_dim(fare\_base\_key),

CONSTRAINT date\_fk\_cons FOREIGN KEY(date\_key) REFERENCES date\_dim(date\_key) );

Table

Description automatically generated with medium confidence

1. Create Transit Hotel Fact:

CREATE TABLE transit\_hotel\_fact (

flight\_key NUMBER,

passenger\_key NUMBER,

reservation\_channel\_key NUMBER,

hotel\_key NUMBER,

date\_key NUMBER,

ticket\_number NUMBER,

number\_of\_nights NUMBER );

ALTER TABLE transit\_hotel\_fact

ADD (

CONSTRAINT flight\_fk\_conss FOREIGN KEY(flight\_key) REFERENCES flight\_dim(flight\_key),

CONSTRAINT pass\_fk\_conss FOREIGN KEY(passenger\_key) REFERENCES passenger\_dim(passenger\_key),

CONSTRAINT rese\_fk\_conss FOREIGN KEY(reservation\_channel\_key) REFERENCES reservation\_channel\_dim(channel\_key),

CONSTRAINT hotel\_fk\_conss FOREIGN KEY(hotel\_key) REFERENCES hotel\_dim(hotel\_key),

CONSTRAINT date\_fk\_conss FOREIGN KEY(date\_key) REFERENCES date\_dim(date\_key));

Table

Description automatically generated

1. Create Customer Care Fact:

CREATE TABLE customer\_care\_fact(

case\_key NUMBER CONSTRAINT case\_pk\_cons PRIMARY KEY,

passenger\_key NUMBER,

flight\_key NUMBER,

hotel\_key NUMBER,

date\_key NUMBER,

interaction\_key NUMBER,

employee\_key NUMBER,

problem\_severity NUMBER );

ALTER TABLE customer\_care\_fact

ADD (

CONSTRAINT pass\_fk\_consss FOREIGN KEY(passenger\_key) REFERENCES passenger\_dim(passenger\_key),

CONSTRAINT flight\_fk\_consss FOREIGN KEY(flight\_key) REFERENCES flight\_dim(flight\_key),

CONSTRAINT hotel\_fk\_consss FOREIGN KEY(hotel\_key) REFERENCES hotel\_dim(hotel\_key),

CONSTRAINT date\_fk\_consss FOREIGN KEY(date\_key) REFERENCES date\_dim(date\_key),

CONSTRAINT inter\_fk\_consss FOREIGN KEY(interaction\_key) REFERENCES interation\_dim(interaction\_key),

CONSTRAINT emp\_fk\_consss FOREIGN KEY(employee\_key) REFERENCES cutomer\_services\_dim(employee\_key) );

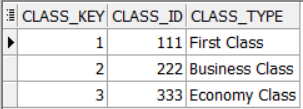
Table

Description automatically generated

Insert Dummy Data into The Physical Tables:

Based on the HR schema that ORACLE provides for leaning, and using some other techniques and functions, some dummy data inserted into the tables for testing and answering some business questions.

