Airline → Data Warehouse

Time	Prepared By	Gate	Status
12:01	HUSSEIN	A19	ON TIME
12:07	MARINA	B26	ON TIME
13:45	RADWA	C10	ON TIME

AirOps Data Warehouse

By:

Marina Fawzy
Hussein Khaled
Radwa Esmaiel

1. Overview of the Project

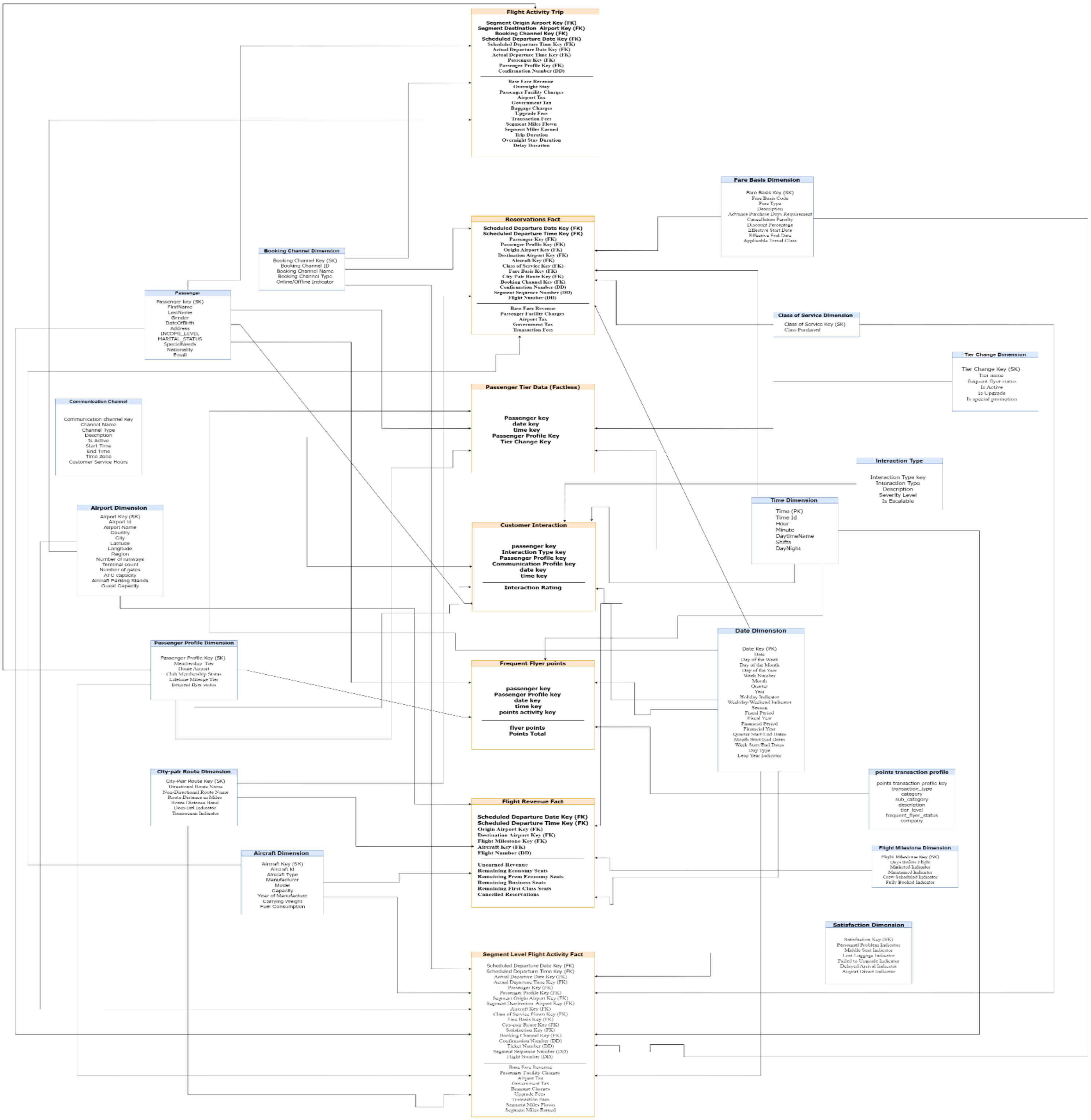
The goal of this project is to build a data warehouse for an airline to analyze its operations and improve customer satisfaction. We want to give executives a way to track flight activity, reservations, flight revenue, and segment-level details. By collecting both operational data and customer-centric information, the airline can see the full picture of its business.

Business Processes Involved

Flight Activity
Reservations
Flight Revenue
Frequent Flyer Program
Customer Interaction
Passenger Tier Changes

2. Data Model Design

The data warehouse employs a dimensional structure with a star schema design to support analytical processing and business insights. This design was chosen for several key advantages, including scalability, query performance, and ease of understanding. It also aligns with our goal to maintain a logical separation of data, while still ensuring strong connections between related information. Moreover, it provides the flexibility needed to allow for future expansion as business requirements evolve.

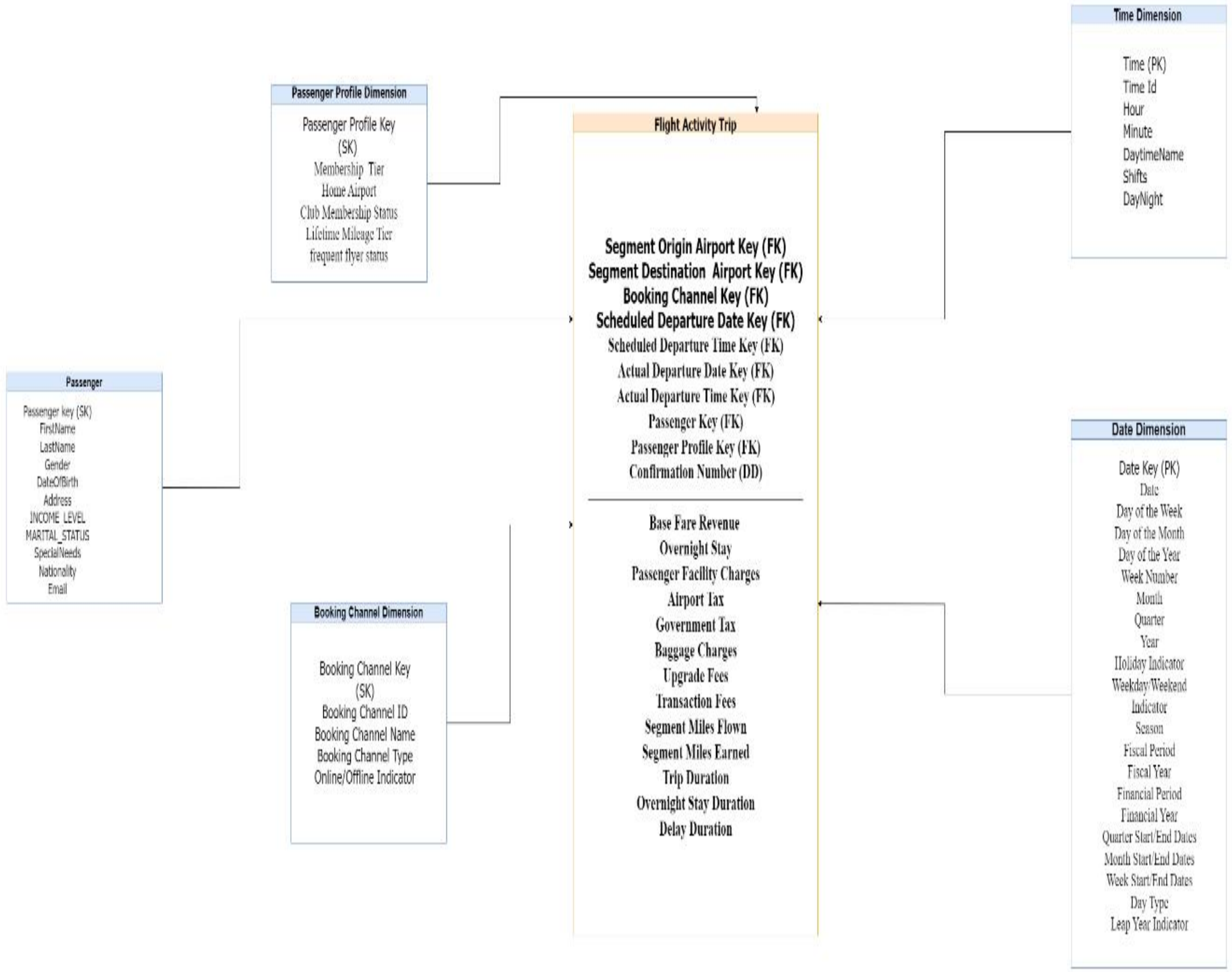


3. Data Model Components

Our data warehouse design encompasses multiple fact tables, each serving a distinct role in capturing and analyzing key aspects of airline operations. Here's an overview of the fact tables and how they contribute to business insights.

