**Flight Spanners Data Requirements**

* Holiday has one holiday type.
* Approval has a specific aircraft type and contains an approval detail.
* Spanner could only registered in one group.
* Spanner has one qualification.
* Spanner approved on one or more approvals of specific aircraft types and in that type the spanner may approved on the aircraft model only, on the engine model only, or on both.
* Spanner works at one department.
* Spanner could has one or more in-active period.
* Organizer could organizes one or more group.
* Organizer is responsible to enter the flight record for his group.
* Flight record should contain the following: - Flight Company, Eligibility type, Destination type, the approval for the spanner if the flight is distributed, the flight aircraft type, the flight data, and the organizer who enter the record.
* Flight data contains one origin and one destination airport, and the Distance type.

**Flight Spanners Entities and their attributes**

There are 18 entities in the Flight Spanners project as follows:

1. **HolidayType** contains the HolidayTypeName, & HolidayTypeConstant (Weekly with constant=1, Monthly with constant=1, Yearly with constant=1).
2. **Holiday** contains the HolidayName, HolidayDay, HolidayMonth, HolidayYear, & IsHolidayRepeated.

(Labor Day which its type is yearly and Day=25, Month=6, year=2017, IsRepeated=true, as each year at the same day and month)

(Feast of el Fitr Day 1 which its type is yearly and Day=1, Month=5, year=null

, IsRepeated=false, as each year has a different date)

(Friday which its type is Weekly with Day=null, Month=null, year=null, IsRepeated=true, as it is repeated each week)

(Fifteen which type is Monthly with Day=15, Month=null, year=null, IsRepeated=true, as it is repeated each half of each month).

1. **AircraftType** has AircraftTypeId, AicraftTypeConstant, AircraftModel & EngineModel. Each AircraftModel should have one or more EngineType.
2. **ApprovalDetail** has ApprovalRating (Airframe, Engine, Airframe & Engine, Electric, Avionics, or Multi-X), ApprovalCategory (B1, B2) and ApprovalConstant.
3. **Approval** contains each spanner with his approved AircraftType and ApprovalDetail.
4. **Qualification** has QualificationName (Engineer, or Technician), QualificationDegree (Associate, Bachelor, Master, or Doctoral), QualificationMajor (Mechanical, or Electrical), QualificationConstant.
5. **Department** has a unique DepartmentName (Hanger8000, Hanger7000, Engineering ...etc.), & DepartmentConstant.
6. **InActivePeriod** the period that each spanner is not available to be selected to flights it has InActiveFromDate, InActiveToDate, the period that not active derived from InActiveFromDate & InActiveToDate.
7. **Spanner** has unique LicenceNo, unique Code, Name, Gender, SpannerBirthday, SpannerHireDate, Mobile (2 numbers with the second optional), Email, Password, IsSpannerViewGroupData (true if his organizer give him permission to view other spanners data of his group), IsSpannerHasCar (true if the spanner has a private car), & Balance (which is derived from spanner and other entities data).
8. **Group** has GroupName (which is the group name that the spanner belongs), RecordStarting (The starting date of record), GroupConstant, & IsCalculationSectorTime (If true means that the Organizer of the Group decides that the calculation of the balance based on distance is according to the default sector time, and the DistanceType Entity is ignored).
9. **Organizer** has unique Code, Name, Mobile (2 numbers with the second optional), Email, Password, & Occupation.
10. **FlightCompany** has FlightCompayName (the name of the company that own the rights to have the flight), & FlightCompanyConstant.
11. **DestinationType** contains the DestinationTypeName, & DestinationTypeConstant (Single with constant=1, Multiple with constant =0.5). Multiple here means multiple destinations until the flight reaches its final destination.
12. **EligibilityType** contains EligibilityTypeName, & EligibilityTypeConstant (Normal with constant=0, Bonus with constant=0, Apology with constant=1). Apology means that the spanner is apologize from a flight without acceptable excuse. Bonus means that the flight is given as an award and does not added to the spanner balance.
13. **FlightRecord** has unique Id, Date/Time of the flight, Date/Time of the record registration, the flight company, the eligibility type for the record, destination type of the flight, the flight route, the organizer who enter the flight record, the AircraftType of the flight, the IsRecordAutoSelect (true if the selection is not by organizer but by an AI algorithm and false if the selection is manually by the organizer), Approval (null means no selection of spanner for this record yet) & IsRecordSettled (true means the flight record is processed & confirmed to be part of the balance calculation and can be viewed by his group spanners).

So it can be said that unprocessed FlightRecord can be assigned to the spanner manually by the organizer (IsRecordAutoSelect=false & Approval !=null)🡺processed record, automatically in future versions of this program using AI selection algorithm (IsRecordAutoSelect=true & Approval !=null)🡺processed record, or no selection for the spanner yet (IsRecordAutoSelect=false & Approval=null)🡺unprocessed record.

1. **Airport** has a AirportCountry, AirportCity, unique IATACode, unique ICAOCode (Egypt, Cairo, CAI, HECA), & optionally the city may has an alternate name.
2. **DistanceType** contains the DistanceTypeName with the associated DistanceTypeConstant, UpperSectorTime, LowerSectorTime, UpperOperator, LowerOperator. (Short with constant=1, SectorTime<=3 hrs, Long with constant=0.65 & 3<SectorTime<=4 hrs, Extra-Long with constant=0.5 & SectorTime>=4 hrs)
3. **FlightData** contains all the possible flights airports Origin & Destination pairs, SectorTime (the time of one sector of the flight), IsFlightLocal (true the flight is local not international flight), IsFlightCargo (true if the flight is cargo type), FlightBonusCard (number of bonus cards if any given for the spanner at this flight) & the derived DistanceType from the DefaultSectorTime.

**Note**: - The entities 11 constants & 1 Boolean attribute are the tool that help the organizers to equally distribute the flights to his group of spanners to overcome the unfair monetarily issue.

This performed by customizing the calculation of the Balance of each spanner by using these attributes. These constants are: - FlightDataConstant, DistanceTypeConstant, GroupConstant, DepartmentConstant, QualificationConstant, ApprovalConstant, AicraftTypeConstant, FlightCompanyConstant, EligibilityTypeConstant, DestinationTypeConstant, & HolidayTypeConstant.

The Boolean attribute is IsCalculationSectorTime on the Group entity

**Flight Spanners Relational Schema**

There are 20 tables in the Flight Spanners project similar to Entities except the following:

* The Entities are 18 whereas the Tables are 20 that is because the Organizer/Group relationship is many-to-many so this result to a new table called OrganizerGroup.
* All the columns of a specific table is equivalent to the related entity attributes except the Name & Mobile columns of (Organizer, Admin, & Spanner).
* The Name attribute (which is composite type) is converted to 4 columns on the table which are: FName, M1Name, M2Name, & LName.
* The Mobile attribute (which is multi-value type) is transformed into 2 columns on the related table which are: Mobile1, & Mobile2.
* In the relational schema all primary keys are one column only, also no primary key columns (which used as foreign keys) with char type more than 5 in length. As a result of that there are a 14 auto-generated integer type primary key columns as follows:- DestinationId, EligibilityId, CompanyId, AircraftTypeId, FlightRecordId, ApprovalId, ApprovalDetailId, FlightDataId, OrganizerGroupId, GroupId, InActivePeriodId, DepartmentId, QualificationId, DistanceTypeId.