* The second delivery of the project focuses on the following:
  + Loyalty Program.
  + Performance Enhancement.
  + Profit & Cost Analysis.
* We will follow a reversed approach for time’s sake, modeling each business process separately.
* We may find the time to complete the second deliverable before the deadline, if not we will deliver what we have achieved so far!

**Step 1**

Step one will be modeling the loyalty program business process.

**Defining Business Process**

* Loyalty Program works as the following:
  + We make agreements with companies called partners.
  + When a frequent flyer (A customer holding a loyalty program card) deal with any of the partners he gets awarded some points called frequent miles.
  + The passenger can collect points also by dealing directly with us.
  + The passenger can redeem these points as a discount on a purchased ticket, as an upgrade on a flight, as extra luggage, or any other service we provide.
  + Not only that, but they can also redeem points into free drinks, free food, hotel rooms, car rental services, or any other service provided by any of our partners.
  + So, it is a 2-way agreement between us and the partners.

**Defining Granularity**

* We will work on the grain of each transaction made by a frequent flyer using the service of any of the agreement’s parties.

**Defining Dimensions**

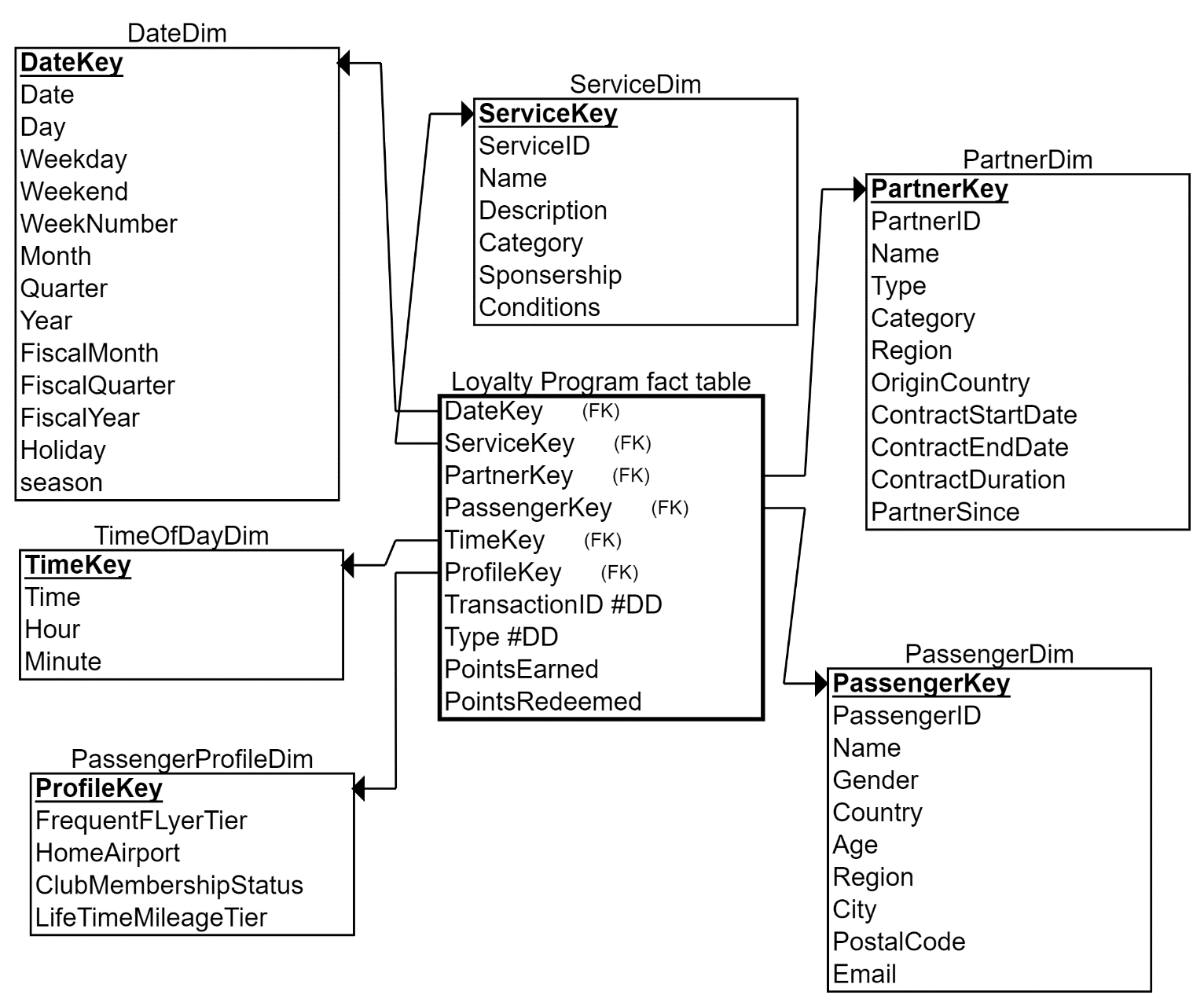
1. **TransactionID:** Degenerate Dimension that represents a unique identifier for each transaction.
2. **DateKey:** Foreign key linking to the **DateDim** table for transaction date.
3. **TimeKey:** Foreign key linking to the **TimeOfDayDim** table for transaction time.
4. **PassengerKey:** Foreign key linking to the **PassengerDim** table.
5. **PassengerProfileKey:** Foreign key linking to the **PassengerProfileDim** table.
6. **ServiceKey:** Foreign key linking to a new **ServiceDim** table.
7. **PartnerKey:** Foreign key linking to a new **PartnerDim** table.
8. **Type:** Type of transaction (e.g., Earn or Redeem).