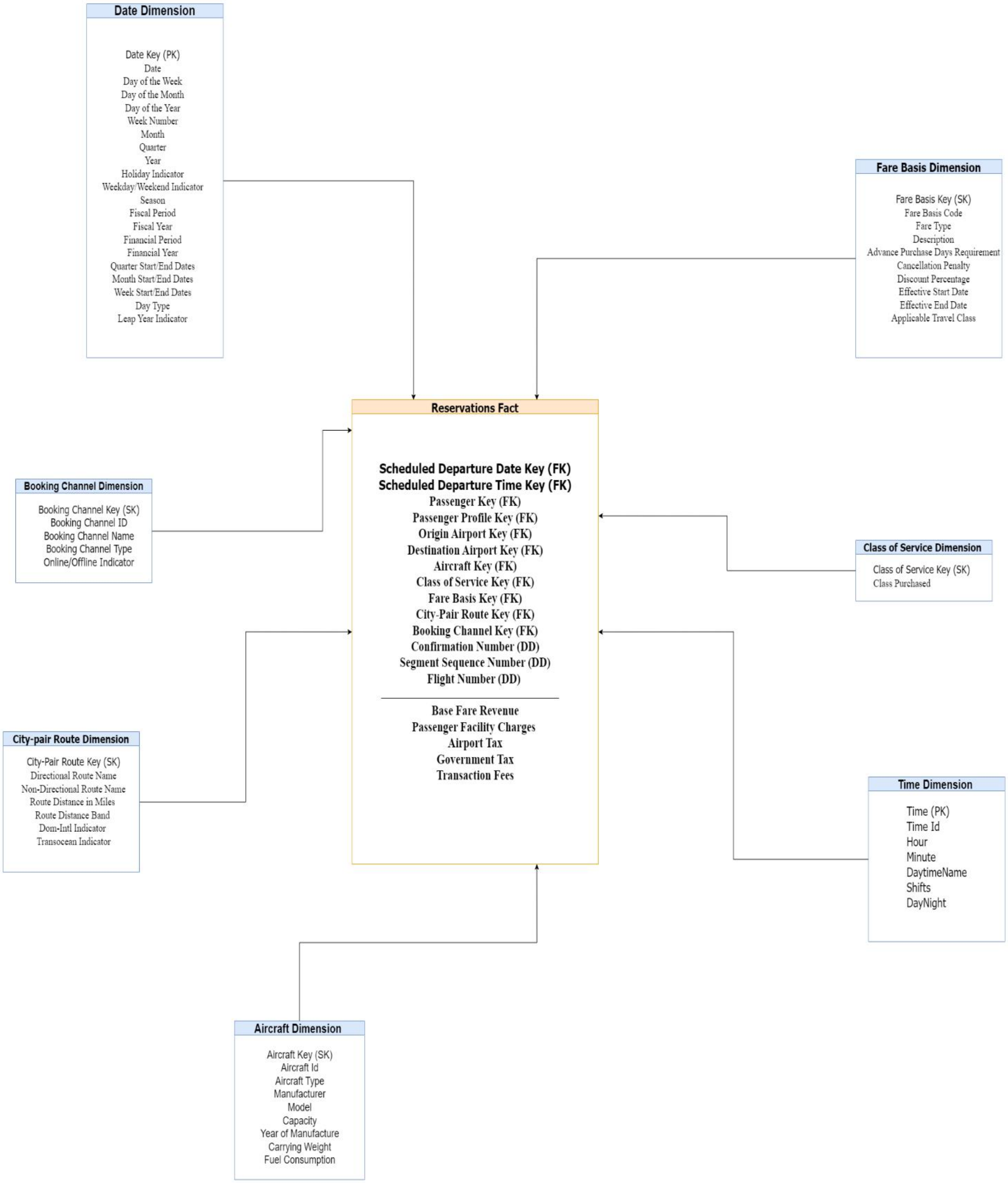
Flight Activity Trip

The Flight Activity Trip fact table provides detailed information about flight trips (flight segments grouped into full trips) and related metrics. It includes data on origin and destination airports, booking channels, passenger profiles, and various financial elements like fare revenue, taxes, and fees.



Reservations Fact

The Reservations Fact table captures comprehensive reservation details, including passenger data, origin and destination airports, aircraft types, and class of service. It provides a detailed view of how reservations are made and the financial implications.



Date Dimension

Date Key (PK)
Date
Day of the Week
Day of the Month
Day of the Year
Week Number
Month
Quarter
Year
Holiday Indicator
Weekday/Weekend Indicator
Season
Fiscal Period
Fiscal Year
Financial Period
Financial Year
Quarter Start/End Dates
Month Start/End Dates
Week Start/End Dates
Day Type
Leap Year Indicator

Fare Basis Dimension

Fare Basis Key (SK)
Fare Basis Code
Fare Type
Description
Advance Purchase Days Requirement
Cancellation Penalty
Discount Percentage
Effective Start Date
Effective End Date
Applicable Travel Class

Reservations Fact

Scheduled Departure Date Key (FK)
Scheduled Departure Time Key (FK)
Passenger Key (FK)
Passenger Profile Key (FK)
Origin Airport Key (FK)
Destination Airport Key (FK)
Aircraft Key (FK)
Class of Service Key (FK)
Fare Basis Key (FK)
City-Pair Route Key (FK)
Booking Channel Key (FK)
Confirmation Number (DD)
Segment Sequence Number (DD)
Flight Number (DD)

Base Fare Revenue
Passenger Facility Charges
Airport Tax
Government Tax
Transaction Fees

Booking Channel Dimension

Booking Channel Key (SK)
Booking Channel ID
Booking Channel Name
Booking Channel Type
Online/Offline Indicator

Class of Service Dimension

Class of Service Key (SK)
Class Purchased

City-pair Route Dimension

City-Pair Route Key (SK)
Directional Route Name
Non-Directional Route Name
Route Distance in Miles
Route Distance Band
Dom-Intl Indicator
Transocean Indicator

Time Dimension

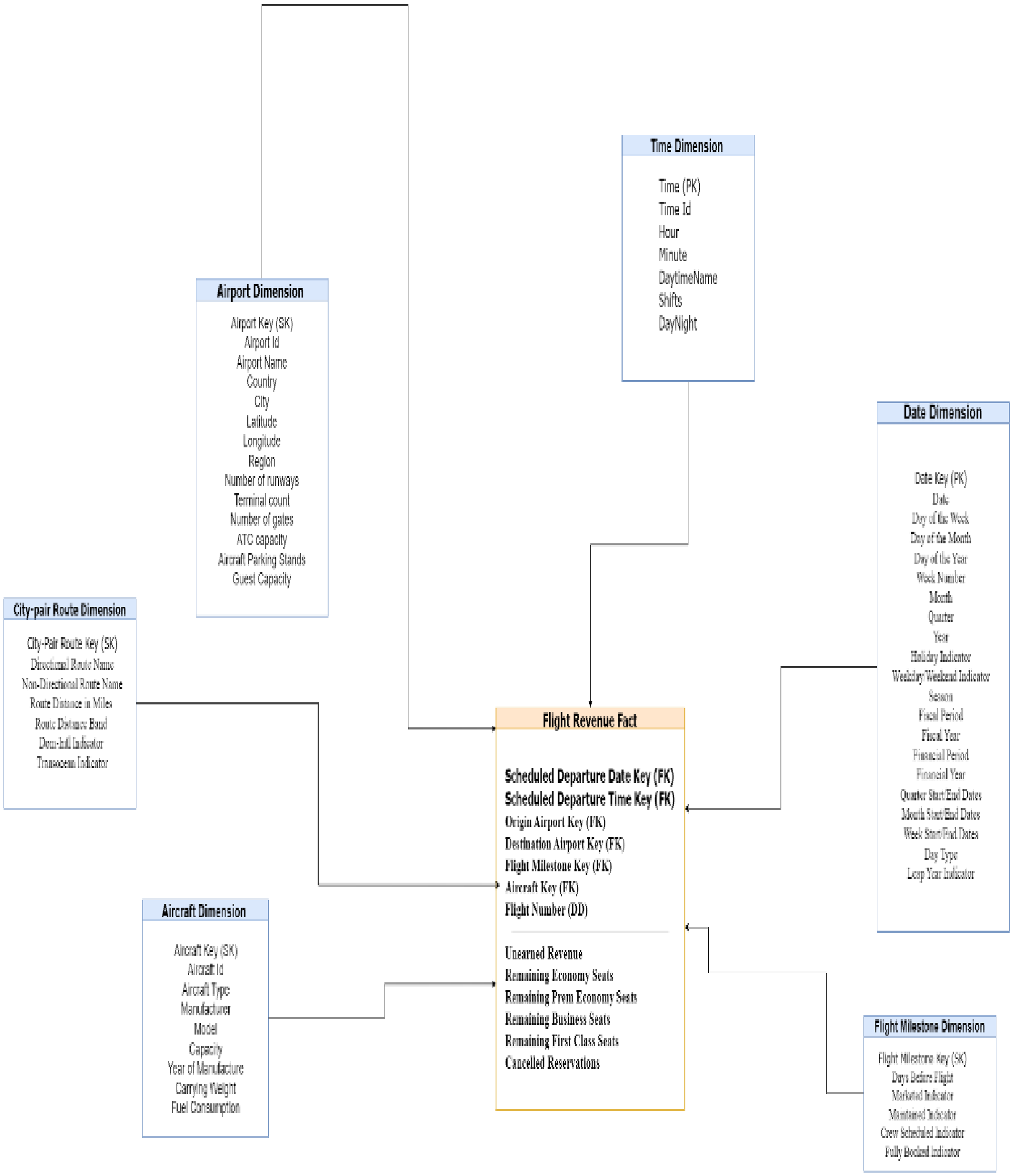
Time (PK)
Time Id
Hour
Minute
DaytimeName
Shifts
DayNight