1. Class Dimension Data:



1. Customer Services Dimension Data:

Table

Description automatically generated

1. Date Dimension Data:

Table

Description automatically generated

1. Fare Base Dimension Data:

Table

Description automatically generated

1. Flight Dimension Data:

Table

Description automatically generated

1. Hotel Dimension Data:

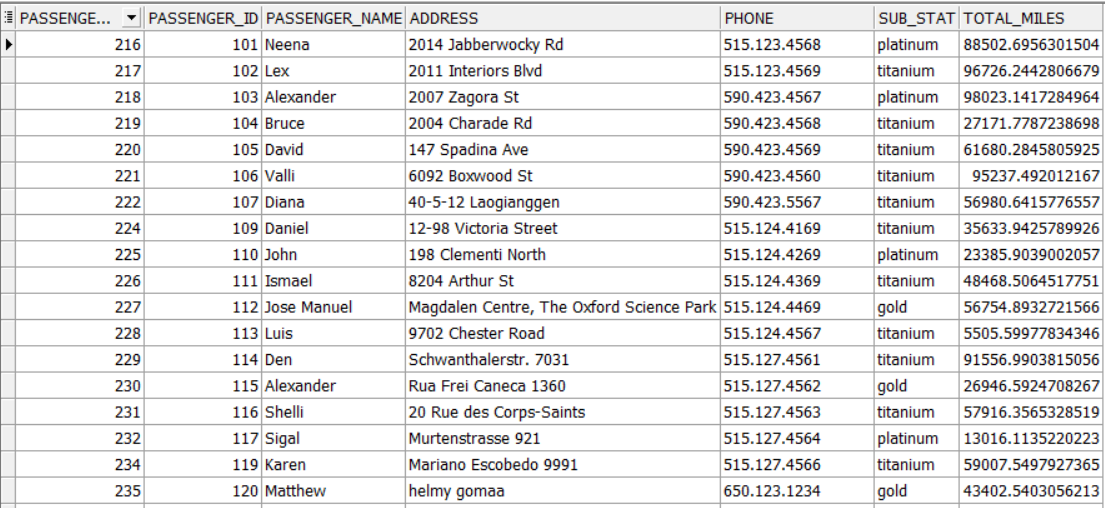


1. Interaction Dimension Data:

Table

Description automatically generated

1. Passenger Dimension Data:



1. Reservation Channel Dimension Data:

Table

Description automatically generated

1. Customer Care Fact Data:

Table

Description automatically generated

1. Reservation Fact Data:

Table

Description automatically generated with medium confidence

1. Transit Hotel Fact Data:

Table

Description automatically generated

Insert The Measured Data Into Fact Tables

1. Insert Data into Discount Major in Reservation Fact:

DECLARE

v\_sub VARCHAR(250);

CURSOR Cur IS

SELECT RESERVATION\_KEY, PASSENGER\_KEY, DISCOUNT

from RESERVATION\_FACT

FOR UPDATE OF DISCOUNT;

BEGIN

FOR Rec IN Cur

LOOP

SELECT sub\_stat

INTO v\_sub

FROM passenger\_dim

WHERE passenger\_key = Rec.passenger\_key;

IF(v\_sub = 'gold') THEN

UPDATE RESERVATION\_FACT

SET discount = 0.10

WHERE CURRENT OF Cur;

ELSIF(v\_sub = 'platinum') THEN

UPDATE RESERVATION\_FACT

SET discount = 0.15

WHERE CURRENT OF Cur;

ELSIF(v\_sub = 'titanium') THEN

UPDATE RESERVATION\_FACT

SET discount = 0.25

WHERE CURRENT OF Cur;

END IF;

END LOOP;

END;

1. Insert Data into Total Price Major in Reservation Fact:

DECLARE

v\_price NUMBER;

v\_discount NUMBER;

CURSOR Cur IS

SELECT RESERVATION\_KEY, flight\_key , total\_price, discount

from RESERVATION\_FACT

FOR UPDATE OF total\_price;

BEGIN

FOR Rec IN Cur

LOOP

SELECT actual\_price

INTO v\_price

FROM flight\_dim

WHERE flight\_key = Rec.flight\_key;

UPDATE RESERVATION\_FACT

SET total\_price = v\_price – (v\_price\*Rec.discount)

WHERE CURRENT OF Cur;

END LOOP;

END;

1. Insert Data into Seat Number Major in Reservation Fact:

DECLARE

CURSOR Cur IS

SELECT flight\_key , total\_price, seat\_number

from RESERVATION\_FACT

FOR UPDATE OF seat\_number;

BEGIN

FOR Rec IN Cur

LOOP

IF (Rec.flight\_key IS IN (SELECT flight\_key

FROM flight\_dim))

THEN

UPDATE RESERVATION\_FACT

SET seat\_number = TRUNC(***dbms\_random.value***(1,300),0)

WHERE CURRENT OF Cur;

END IF;

END LOOP;

END;

1. Insert Data into Number of Nights Major in Transit Hotel Fact:

DECLARE

v\_date DATE;

CURSOR Cur IS

SELECT FLIGHT\_KEY ,NUMBER\_OF\_NIGHTS

from TRANSIT\_HOTEL\_FACT

FOR UPDATE OF NUMBER\_OF\_NIGHTS;

BEGIN

FOR Rec IN Cur

LOOP

SELECT arraival\_date

INTO v\_date

FROM flight\_dim

WHERE flight\_key = Rec.flight\_key;