New Dimension Description:

* ServiceDim:
* A new dimension that describes each service provided by our company and its partners, including:
  + ServiceKey
  + ServiceID
  + Name
  + Description
  + Category (e.g., In-flight service, baggage, booking upgrades, etc.)
  + Sponsorship (fully sponsored, partially sponsored, only 1st tier, etc.)
  + Conditions (any additional conditions that must be applied in order to earn/redeem points from this service)
* PartnerDim:
* A new dimension for partner companies, it holds the following attributes:
  + PartnerKey
  + PartnerID
  + Name
  + Type (e.g., Airline, Rental Car, Hotel, Retail, etc.)
  + Category (e.g., Partner, sponsor, Subsidiary, etc.)
  + Region
  + OriginCountry
  + ContractStartDate
  + ContractEndDate
  + ContractDuration
  + PartnerSince

**Defining Measurements**

1. **PointsEarned:** Number of points earned in the transaction.
2. **PointsRedeemed:** Foreign key linking to the **DateDim** table for transaction date.

Now as we have integrated all flyer miles transactions together, we can fully analyze how passengers earn and redeem frequent flyer miles.

**Step 2**

Step two will be creating aggregated fact tables on top existing ones.

* It is always a tradeoff between disk usage and performance, but in most cases, we will prioritize time/performance over storage in DWH.
* So, in this step we will create pre-aggregated fact table to get red of aggregation on run time.

**ReservationsAgg Fact Table:**

* We will create a monthly aggregated fact table on top of reservations fact table.
* The book suggests creating a 90-day aggregated fact table to capture reservations, but I don’t see why 90 days, so we will make it a monthly grained fact table.

**Defining Business Process:**

* Flight Reservations.

**Defining Granularity:**

* Monthly.

**Defining Dimension:**

1. **MonthKey:** Foreign key linking to a new **DateDim** table (points to a row indicating start of month e.g. 1/1/2024).
2. **SrcAirportKey:** Foreign key linking to a new **Airport** table.
3. **DstAirportKey**: Foreign key linking to a new **Airport** table.
4. **ChannelKey**: Foreign key linking to a new **ChannelDim** table.
5. **ClassKey**: Foreign key linking to a new **ClassDim** table.
6. **FareBasisKey**: Foreign key linking to a new **FareBasisDim** table.

* Using this table, we could get our monthly sales for a specific airport for example in a much faster manner.

**Defining Measurements:**

* **TotalReservations:** The total number of reservations made in the month.
* **TotalFareRevenue**: Sum of TotalFare from the Reservations fact table.
* **TotalAncillaryRevenue**: Sum of AncillaryRevenue.
* **TotalGovernmentalTaxes**: Sum of GovernmentalTaxes.
* **TotalAirportTaxes**: Sum of AirportTaxes.
* **TotalSecurityFees**: Sum of SecurityFees.
* **TotalBaggageFees**: Sum of BaggageFees.
* **TotalUpgradeFees**: Sum of UpgradeFees.
* **TotalOtherFees**: Sum of OtherFees.
* **AverageDiscount**: Average of Discount applied.
* **TotalSales**: Sum of all charges – discount.

A diagram of a data flow

Description automatically generated**Model**

**FlightsAgg Fact Table:**

* We will create a monthly aggregated fact table on top of flight activity fact table.
* We will create a monthly fact table to capture flights organized by the company over each month.

**Defining Business Process:**

* Flights.

**Defining Granularity:**

* Monthly.

**Defining Dimension:**

1. **MonthKey:** Foreign key linking to a new **DateDim** table (points to a row indicating start of month e.g. 1/1/2024).
2. **SrcAirportKey:** Foreign key linking to a new **AirportFim** table.
3. **DstAirportKey:** Foreign key linking to a new **AirportFim** table.
4. **AircraftKey**: Foreign key linking to a new **AircraftDim** table.
5. **CaptainKey**: Foreign key linking to a new **EmployeeDim** table.