Aircraft Dimension

Aircraft Key (SK)
Aircraft Id
Aircraft Type
Manufacturer
Model
Capacity
Year of Manufacture
Carrying Weight
Fuel Consumption

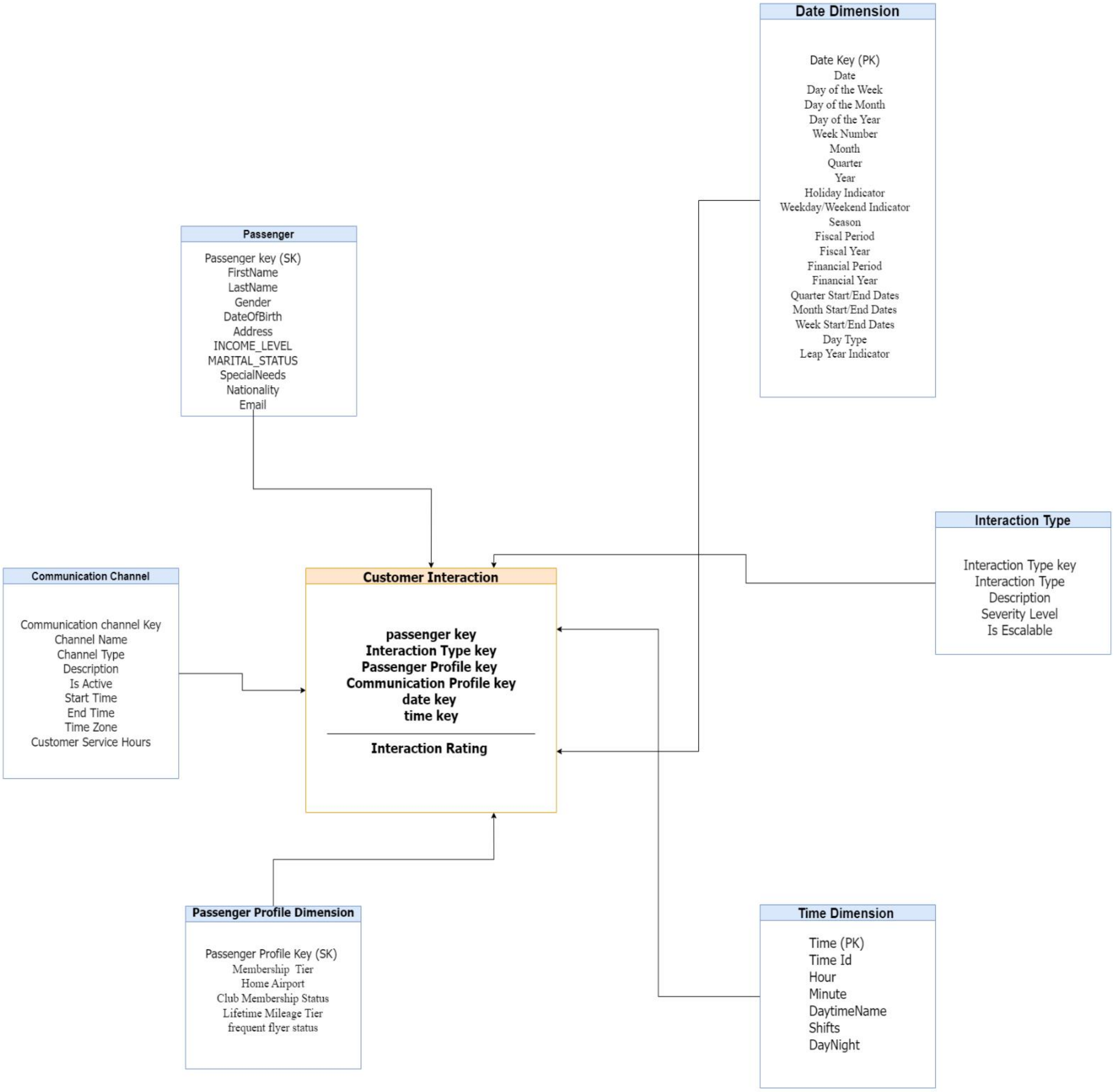
Flight Revenue Fact

The Flight Revenue Fact table focuses on the revenue aspects of individual flights, it’s a daily periodic snapshot fact table, tracking flights info 90 days before takeoff, including unearned revenue and the number of remaining seats by class. It helps the airline track revenue generation and seat capacity utilization.



