My core challenge is managing aviation transport operations with a focus on people and assets, flexible scheduling, accurate execution tracking, and billing as required by clients based on any number of possible base variables - flight hours, acres covered, number of loads, lbs, miles, etc.

Unfortunately this is my best attempt over a few hours but isn't yet all inclusive. You can find some rough drafts at the objects attached here, and some other versions in the chart below: https://www.mermaidchart.com/d/6bce2648-cd2c-44a0-be22-bfd526c590d1

There are some catches in the data set that if we structure things a certain way, it may cause other problems.
I'm looking for two things:

1. Your recommendation in how to solve this problem set
2. Your proposal for how to create the solution for my problem in salesforce

We try to put our operations into some major phases that occur in two worlds:

Overall we do this:

1. Plan
2. Operate
3. Report

Our departments are:

1. Sales
2. Operations
3. Maintenance

Internal Stages to our ops:

1. Plan and communicate the plan and price
    a. Take note of previous safety implementations, other notes on the mission type, location, client, job, etc
2. Prepare ourselves and the client for the job
3. Execute the job brilliantly
4. Tally up the totals, debrief, and record possible improvements to the process
    a. Scheduler reviews total amounts of flight time etc, approves it and sends to client.
5. Billing
    a. Tally up the total to invoice, and send it to AR
    b. AR takes the billing and ensure it matches to contract and funding line
    c. AR sends out the invoice to the client
6. Improvement
    a. Safety report recommended solutions are implemented, snags and hiccups are noted where expected
    b. Notes for mission types and mission templates are updated

c. notes for locations are updated
d. Pictures of landing zones and locations used update the location database with new information
e. Location is updated with the last landing time

Our client facing stages are:

1. Request
2. Confirm
3. Execute
4. Billing / Reporting

Plan - The planning phase brings in client requests, allocates safety and risk management tasking, assets, people onto a set of calendar days, and then passes it along to the team for execution. The Planning phase begins with sales, then is passed to operations, who validates the plan and confirms allocation to the asset with maintenance based on the projected flight hours. The people look at the calendar, understand the upcoming tasking, and begin to individually prepare to execute the coming jobs and support them.

Operate - The operate phase begins when the plan is confirmed and entered in the system. The flight is flown, data about the operation is entered into the appropriate data fields to capture locations, routes, flight time, standby time, loads taken, passengers flown, and other data points.

Report - Reporting is the process of sending confirmed data out to our clients, this data should validate our value to our client. The reporting also comes internally, where our team is reporting what does or does not work about a leg/job/work order and reports it, these reports are to be entered into the mission type to refine the leg/job/work order/location depending on the scope so the next time we execute we know to implement that change.

Request - Client requests a flight, or requests a proposal/estimate/quote for a flight type, or something else. Our job is to respond to this as accurately and as quickly as possible. Sometimes a request is very open ended, and the client doesn't know what they want.
Confirm - Vertical Worx confirms to the client via their preferred modality such as text that the flight is confirmed and sends the leg(s) assigned to the client and a tentative schedule
Execute - the execution begins with a safety brief, then the flight portion and completion of the leg or the task or the loads
Billing / Reporting - the client gets the report, 5000lbs moved 50 miles over 10 trips / 10 loads of 500 lbs / 5 passengers moved 50 miles. This is the load reporting and is critical to substantiating the bill and securing our position as best in class. The Billing follows and is calculated based on the rate card and the actual travel taking place.

Maintenance needs to see the planned amount of flight time that we expect to see happen so they can plan for adequate maintenance to ensure each aircraft can perform as required for the flight event.

The client should be able to request a flight, as granular as they'd like to request it to include individual legs and passengers if they choose, or they can simply enter a description of the operation they want and the schedular will take that and create the flight legs they would like.

The client, if they request the flight legs or the operation on the calendar, will not see it confirmed until the scheduler confirms each of the flight legs on the calendar, considering the total flight time, total work time, and fuel load requirements on site.

The confirmation then allows the client to see their flights on the calendar. If the client is part of a conglomerate of associated companies, they all get to see eachother's flights if they should request that.

All clients get to see where assets are planned to be (confirmed) in the future on a map, in order to piggy back operations on an existing deployment.

There are many accounts, or clients that we work with. Each account has many individual contacts. Accounts will often submit a flight request, or a request for quote. A flight request is a simple request that outlines what someone wants to do, by saying where they or their cargo want to go.
Accounts can have many flight requests
Accounts can have many contracts
Accounts can have many invoices
Accounts can have many Jobs
Accounts can have many legs

Either a flight request or a request for quote can be added to the schedule if they are either tied to a Funding_Line. A flight request is driven by the customer to make flight legs.