* Using this table, we could get the number of monthly flights for each airport in a much faster manner.

**Defining Measurements:**

* **TotalFlights:** The total number of flights during the month.
* **TotalPassengers:** Sum of PassengersCount.
* **TotalEmptySeats:** Sum of EmptySeats.
* **TotalFuelConsumption:** Sum of FuelConsumption.
* **AverageCrewCount:** Average of CrewCount per flight.
* **TotalBookedSeats:** Sum of BookedSeats.

**Model**

A diagram of a data flow

Description automatically generated

**Step 3**

Step three will be Analyzing profit.

Problem statement:

* We don’t provide a product that has a selling cost, we provide a service and the only cost we have is operational cost like:
  + Salaries, commissions, fuel, marketing, etc.
  + So, we couldn’t just put the cost or the profit as a measurement in the reservations’ fact (for finance team), because we can’t calculate the cost of a ticket!
  + In the previous deliverable we calculated earned revenue, but here we won’t to calculate actual profit.
* Maintenance itself is a separate business process, and so is HR (for salaries), so is marketing campaigns, and so on.
* With the given timeframe we can’t create a model to cover all these business process in order to have control over all cost and revenue sources.
* Instead, we will assume that stockholders are only interested in in analyzing profit (cost and revenue) rather than deeply analyzing the underlying business process for this cost entry.

**Expenses Fact Table:**

* In this fact table we will capture all company’s expenses.
* Then we will integrate this table with monthly reservations fact table to create a 3-month-aggregated-profit-analysis fact table!

**Defining Business Process:**

* This business process captures each expense entry issued from us.
* We will capture money paid for maintenance, flight operations, salaries, marketing campaigns, commissions, etc.

**Defining Granularity:**

* Each expense Entry will be represented in a single row.

**Defining Dimension:**

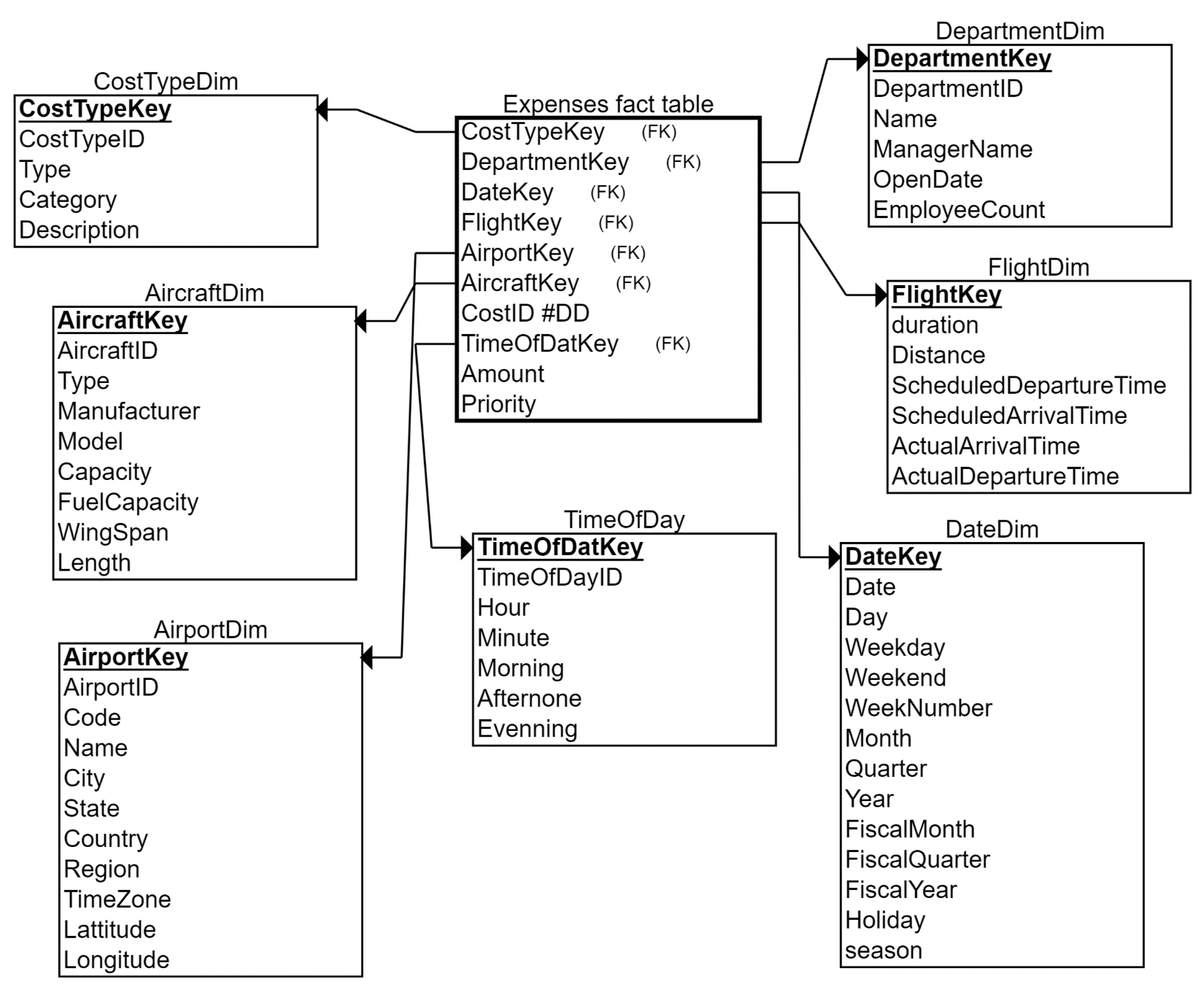
1. **CostID:** Degenerate Dimension that represents a unique identifier for each cost entry.
2. **DateKey:** Foreign key Linking to **DateDim** to record when the transaction accured.
3. **TimeKey:** Foreign key Linking to **TimeOfDayDim** to record when the transaction accured.
4. **CostTypeKey:** Foreign key Linking to **CostTypeDim.**
5. **FlightKey** (optional):If costs can be directly associated with specific flights, link to **FlightDim**.
6. **DepartmentKey** (optional): Link to a **DepartmentDim**; if costs need to be tracked by department (e.g., maintenance, crew, ground services).
7. **AirportDim** (optional):If costs can be directly associated with specific airports, link to **AirportDim**.
8. **AircraftDim** (optional):If costs can be directly associated with specific aircraft, link to **AircraftDim**.