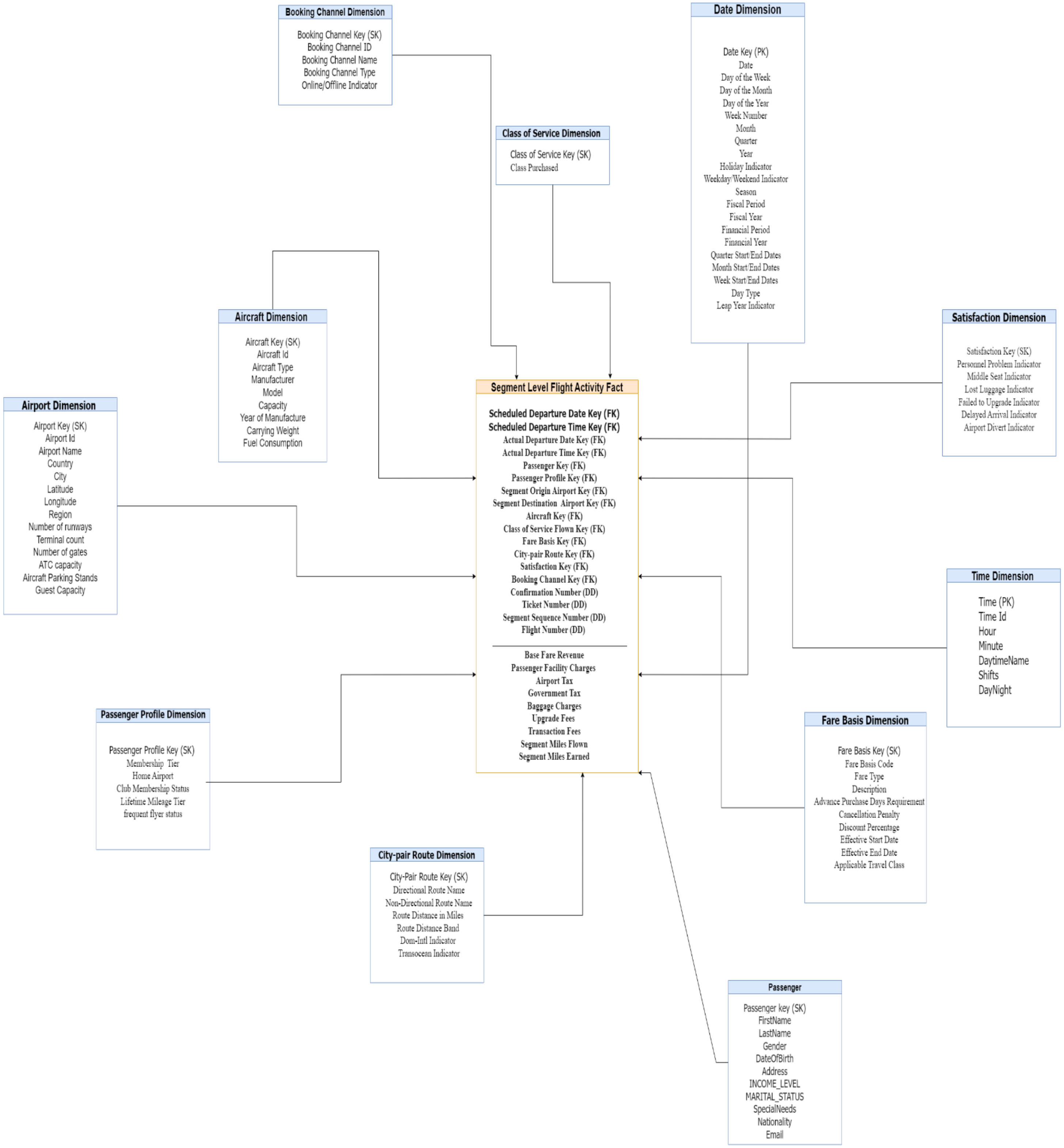
Customer Interaction

The Customer Interaction fact table captures various interactions between the airline and its customers, including inquiries, complaints, and feedback. This table is crucial for analyzing customer service quality and understanding passenger satisfaction.



Segment Level Flight Activity:

The Segment Level Flight Activity Fact table provides granular insights into flight segments, capturing segment-specific information such as departure and arrival times, passenger data, and various financial metrics.

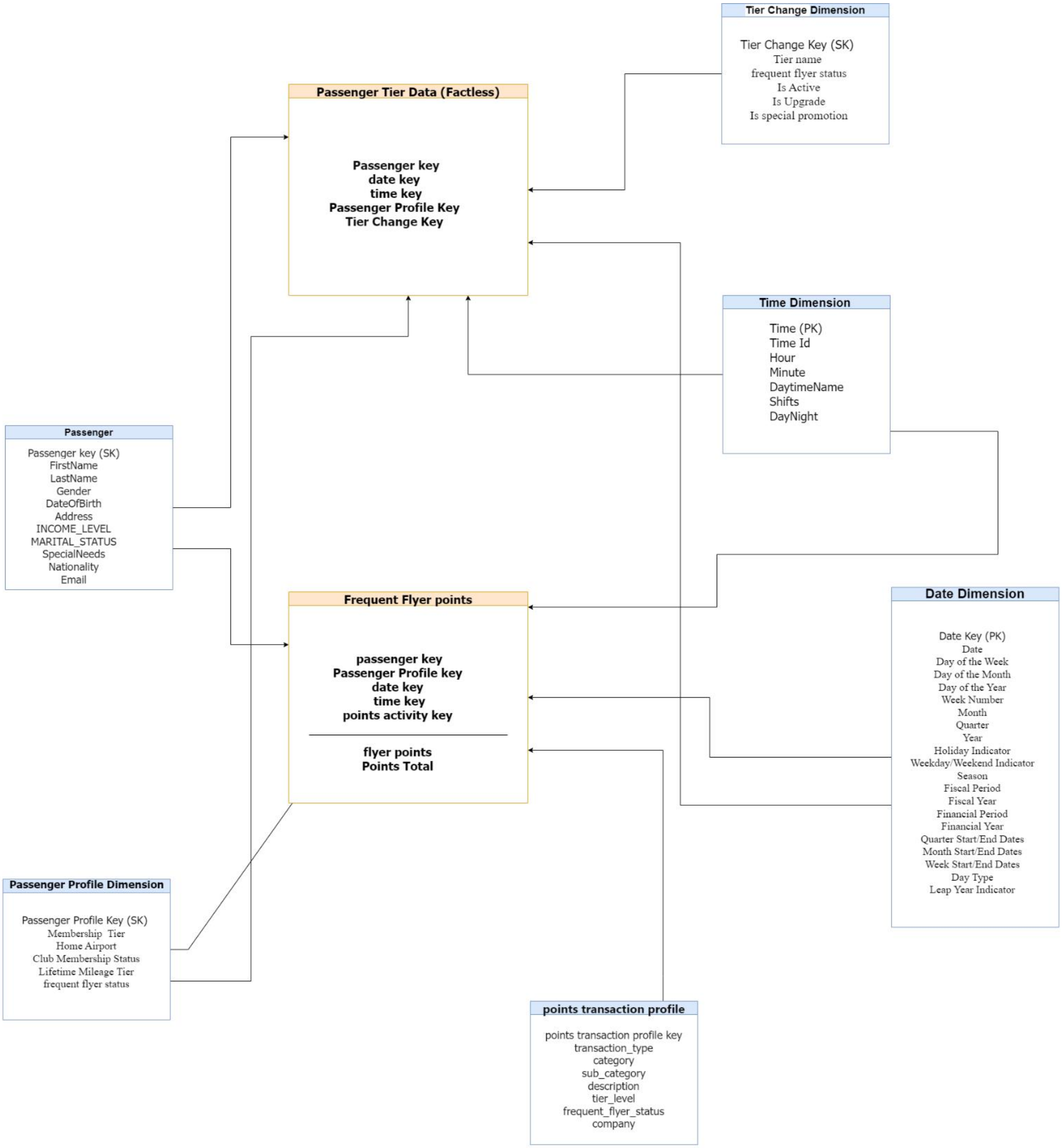


Frequent Flyer Points

The Frequent Flyer Points fact table tracks the activities of the airline's loyalty program. It helps monitor points accumulation, redemptions, and total points earned by frequent flyers.

Passenger Tier Data (Factless)

The Passenger Tier Data factless fact table tracks changes in passenger tiers and related activities. This table is used to analyze passenger tier upgrades and other tier-related interactions.



Physical Model

Table Structure

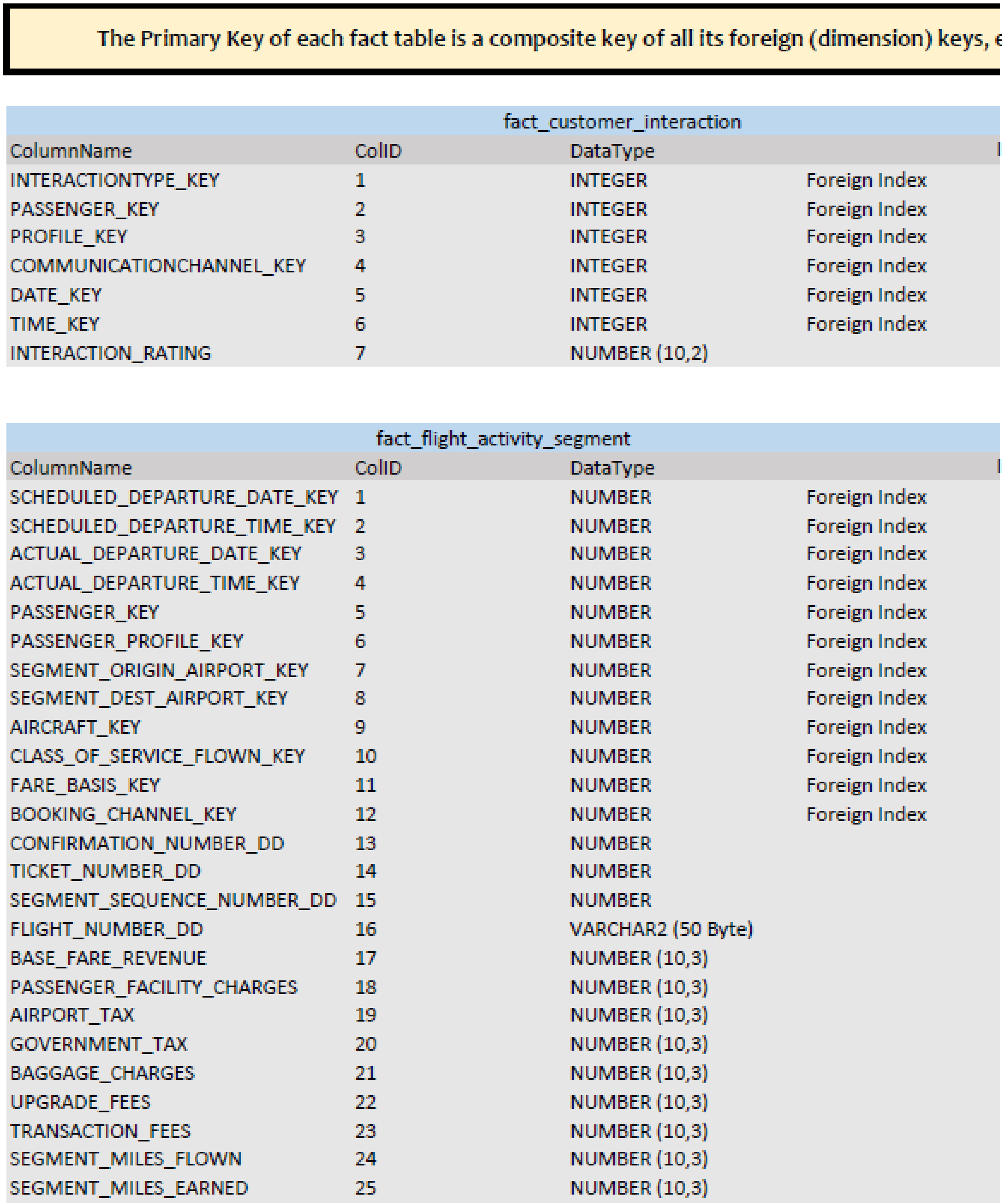
In the Excel-based physical model, each sheet represents a table in the data warehouse. The structure includes:

