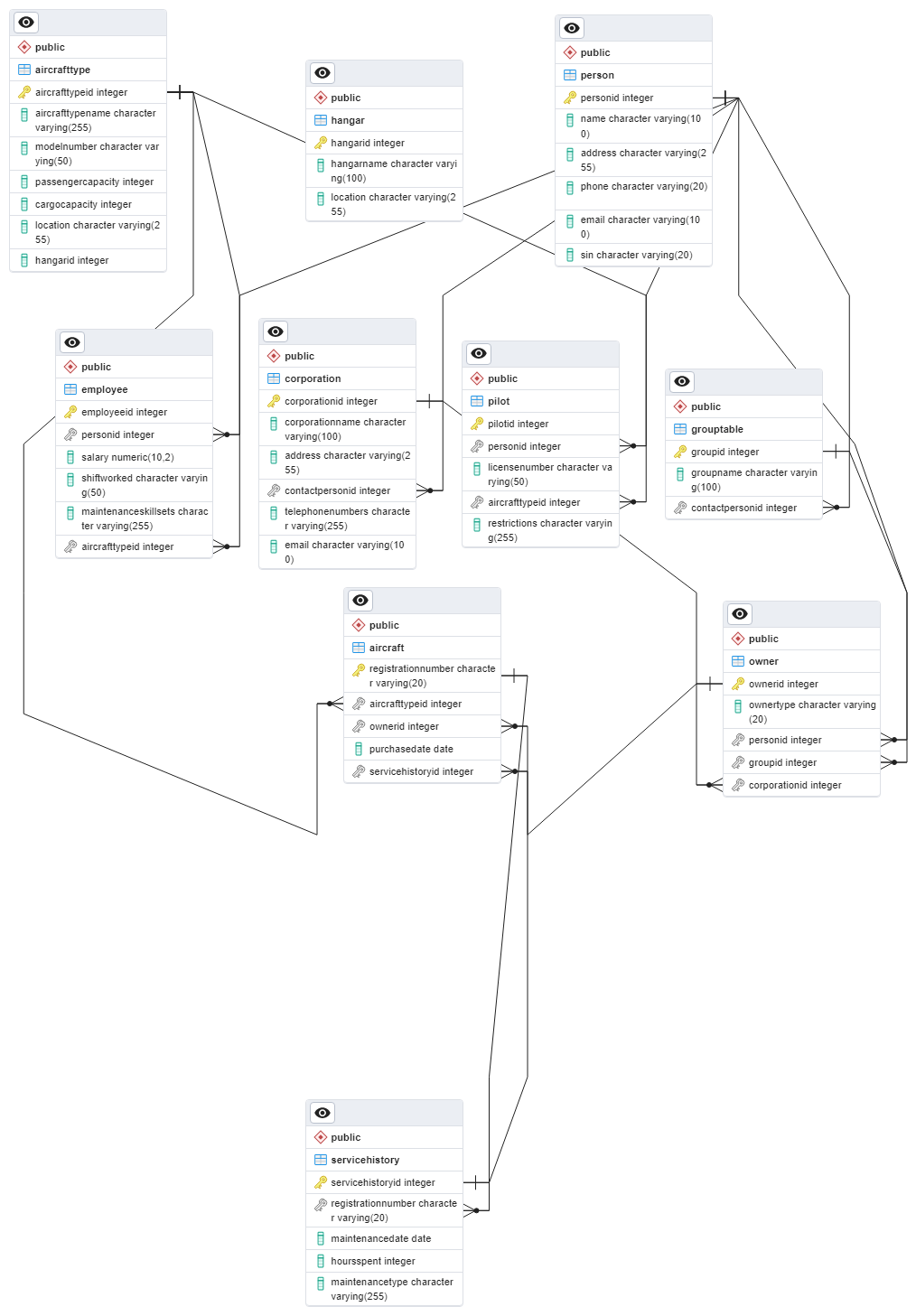
*This is an Airport database.*

**

*ERM ON POSTGRESQL*

*SQL CODE FOR QUERY(DDL CODE FROM QUERY)*

*BEGIN;*

*CREATE TABLE IF NOT EXISTS public.aircraft*

*(*

*CONSTRAINT aircraft\_pkey PRIMARY KEY (registrationnumber)*

*);*

*CREATE TABLE IF NOT EXISTS public.aircrafttype*

*(*

*CONSTRAINT aircrafttype\_pkey PRIMARY KEY (aircrafttypeid)*

*);*

*CREATE TABLE IF NOT EXISTS public.corporation*

*(*

*CONSTRAINT corporation\_pkey PRIMARY KEY (corporationid)*

*);*

*CREATE TABLE IF NOT EXISTS public.employee*

*(*

*CONSTRAINT employee\_pkey PRIMARY KEY (employeeid)*

*);*

*CREATE TABLE IF NOT EXISTS public.grouptable*

*(*

*CONSTRAINT grouptable\_pkey PRIMARY KEY (groupid)*

*);*

*CREATE TABLE IF NOT EXISTS public.hangar*

*(*

*CONSTRAINT hangar\_pkey PRIMARY KEY (hangarid)*

*);*

*CREATE TABLE IF NOT EXISTS public.owner*

*(*

*CONSTRAINT owner\_pkey PRIMARY KEY (ownerid)*

*);*

*CREATE TABLE IF NOT EXISTS public.person*

*(*

*CONSTRAINT person\_pkey PRIMARY KEY (personid)*

*);*

*CREATE TABLE IF NOT EXISTS public.pilot*

*(*

*CONSTRAINT pilot\_pkey PRIMARY KEY (pilotid)*

*);*

*CREATE TABLE IF NOT EXISTS public.servicehistory*

*(*

*CONSTRAINT servicehistory\_pkey PRIMARY KEY (servicehistoryid)*

*);*

*ALTER TABLE IF EXISTS public.aircraft*

*ADD CONSTRAINT fk\_aircraft\_aircrafttype FOREIGN KEY (aircrafttypeid)*

*REFERENCES public.aircrafttype (aircrafttypeid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.aircraft*

*ADD CONSTRAINT fk\_aircraft\_owner FOREIGN KEY (ownerid)*

*REFERENCES public.owner (ownerid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.aircraft*

*ADD CONSTRAINT fk\_aircraft\_servicehistory FOREIGN KEY (servicehistoryid)*