New Dimension Description:

* CostTypeDim:
* A new dimension that describes the type of cost entry.
* Categorizes each type of cost.
* e.g., maintenance, salaries, commissions, marketing.
  + CostTypeKey
  + CostTypeID
  + Type (e.g., Maintenance, Salaries, Commissions, Marketing, etc.)
  + Category (e.g. Operational, HR, Marketing, etc.)
  + Description
* DepartmentDim:
* A new dimension describing the department associated with this cost Entry.
  + DepartmentKey
  + DepartmentID
  + Name
  + ManagerName
  + OpenDate
  + EmployeesCount

**Defining Measurements:**

* **Amount:** The monetary amount of the cost.
* **Priority:** How much this cost entry is crucial as a scale from 1 to 10.

**Model**

* Now we have modeled revenues in the ReservationsAgg Fact table and modeled expenses (cost) in Expenses Fact Table.
* But we need to integrate both fact table to analyze profit as:
  + Profit=Total Revenue−Total Costs
* We will create a monthly aggregated fact table to capture the company’s profit.
* This fact table will be created on top of 2 fact tables (ReservationsAgg and expenses)

**Profit Fact Table:**

* We will create a monthly aggregated profit fact table integrating both Expenses fact table and Reservations aggregated fact table.
* This fact table will combine monthly revenues and monthly expenses in order to analyze monthly profit.

**Defining Business Process:**

* Integrating Expenses Business Process with Revenue Business Process to calculate the company’s profit.

**Defining Granularity:**

* Monthly.

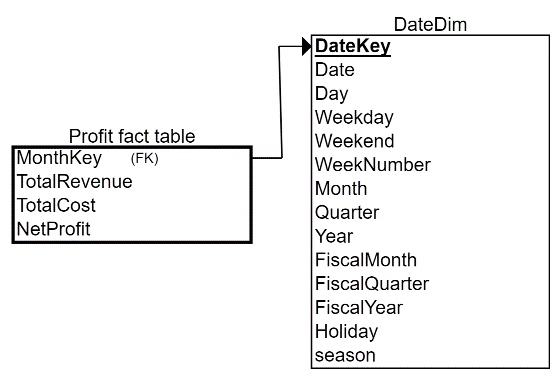
**Defining Dimension:**

1. **MonthKey:** Foreign key linking to a new **DateDim** table (points to a row indicating start of month e.g. 1/1/2024).

* We will aggregate Expenses fact table on monthly basis, then join the aggregated table with Reservations monthly aggregated table using month to form the monthly aggregated profit table.

**Defining Measurements:**

* **TotalRevenue:** The total monthly revenue
  + **TotalSales** attributable in ReservationsAgg monthly aggregated fact.
* **TotalCost:** Sum of all cost-related figures for the month from the Expenses fact table.
* **NetProfit:** TotalRevenue - TotalCost.

**Model**