- Table Name: The name of the table as implemented in the database.
- Columns: A list of columns with their data types, sizes, and constraints.
- Primary Keys: The columns that form the primary key for each table.
- Foreign Keys: The relationships between tables, showing which columns are used as foreign keys.
- Indexes: A list of indexes, including primary, composite, and bitmap indexes.

Indexes and Their Usage

Indexes are a critical component of the physical model, as they improve query performance and data retrieval efficiency. Here's an overview of the possible indexes and how they are used in the data warehouse:

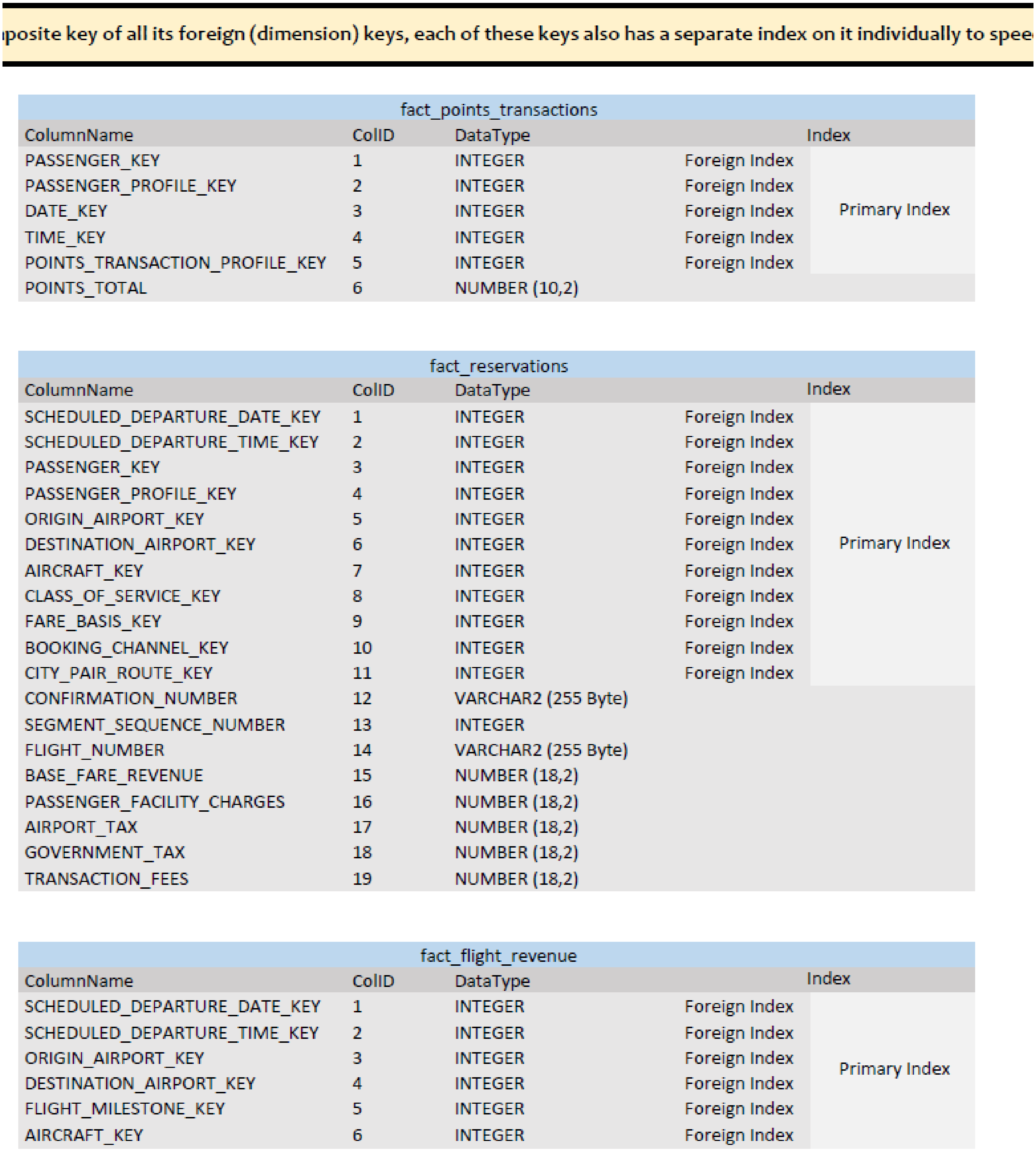
- Primary Indexes: These are created on the primary keys of the tables, ensuring unique identification for each record and enabling fast lookups. For example, the Passenger Tier Data table has a composite primary index on the following columns: Passenger Key, Date Key, Time Key, Passenger Profile Key, and Tier Change Key.
- Composite Indexes: Composite indexes are used to improve query performance for multi-column searches. For example, the Frequent Flyer Points table has a composite index on Passenger Key, Passenger Profile Key, Date Key, Time Key, and Points Activity Key.
- Bitmap Indexes: Bitmap indexes are ideal for columns with low cardinality. They offer efficient filtering and can significantly improve the performance of complex queries. In our physical model, bitmap indexes are used on Frequent Flyer Status, Tier Name, and Day Time, as these columns have a limited set of unique values.



The Primary Key of each fact table is a composite key of all its foreign (dimension) keys, e

fact_customer_interaction				
ColumnName	ColID	DataType		I
INTERACTIONTYPE_KEY	1	INTEGER	Foreign Index	
PASSENGER_KEY	2	INTEGER	Foreign Index	
PROFILE_KEY	3	INTEGER	Foreign Index	
COMMUNICATIONCHANNEL_KEY	4	INTEGER	Foreign Index	
DATE_KEY	5	INTEGER	Foreign Index	
TIME_KEY	6	INTEGER	Foreign Index	
INTERACTION_RATING	7	NUMBER (10,2)		

fact_flight_activity_segment				
ColumnName	ColID	DataType		I
SCHEDULED_DEPARTURE_DATE_KEY	1	NUMBER	Foreign Index	
SCHEDULED_DEPARTURE_TIME_KEY	2	NUMBER	Foreign Index	
ACTUAL_DEPARTURE_DATE_KEY	3	NUMBER	Foreign Index	
ACTUAL_DEPARTURE_TIME_KEY	4	NUMBER	Foreign Index	
PASSENGER_KEY	5	NUMBER	Foreign Index	
PASSENGER_PROFILE_KEY	6	NUMBER	Foreign Index	
SEGMENT_ORIGIN_AIRPORT_KEY	7	NUMBER	Foreign Index	
SEGMENT_DEST_AIRPORT_KEY	8	NUMBER	Foreign Index	
AIRCRAFT_KEY	9	NUMBER	Foreign Index	
CLASS_OF_SERVICE_FLOWN_KEY	10	NUMBER	Foreign Index	
FARE_BASIS_KEY	11	NUMBER	Foreign Index	
BOOKING_CHANNEL_KEY	12	NUMBER	Foreign Index	
CONFIRMATION_NUMBER_DD	13	NUMBER		
TICKET_NUMBER_DD	14	NUMBER		
SEGMENT_SEQUENCE_NUMBER_DD	15	NUMBER		
FLIGHT_NUMBER_DD	16	VARCHAR2 (50 Byte)		
BASE_FARE_REVENUE	17	NUMBER (10,3)		
PASSENGER_FACILITY_CHARGES	18	NUMBER (10,3)		
AIRPORT_TAX	19	NUMBER (10,3)		
GOVERNMENT_TAX	20	NUMBER (10,3)		
BAGGAGE_CHARGES	21	NUMBER (10,3)		
UPGRADE_FEES	22	NUMBER (10,3)		
TRANSACTION_FEES	23	NUMBER (10,3)		
SEGMENT_MILES_FLOWN	24	NUMBER (10,3)		
SEGMENT_MILES_EARNED	25	NUMBER (10,3)		