UPDATE TRANSIT\_HOTEL\_FACT

SET NUMBER\_OF\_NIGHTS = TRUNC(SYSDATE - v\_date ,0)

WHERE CURRENT OF Cur;

END LOOP;

END;

1. Insert Data into Problem Severity Major in Customer Care Fact:

DECLARE

v\_type VARCHAR(250);

v\_time VARCHAR(250);

CURSOR Cur IS

SELECT CASE\_KEY ,PROBLEM\_SEVERITY, INTERACTION\_KEY

from CUSTOMER\_CARE\_FACT

FOR UPDATE OF PROBLEM\_SEVERITY;

BEGIN

FOR Rec IN Cur

LOOP

SELECT INTERACTION\_TIME, INTERACTION\_TYPE

INTO v\_time, v\_type

FROM INTERATION\_DIM

WHERE INTERACTION\_KEY = Rec.INTERACTION\_KEY;

IF (v\_time = 'within flight' AND v\_type = 'complaint')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 1

WHERE CURRENT OF Cur;

ELSIF (v\_time = 'before flight' AND v\_type = 'complaint')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 2

WHERE CURRENT OF Cur;

ELSIF (v\_time = 'after flight' AND v\_type = 'complaint')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 3

WHERE CURRENT OF Cur;

ELSIF (v\_time = 'within flight' AND v\_type = 'inquiry')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 4

WHERE CURRENT OF Cur;

ELSIF (v\_time = 'within flight' AND v\_type = 'feedback')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 5

WHERE CURRENT OF Cur;

ELSIF (v\_time = 'before flight' AND v\_type = 'inquiry')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 6

WHERE CURRENT OF Cur;

ELSIF (v\_time = 'before flight' AND v\_type = 'feedback')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 7

WHERE CURRENT OF Cur;

ELSIF (v\_time = 'after flight' AND v\_type = 'feedback')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 8

WHERE CURRENT OF Cur;

ELSIF (v\_time = 'after flight' AND v\_type = 'inquiry')

THEN

UPDATE CUSTOMER\_CARE\_FACT

SET PROBLEM\_SEVERITY = 9

WHERE CURRENT OF Cur;

END IF;

END LOOP;

END;

SQL Queries to Answer Business Questions:

1. Find the Most Common Arrival Destinations:

SELECT ARRAIVAL\_AIRPORT, COUNT(ARRAIVAL\_AIRPORT) AS "Number of Visits"

FROM flight\_dim

GROUP BY ARRAIVAL\_AIRPORT

ORDER BY COUNT(ARRAIVAL\_AIRPORT) DESC;



1. Find the Interactions with Type “Complaint” and the Employee Who Works on It:

SELECT i.INTERACTION\_TYPE, i.DISCRIPTION, c.EMPLOYEE\_NAME

FROM INTERATION\_DIM i JOIN CUSTOMER\_CARE\_FACT f

ON i.INTERACTION\_KEY = f.INTERACTION\_KEY

JOIN CUTOMER\_SERVICES\_DIM c

ON F.EMPLOYEE\_KEY = C.EMPLOYEE\_KEY

WHERE I.INTERACTION\_TYPE = 'complaint';



1. Find the Top Passengers with Largest Total Miles:

SELECT PASSENGER\_NAME, ROUND(TOTAL\_MILES,2)

FROM PASSENGER\_DIM

WHERE ROWNUM <= 10

ORDER BY TOTAL\_MILES DESC;

Table

Description automatically generated

1. Find the Total Airline Companies and Their Profit:

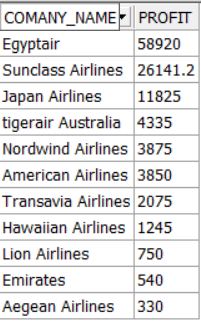
SELECT f.COMANY\_NAME, SUM(r.TOTAL\_PRICE) profit

FROM FLIGHT\_DIM f JOIN RESERVATION\_FACT r

ON f.FLIGHT\_KEY = r.FLIGHT\_KEY

GROUP BY COMANY\_NAME

ORDER BY SUM(r.TOTAL\_PRICE) DESC;



1. Find the Number of Passengers for Each Airline Company:

SELECT f.COMANY\_NAME, COUNT(p.PASSENGER\_KEY) "Number of Passengers"