The Contract defines a binding agreement with a client that drives the funding. Non-Contract clients agree to a standard credit card authorization for funding. The contract itself is not the funding vessel, it is the "Funding_Line" tied to the contract that defines the funding. Contracts have max obligated amount, rate, and other items to track. Contracts can have many funding lines.

A Funding line, if not a prepaid or a credit card, will be tied to a contract. A Funding Line is intended to be a reference to tie invoicing and billing to the client's agreed upon rates. The

funding line tracks the type of funding, the related contract, the limit of obligated funds, the associated RateCards are linked to the funding line. Many funding lines can link to one contract. Funding Line links to many jobs, rateCardLine, Charges, Invoices.

RateCard is a catalogue of Rate structures.

RateCard Version is a version of of a rate card, and connects to many rate card lines, many versions connect to just one rate card

RateCardLine is an individual line entry of a rate, such as Flight time hourly = $3000 / hr

Some clients can only go on the schedule if they have an active purchase order to bill against. Only some contacts are approved by the client to create a flight request.

Job is defined as a single scope of work for a single client that can span multiple days, the job carries its own requirements, risk mitigation plans, job planning surveys, etc as needed, and Jobs are bucketed into Mission Types and Operations Types to further guide the planning process

Mission_Template is an object to bring mission types and operation types into a role to define rulesets around planning, and support, etc

The Work Order is a collection of a single day's worth of work meant for a single asset (such as a helicopter) The work order ties together all of the legs.

Legs are individual portions of the flight characterized by a takeoff and a landing event. Legs can have a Manifest (list of people by name with weights)

Legs are the planned portion of the flight that are for purposes of planning and organized information, and final data is entered into the Flight_Execution__c automatically from in-asset trackers, and they link to actual asset transportation events, and links to individual loads. Before closing out a Work Order, each flight execution has to be matched to the account. The account then matches the work order and flight legs for the day to link to the RateCardLines to calculate total billed. Flight Hours are the amount automatically calculated from the in-asset trackers, and Billable flight hours are the actual amount charged to the client in the case that some discount should happen, or a manual re-allocation of flight time should occur.

Load - a load is a weight on and off event that happens inside of a leg, as reported by the in-asset tracker, and measured by the load cell in the aircraft. Many load events link to a single Flight_Execution__c

Work Order Client is a list of clients connected to a work order for a day's work

JobRateLineSnapshot

Charge is billing for a line item - derived from a total of legs inside of a single work order - many charges make an invoice line item,

Invoice is a client billing document

Invoice line is a line on the billing document

Unknowns:

- Standby is defined as the time the aircraft is dedicated to a Job but is not actively flying, the aircraft is standing by. This can be defined by the equation: Standby = (time of return - time of departure) - Flight time. How do we integrate standby in a way that allows us to automatically calculate the standby and attach it to a leg without polluting the data when standby can occur between two clients and potentially be at the behest of either one or the other?

- Sometimes we are waiting on an uncontrollable event to take place in order to sell an operation. This causes us to have "ghost dates" or "possible execution ranges" these need to be on the radar for ops in case the thing we don't control can take place and we can execute the job, but also they can't complete allocate resources until this thing is set. We need to keep these moving as they are critical and often GOOD money, but we also can't just set a calendar date and keep shifting it as that causes issues with clients that have money to put down on the job. What ways can our process integrate these kinds of operations as efficiently as possible? I.E. They need planning still, we need to be ready to do it, but it isn't certain yet.

- How do I have a very flexible collection of individual tasks under a job that may be spread over multiple days in a very flexible way so that it isn't painful or repetitive to make a number of things that are individual actions for a single job over a number of days without making a painful process somewhere else?

- How do we leverage the stock components of Sales Force Field Service as best as possible without getting pigeon holed into the way they present data?

- How can we dynamically or intuitively assign pilots and assets to missions jobs and legs, and how do we mix and match legs onto different aircraft when we have aircraft get broken?

- How do we build this up with maximum flexibility after the job has been planned and grouped together to ungroup it and move things as needed to make it work?
- How do we capture and report anticipated flight times effectively to the maintenance
- How do we make it easy to move passengers on manifests on flight legs to other flight legs and manifests?
- Is it possible to seamlessly bill individual passengers in the future with this data structure?

- How can a maintenance module be built onto the back end

REPORTS:
Planned flight hours vs actual
Planned revenue vs actual
Cancellation rate per client or location or asset
Planned work order related safety notes for each asset / client / location / mission type / etc
Planned flight hours