ıposite key of all its foreign (dimension) keys, each of these keys also has a separate index on it individually to spee

fact_points_transactions

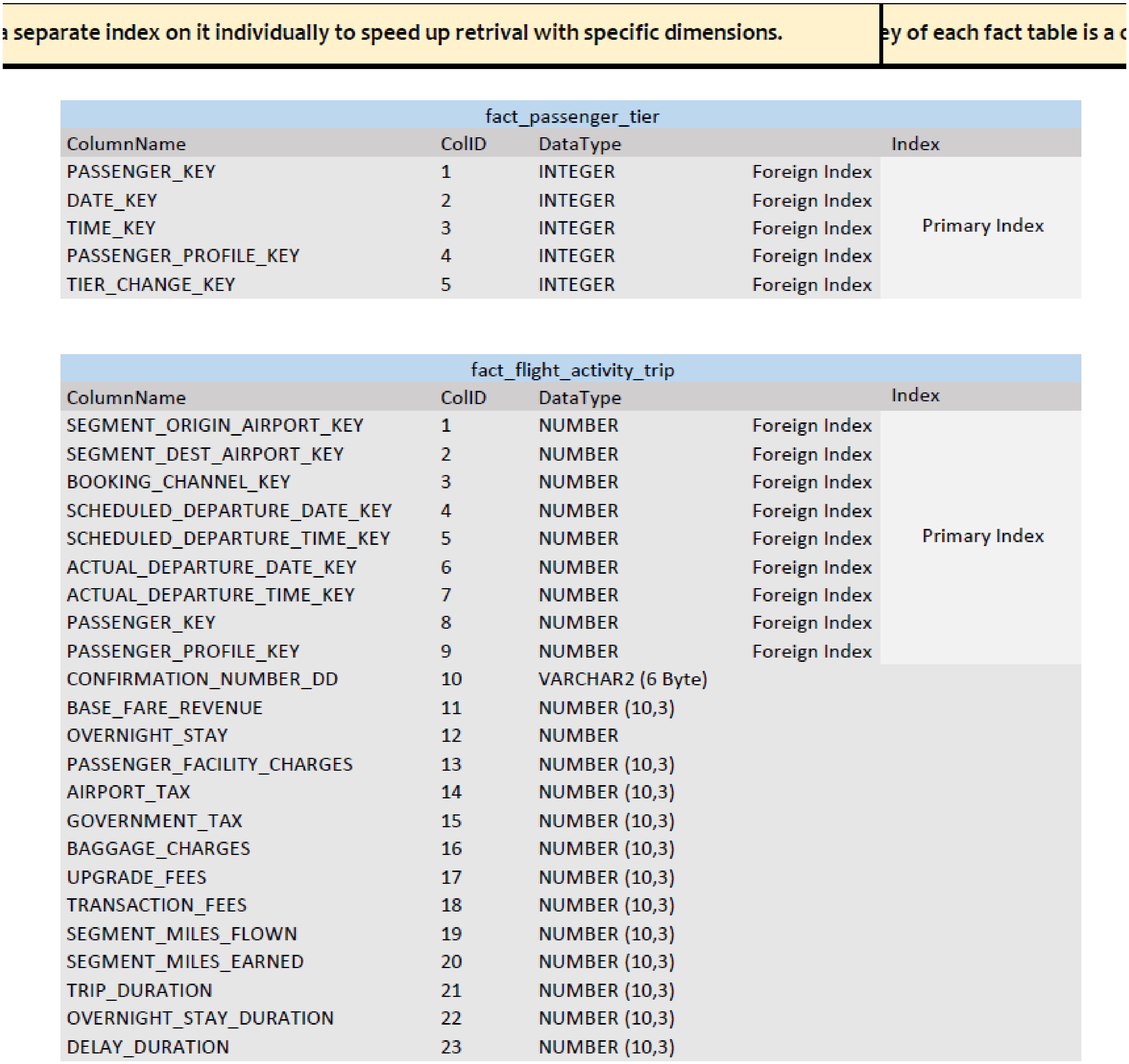
ColumnName	ColID	DataType		Index
PASSENGER_KEY	1	INTEGER	Foreign Index	Primary Index
PASSENGER_PROFILE_KEY	2	INTEGER	Foreign Index	
DATE_KEY	3	INTEGER	Foreign Index	
TIME_KEY	4	INTEGER	Foreign Index	
POINTS_TRANSACTION_PROFILE_KEY	5	INTEGER	Foreign Index	
POINTS_TOTAL	6	NUMBER (10,2)		

fact_reservations

ColumnName	ColID	DataType		Index
SCHEDULED_DEPARTURE_DATE_KEY	1	INTEGER	Foreign Index	Primary Index
SCHEDULED_DEPARTURE_TIME_KEY	2	INTEGER	Foreign Index	
PASSENGER_KEY	3	INTEGER	Foreign Index	
PASSENGER_PROFILE_KEY	4	INTEGER	Foreign Index	
ORIGIN_AIRPORT_KEY	5	INTEGER	Foreign Index	
DESTINATION_AIRPORT_KEY	6	INTEGER	Foreign Index	
AIRCRAFT_KEY	7	INTEGER	Foreign Index	
CLASS_OF_SERVICE_KEY	8	INTEGER	Foreign Index	
FARE_BASIS_KEY	9	INTEGER	Foreign Index	
BOOKING_CHANNEL_KEY	10	INTEGER	Foreign Index	
CITY_PAIR_ROUTE_KEY	11	INTEGER	Foreign Index	
CONFIRMATION_NUMBER	12	VARCHAR2 (255 Byte)		
SEGMENT_SEQUENCE_NUMBER	13	INTEGER		
FLIGHT_NUMBER	14	VARCHAR2 (255 Byte)		
BASE_FARE_REVENUE	15	NUMBER (18,2)		
PASSENGER_FACILITY_CHARGES	16	NUMBER (18,2)		
AIRPORT_TAX	17	NUMBER (18,2)		
GOVERNMENT_TAX	18	NUMBER (18,2)		
TRANSACTION_FEES	19	NUMBER (18,2)		

fact_flight_revenue

ColumnName	ColID	DataType		Index
SCHEDULED_DEPARTURE_DATE_KEY	1	INTEGER	Foreign Index	Primary Index
SCHEDULED_DEPARTURE_TIME_KEY	2	INTEGER	Foreign Index	
ORIGIN_AIRPORT_KEY	3	INTEGER	Foreign Index	
DESTINATION_AIRPORT_KEY	4	INTEGER	Foreign Index	
FLIGHT_MILESTONE_KEY	5	INTEGER	Foreign Index	
AIRCRAFT_KEY	6	INTEGER	Foreign Index	



a separate index on it individually to speed up retrival with specific dimensions. ey of each fact table is a c

fact_passenger_tier				
ColumnName	ColID	DataType		Index
PASSENGER_KEY	1	INTEGER	Foreign Index	Primary Index
DATE_KEY	2	INTEGER	Foreign Index	
TIME_KEY	3	INTEGER	Foreign Index	
PASSENGER_PROFILE_KEY	4	INTEGER	Foreign Index	
TIER_CHANGE_KEY	5	INTEGER	Foreign Index	