*REFERENCES public.servicehistory (servicehistoryid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.corporation*

*ADD CONSTRAINT fk\_corporation\_contactperson FOREIGN KEY (contactpersonid)*

*REFERENCES public.person (personid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.employee*

*ADD CONSTRAINT fk\_employee\_aircrafttype FOREIGN KEY (aircrafttypeid)*

*REFERENCES public.aircrafttype (aircrafttypeid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.employee*

*ADD CONSTRAINT fk\_employee\_person FOREIGN KEY (personid)*

*REFERENCES public.person (personid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.grouptable*

*ADD CONSTRAINT fk\_grouptable\_contactperson FOREIGN KEY (contactpersonid)*

*REFERENCES public.person (personid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.owner*

*ADD CONSTRAINT fk\_owner\_corporation FOREIGN KEY (corporationid)*

*REFERENCES public.corporation (corporationid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.owner*

*ADD CONSTRAINT fk\_owner\_group FOREIGN KEY (groupid)*

*REFERENCES public.grouptable (groupid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.owner*

*ADD CONSTRAINT fk\_owner\_person FOREIGN KEY (personid)*

*REFERENCES public.person (personid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.pilot*

*ADD CONSTRAINT fk\_pilot\_aircrafttype FOREIGN KEY (aircrafttypeid)*

*REFERENCES public.aircrafttype (aircrafttypeid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.pilot*

*ADD CONSTRAINT fk\_pilot\_person FOREIGN KEY (personid)*

*REFERENCES public.person (personid) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*ALTER TABLE IF EXISTS public.servicehistory*