FROM FLIGHT\_DIM f JOIN RESERVATION\_FACT r

ON f.FLIGHT\_KEY = r.FLIGHT\_KEY

JOIN PASSENGER\_DIM p

ON r.PASSENGER\_KEY = p.PASSENGER\_KEY

GROUP BY COMANY\_NAME

ORDER BY COUNT(p.PASSENGER\_KEY) DESC;

Table

Description automatically generated

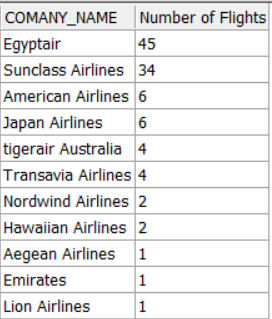
1. Find the Number of Flights for Each Airline Company:

SELECT COMANY\_NAME, COUNT(FLIGHT\_KEY) "Number of Flights"

FROM FLIGHT\_DIM

GROUP BY COMANY\_NAME

ORDER BY COUNT(FLIGHT\_KEY) DESC;



1. Find The Total Sales from Each Payment Method:

SELECT c.PAYMENT\_METHOD "Payment Method", SUM(r.TOTAL\_PRICE) profit

FROM RESERVATION\_CHANNEL\_DIM c JOIN RESERVATION\_FACT r

ON c.CHANNEL\_KEY = r.RESERVATION\_CHANNEL\_KEY

GROUP BY c.PAYMENT\_METHOD

ORDER BY SUM(r.TOTAL\_PRICE) DESC;

Table

Description automatically generated

1. Find the Total Sales from Each Reservation Channel:

SELECT reservation\_channel\_type AS "channel type", SUM(total\_price) AS "total cost"

FROM reservation\_fact rf

JOIN RESERVATION\_CHANNEL\_DIM rd

ON rf.reservation\_channel\_key = rd.channel\_key

GROUP BY reservation\_channel\_type

ORDER BY "total cost" DESC;

Table

Description automatically generated

1. Find the Average Number of Nights That Passengers Stayed in Each Hotel:

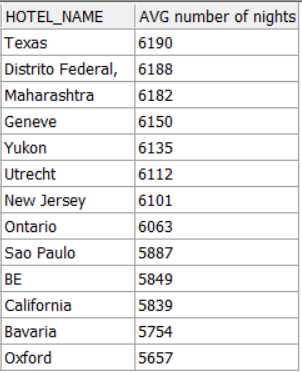
SELECT h.HOTEL\_NAME, TRUNC(AVG(n.NUMBER\_OF\_NIGHTS),0) "AVG number of nights"

FROM HOTEL\_DIM h JOIN TRANSIT\_HOTEL\_FACT n

ON H.HOTEL\_KEY = N.HOTEL\_KEY

GROUP BY h.hotel\_name

ORDER BY AVG(n.NUMBER\_OF\_NIGHTS) DESC;



1. Find The Most Received Classes at Each Company:

SELECT comany\_name AS "company" , class\_type AS "class"

FROM class\_dim cd

JOIN reservation\_fact rf

ON CD.CLASS\_KEY = RF.CLASS\_KEY

JOIN flight\_dim fd

ON rf.flight\_key= FD.FLIGHT\_KEY

GROUP BY comany\_name , class\_type ;

Table

Description automatically generated

1. Find the Frequent Passengers and How Much Miles they Passed:

SELECT PASSENGER\_ID,SUM( FLIGHT\_MILES)as "flight miles "

FROM FLIGHT\_dim

join reservation\_fact on reservation\_fact.flight\_key = FLIGHT\_DIM.FLIGHT\_KEY

join passenger\_dim on PASSENGER\_DIM.PASSENGER\_key = RESERVATION\_FACT.PASSENGER\_KEY

GROUP BY PASSENGER\_ID;

Table

Description automatically generated with medium confidence

1. Find The Percentage of Each Subscription Type:

SELECT SUB\_STAT , ROUND(COUNT(SUB\_STAT)/ (SELECT COUNT(\*) FROM PASSENGER\_dim) \*100,1)||'%'

FROM PASSENGER\_dim

GROUP BY SUB\_STAT;

Table

Description automatically generated