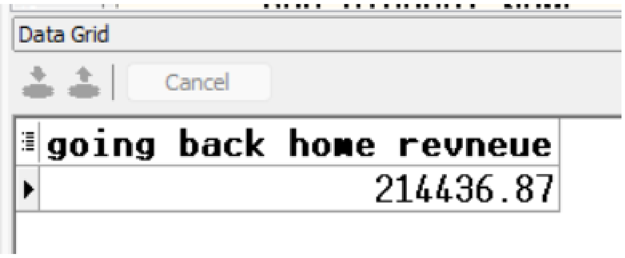
fact_flight_activity_trip				
ColumnName	ColID	DataType		Index
SEGMENT_ORIGIN_AIRPORT_KEY	1	NUMBER	Foreign Index	Primary Index
SEGMENT_DEST_AIRPORT_KEY	2	NUMBER	Foreign Index	
BOOKING_CHANNEL_KEY	3	NUMBER	Foreign Index	
SCHEDULED_DEPARTURE_DATE_KEY	4	NUMBER	Foreign Index	
SCHEDULED_DEPARTURE_TIME_KEY	5	NUMBER	Foreign Index	
ACTUAL_DEPARTURE_DATE_KEY	6	NUMBER	Foreign Index	
ACTUAL_DEPARTURE_TIME_KEY	7	NUMBER	Foreign Index	
PASSENGER_KEY	8	NUMBER	Foreign Index	
PASSENGER_PROFILE_KEY	9	NUMBER	Foreign Index	
CONFIRMATION_NUMBER_DD	10	VARCHAR2 (6 Byte)		
BASE_FARE_REVENUE	11	NUMBER (10,3)		
OVERNIGHT_STAY	12	NUMBER		
PASSENGER_FACILITY_CHARGES	13	NUMBER (10,3)		
AIRPORT_TAX	14	NUMBER (10,3)		
GOVERNMENT_TAX	15	NUMBER (10,3)		
BAGGAGE_CHARGES	16	NUMBER (10,3)		
UPGRADE_FEES	17	NUMBER (10,3)		
TRANSACTION_FEES	18	NUMBER (10,3)		
SEGMENT_MILES_FLOWN	19	NUMBER (10,3)		
SEGMENT_MILES_EARNED	20	NUMBER (10,3)		
TRIP_DURATION	21	NUMBER (10,3)		
OVERNIGHT_STAY_DURATION	22	NUMBER (10,3)		
DELAY_DURATION	23	NUMBER (10,3)		

Database Design

The Oracle implementation followed a star schema design, with fact tables at the center and dimension tables providing descriptive attributes. This structure enables efficient analytical processing and supports complex queries for business insights.

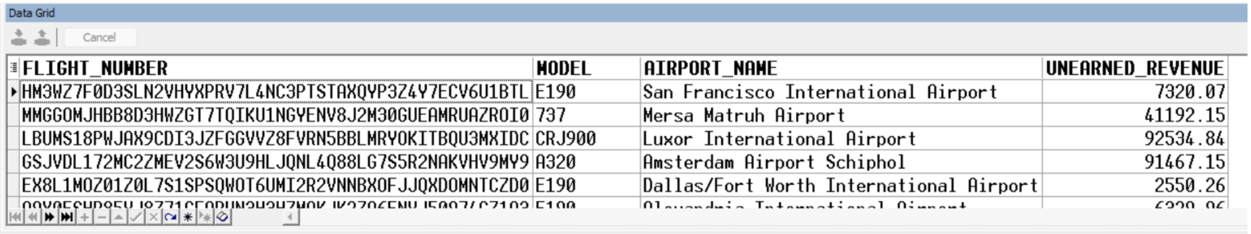
Here's a summary of the types of queries we executed to obtain insights related to flight operations, revenue generation, customer behavior, and frequent flyer activity.

```
-- calculates all revenue (earned and unearned) from people taking flights back home

SELECT SUM (F.BASE_FARE_REVENUE) AS "going back home revneue"
  FROM fact_reservations f
       JOIN dim_passenger dp
         ON F.PASSENGER_KEY = DP.passenger_KEY
       JOIN dim_airport da
         ON F.DESTINATION_AIRPORT_KEY = DA.AIRPORT_KEY
       JOIN dim_city_pair_route dc
         ON F.CITY_PAIR_ROUTE_KEY = DC.CITY_PAIR_ROUTE_KEY
 WHERE DP.country = DA.country AND DC.DOM_INTL_INDICATOR = 'International';
```



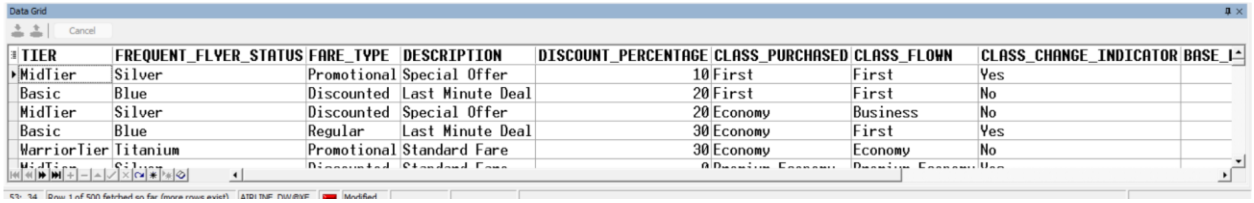
```
going back home revneue
              214436.87
```

```
-- Fully Booked Flights 30 days before taking off, the airplane model for the flight, the airport it's heading to, and the
revenue it generated

SELECT FFR.FLIGHT_NUMBER,
       DA.MODEL,
       DAP.AIRPORT_NAME,
       FFR.UNEARNED_REVENUE
  FROM fact_flight_revenue ffr
       JOIN dim_flight_milestone dfm
         ON FFR.FLIGHT_MILESTONE_KEY = DFM.FLIGHT_MILESTONE_KEY
       JOIN dim_aircraft da
         ON FFR.AIRCRAFT_KEY = DA.AIRCRAFT_KEY
       JOIN dim_airport dap
         ON FFR.DESTINATION_AIRPORT_KEY = DAP.AIRPORT_KEY
 WHERE DFM.FULLY_BOOKED_INDICATOR = 'Fully Booked'
       AND DFM.DAYS_BEFORE_FLIGHT = 30;
```



FLIGHT_NUMBER	MODEL	AIRPORT_NAME	UNEARNED_REVENUE
HM3WZ7F0D3SLN2VHYXPRV7L4NC3PTSTAXQYP3Z4Y7ECV6U1BTL	E190	San Francisco International Airport	7320.07
MMGGOMJHBB8D3HWZGT7TQIKU1NGYENV8J2M30GUEAMRUAZROI0	737	Mersa Matruh Airport	41192.15
LBUMS18PWJAX9CDI3JZFGGVVZ8FVRN5BBLMRYOKITBQU3MXIDC	CRJ900	Luxor International Airport	92534.84
GSJVDL172MC2ZMEV2S6W3U9HLJQNL4Q88LG7S5R2NAKVHV9MY9	A320	Amsterdam Airport Schiphol	91467.15
EX8L1MOZ01Z0L7S1SPSQWOT6UMI2R2VNNBXOFJJQXDOMNTCZD0	E190	Dallas/Fort Worth International Airport	2550.26
[illegible]	E190	Alexandria International Airport	[illegible]

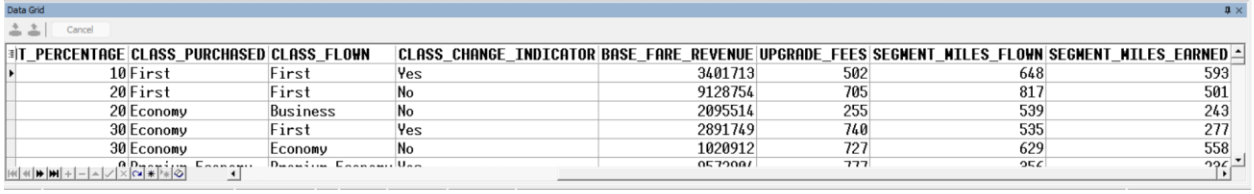
```
-- displays frequent flyer tier and status, base fare info, class of service flown info, and the measures related to one flight segment
-- filtered by home airport

SELECT pp.TIER,
       pp.FREQUENT_FLYER_STATUS,
       fb.FARE_TYPE,
       fb.DESCRIPTION,
       fb.DISCOUNT_PERCENTAGE,
       cas.CLASS_PURCHASED,
       cas.CLASS_FLOWN,
       cas.CLASS_CHANGE_INDICATOR,
       fasa.BASE_FARE_REVENUE,
       fasa.UPGRADE_FEES,
       fasa.SEGMENT_MILES_FLOWN,
       fasa.SEGMENT_MILES_EARNED
 FROM FACT_FLIGHT_ACTIVITY_SEGMENT fasa
       JOIN DIM_PASSENGER_PROFILE pp
         ON fasa.PASSENGER_PROFILE_KEY = pp.PROFILE_KEY
       JOIN DIM_FARE_BASIS fb
         ON fasa.FARE_BASIS_KEY = fb.FARE_BASIS_KEY
       JOIN DIM_CLASS_OF_SERVICE cas
         ON fasa.CLASS_OF_SERVICE_FLOWN_KEY = cas.CLASS_OF_SERVICE_KEY
 WHERE PP.HOME_AIRPORT = 'Texas';
```