*ADD CONSTRAINT fk\_servicehistory\_aircraft FOREIGN KEY (registrationnumber)*

*REFERENCES public.aircraft (registrationnumber) MATCH SIMPLE*

*ON UPDATE NO ACTION*

*ON DELETE NO ACTION;*

*END;*

*VARİABLES OF TABLE*

*-- Aircraft Type Table*

*AircraftTypeID INT*

*AircraftTypeName VARCHAR(255)*

*ModelNumber VARCHAR(50)*

*PassengerCapacity INT*

*CargoCapacity INT*

*Location VARCHAR(255)*

*HangarID INT*

*-- Aircraft Table*

*RegistrationNumber VARCHAR(20)*

*AircraftTypeID INT*

*OwnerID INT*

*PurchaseDate DATE*

*ServiceHistoryID INT*

*);*

*-- Owner Table*

*OwnerID INT*

*OwnerType VARCHAR(20)*

*PersonID INT*

*GroupID INT*

*CorporationID INT*

*-- Person Table*

*PersonID INT*

*Name VARCHAR(100)*

*Address VARCHAR(255)*

*Phone VARCHAR(20)*

*Email VARCHAR(100)*

*SIN VARCHAR(20)*

*-- Group Table*

*GroupID INT*

*GroupName VARCHAR(100)*

*ContactPersonID INT*

*-- Corporation Table*

*CorporationID INT*

*CorporationName VARCHAR(100),*

*Address VARCHAR(255),*

*ContactPersonID INT,*

*TelephoneNumbers VARCHAR(255),*

*Email VARCHAR(100),*

*PilotID INT*

*PersonID INT*

*LicenseNumber VARCHAR(50)*

*AircraftTypeID INT*

*Restrictions VARCHAR(255)*

*,*

*-- Employee Table*

*EmployeeID INT*

*PersonID INT,*

*Salary DECIMAL(10, 2),*

*ShiftWorked VARCHAR(50),*

*MaintenanceSkillsets VARCHAR(255),*

*AircraftTypeID INT,*

*-- Hangar Table*

*HangarID INT*

*HangarName VARCHAR(100),*

*Location VARCHAR(255)*

*-- Service History Table*

*ServiceHistoryID INT*

*RegistrationNumber VARCHAR(20),*

*MaintenanceDate DATE,*

*HoursSpent INT,*

*MaintenanceType VARCHAR(255),*

*ANALYSIS OF THE REQUİRMENTS IN FORM OF SEMANTIC TRIPLES:*

1. **Aircraft Type:**
   * (AircraftType, has, AircraftTypeID)
   * (AircraftType, has, AircraftTypeName)
   * (AircraftType, has, ModelNumber)
   * (AircraftType, has, PassengerCapacity)
   * (AircraftType, has, CargoCapacity)
   * (AircraftType, has, Location)
   * (AircraftType, isStoredIn, Hangar)
2. **Aircraft:**
   * (Aircraft, hasRegistration, RegistrationNumber)
   * (Aircraft, ofType, AircraftType)
   * (Aircraft, ownedBy, Owner)
   * (Aircraft, purchasedOn, PurchaseDate)
   * (Aircraft, hasServiceHistory, ServiceHistory)
3. **Owner:**
   * (Owner, hasID, OwnerID)
   * (Owner, hasType, OwnerType)
   * (Owner, isPerson, Person)
   * (Owner, isGroup, Group)
   * (Owner, isCorporation, Corporation)
4. **Person:**
   * (Person, hasID, PersonID)
   * (Person, hasName, Name)
   * (Person, hasAddress, Address)
   * (Person, hasPhone, Phone)
   * (Person, hasEmail, Email)
   * (Person, hasSIN, SIN)
5. **Group:**
   * (Group, hasID, GroupID)
   * (Group, hasName, GroupName)
   * (Group, hasContactPerson, Person)
6. **Corporation:**
   * (Corporation, hasID, CorporationID)
   * (Corporation, hasName, CorporationName)
   * (Corporation, hasAddress, Address)
   * (Corporation, hasContactPerson, Person)
   * (Corporation, hasTelephoneNumbers, TelephoneNumbers)
   * (Corporation, hasEmail, Email)
7. **Pilot:**
   * (Pilot, hasID, PilotID)
   * (Pilot, isPerson, Person)
   * (Pilot, hasLicenseNumber, LicenseNumber)
   * (Pilot, isAuthorizedToFly, AircraftType)
   * (Pilot, hasRestrictions, Restrictions)
8. **Employee:**
   * (Employee, hasID, EmployeeID)
   * (Employee, isPerson, Person)
   * (Employee, hasSalary, Salary)
   * (Employee, worksShift, ShiftWorked)
   * (Employee, hasMaintenanceSkillsets, MaintenanceSkillsets)
   * (Employee, canMaintain, AircraftType)
9. **Hangar:**
   * (Hangar, hasID, HangarID)
   * (Hangar, hasName, HangarName)
   * (Hangar, hasLocation, Location)
10. **Service History:**
    * (ServiceHistory, hasID, ServiceHistoryID)
    * (ServiceHistory, isForAircraft, Aircraft)
    * (ServiceHistory, maintenanceDate, MaintenanceDate)
    * (ServiceHistory, hoursSpent, HoursSpent)
    * (ServiceHistory, hasMaintenanceType, MaintenanceType)