TIER	FREQUENT_FLYER_STATUS	FARE_TYPE	DESCRIPTION	DISCOUNT_PERCENTAGE	CLASS_PURCHASED	CLASS_FLOWN	CLASS_CHANGE_INDICATOR	BASE_I
MidTier	Silver	Promotional	Special Offer	10	First	First	Yes	
Basic	Blue	Discounted	Last Minute Deal	20	First	First	No	
MidTier	Silver	Discounted	Special Offer	20	Economy	Business	No	
Basic	Blue	Regular	Last Minute Deal	30	Economy	First	Yes	
WarriorTier	Titanium	Promotional	Standard Fare	30	Economy	Economy	No	
MidTier	Silver	Discounted	Standard Fare	0	Premium Economy	Premium Economy	Yes	

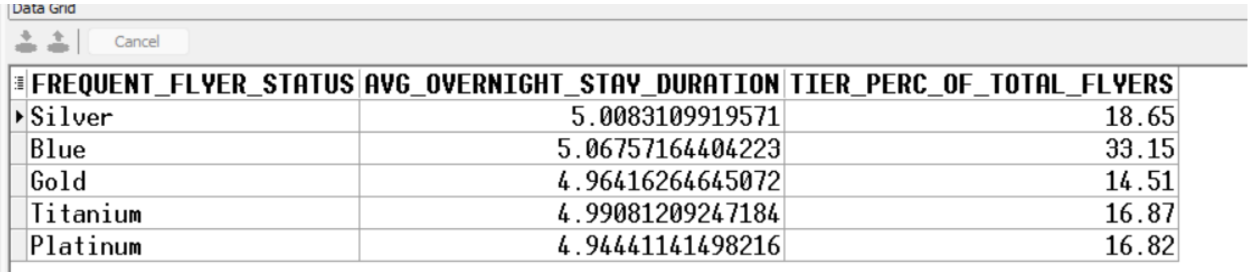
53: 34 | Row 1 of 500 fetched so far (more rows exist) | AIRLINE_DW@XE | Modified

-- calculates the average duration of overnight stays for passengers grouped by their frequent flyer status, and



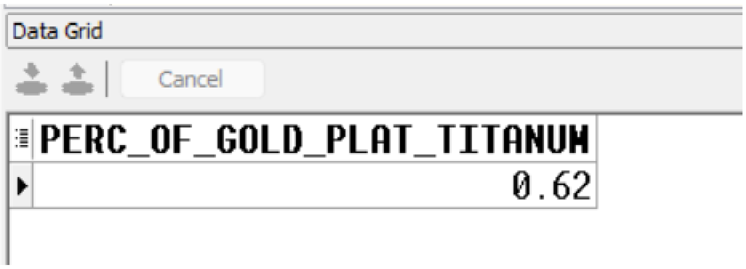
T_PERCENTAGE	CLASS_PURCHASED	CLASS_FLOWN	CLASS_CHANGE_INDICATOR	BASE_FARE_REVENUE	UPGRADE_FEES	SEGMENT_MILES_FLOWN	SEGMENT_MILES_EARNED
10	First	First	Yes	3401713	502	648	593
20	First	First	No	9128754	705	817	501
20	Economy	Business	No	2095514	255	539	243
30	Economy	First	Yes	2891749	740	535	277
30	Economy	Economy	No	1020912	727	629	558
0	Premium Economy	Premium Economy	Yes	[illegible]	777	[illegible]	[illegible]

```sql
-- calculates the average duration of overnight stays for passengers grouped by their frequent flyer status, and
-- displays what percentage of all flyers customers in this tier account for


 SELECT dp.frequent_flyer_status,
        AVG (faf.Overnight_Stay_Duration) AS avg_overnight_stay_duration,
        ROUND ( (COUNT (1) / SUM (COUNT (1)) OVER ()) * 100, 2)
          AS tier_perc_of_total_flyers
   FROM    Fact_flight_activity_trip faf
        JOIN
          dim_passenger_profile dp
        ON faf.Passenger_Profile_Key = dp.profile_key
GROUP BY dp.frequent_flyer_status;
```



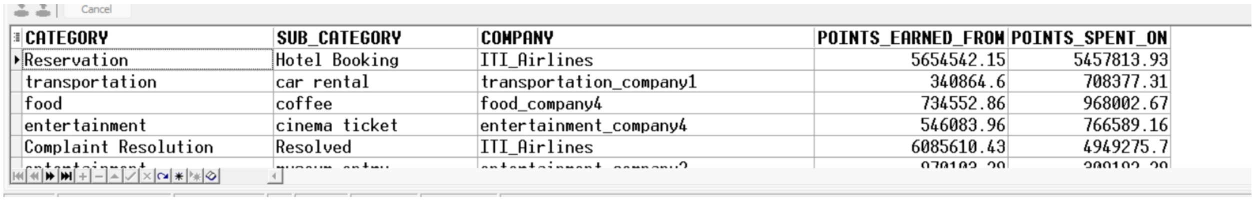
FREQUENT_FLYER_STATUS	AVG_OVERNIGHT_STAY_DURATION	TIER_PERC_OF_TOTAL_FLYERS
Silver	5.0083109919571	18.65
Blue	5.06757164404223	33.15
Gold	4.96416264645072	14.51
Titanium	4.99081209247184	16.87
Platinum	4.94441141498216	16.82

```sql
--what proportion of frequent flyers that upgrade have gold, platinum or titanium status.


SELECT ROUND (
          SUM (
             CASE
                WHEN DCP.FREQUENT_FLYER_STATUS IN
                        ('Gold', 'Platinum', 'Titanium')
                THEN
                   1
                ELSE
                   0
             END)
          / SUM (1),
          2)
          AS perc_of_gold_plat_titanum
 FROM    fact_passenger_tier fpt
       JOIN
          dim_tier_change_profile dcp
       ON DCP.TIER_CHANGE_KEY = FPT.TIER_CHANGE_KEY
 WHERE DCP.IS_UPGRADE = 'Y';
```



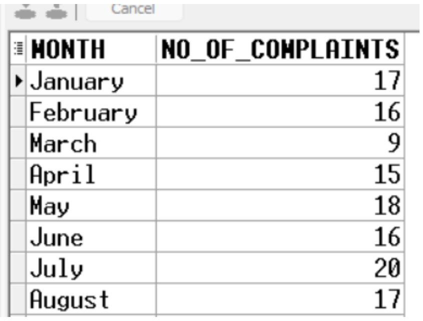
PERC_OF_GOLD_PLAT_TITANUM
0.62

```
-- how they earn and redeem their frequent flyer points

  SELECT DPT.CATEGORY,
         DPT.SUB_CATEGORY,
         DPT.COMPANY,
         SUM (
            CASE
               WHEN dpt.transaction_type = 'earn' THEN points_total
               ELSE 0
            END)
            points_earned_from,
         SUM (
            CASE
               WHEN dpt.transaction_type = 'redeem' THEN points_total
               ELSE 0
            END)
            points_spent_on
    FROM    fact_points_transactions fpt
         JOIN
            dim_points_transaction_profile dpt
         ON FPT.POINTS_TRANSACTION_PROFILE_KEY = DPT.POINTS_ACTIVITY_KEY
GROUP BY DPT.CATEGORY, DPT.SUB_CATEGORY, DPT.COMPANY;
```



CATEGORY	SUB_CATEGORY	COMPANY	POINTS_EARNED_FROM	POINTS_SPENT_ON
Reservation	Hotel Booking	ITI_Airlines	5654542.15	5457813.93
transportation	car rental	transportation_company1	340864.6	708377.31
food	coffee	food_company4	734552.86	968002.67
entertainment	cinema ticket	entertainment_company4	546083.96	766589.16
Complaint Resolution	Resolved	ITI_Airlines	6085610.43	4949275.7

```
--Tracking passenger complaints for each month in 2019

  SELECT dd.month,
         SUM (CASE WHEN DIT.INTERACTIONTYPE = 'Complaint' THEN 1 ELSE 0 END)
            AS no_of_complaints
    FROM fact_customer_interaction fci
         JOIN dim_date dd
            ON FCI.DATE_KEY = dd.date_key
         JOIN DIM_INTERACTION_TYPE dit
            ON DIT.INTERACTIONTYPE_KEY = FCI.INTERACTIONTYPE_KEY
   WHERE dd.year = 2019
GROUP BY dd.month, EXTRACT (MONTH FROM DD.FULL_DATE)
ORDER BY EXTRACT (MONTH FROM DD.FULL_DATE)
```



MONTH	NO_OF_COMPLAINTS
January	17
February	16
March	9
April	15
May	18
June	16
July	20
August	17