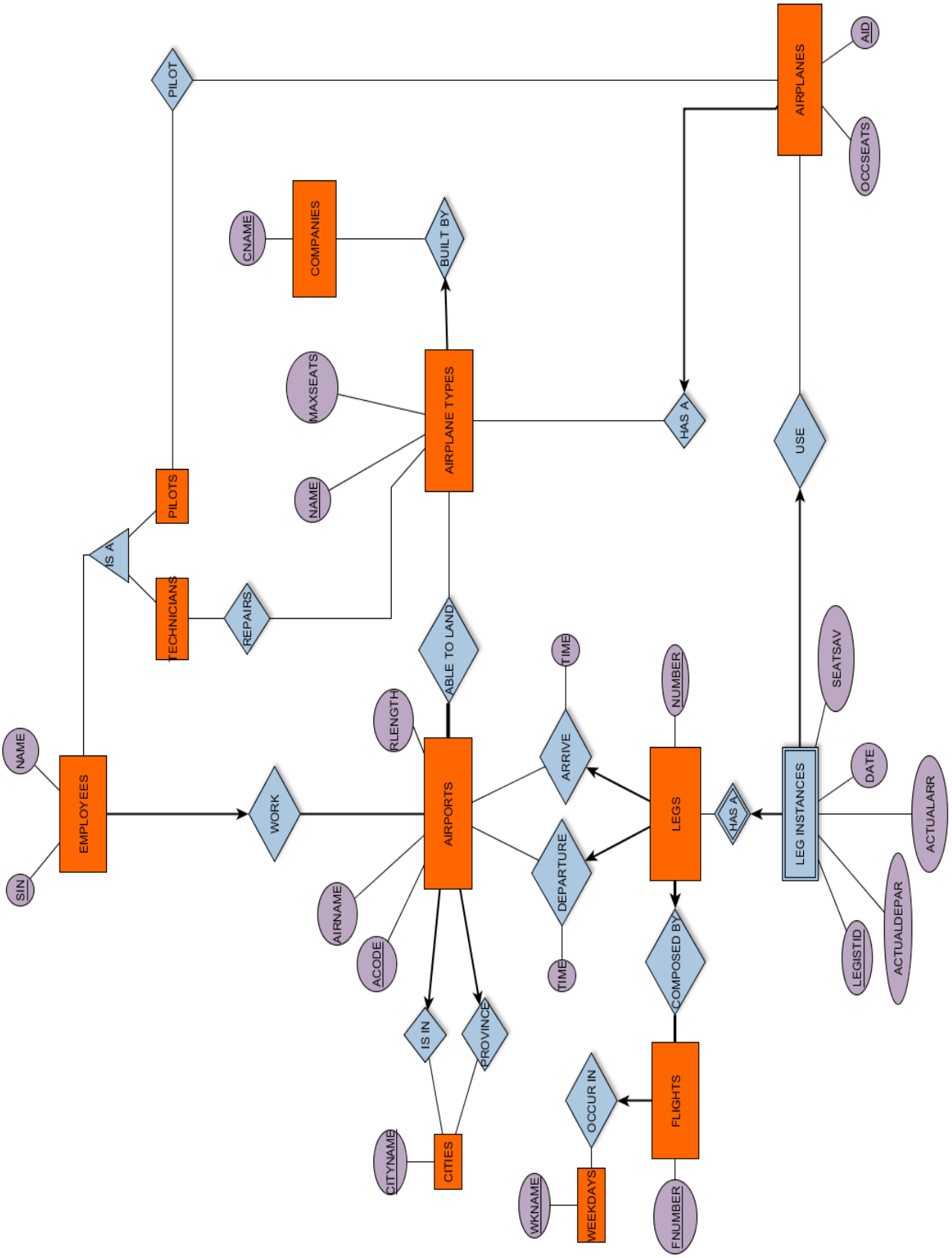
Part A - Alitalia

ER schema



Relational Model

Cities (cityName)

Weekdays (wkname)

Companies (cname)

Airports (acode, airName, rLength, cityName, provinceName)
- cityName is FK to Cities NOT NULL
- provinceName is FK to Cities NOT NULL

Employees (sin, name,acodeWork)
- acodeWork is FK to Airports NOT NULL

In our interpretation an employee can work only in one airport.

Technicians (sin)
- sin is FK to Employees NOT NULL

Pilots (sin)
- sin is FK to Employees NOT NULL

AirplaneTypes (name, maxSeats, cname)
- cname is FK to Companies NOT NULL

An airplane type can only be built by one company.

Airplanes (aid, occseats, name)
- name is FK to AirplaneTypes NOT NULL

PilotsAirplanes (sin, aid)
- sin is FK to Pilots NOT NULL
- aid is FK to Airplanes NOT NULL

Flights (fnumber, wkname)
- wkname is FK to Weekdays NOT NULL

TechniciansAT (name,cname, sin)
- sin is FK to Technicians NOT NULL
- (name,cname) is FK to AirplaneTypes NOT NULL

Legs (number, acodeDep, timeDep, acodeArr, timeArr, fnumber)
- acodeDep is FK to Airports NOT NULL
- acodeArr is FK to Airports NOT NULL
- fnumber is a FK to Flights NOT NULL

In the relational model we can't fully capture the constraints of the "composed by" relationship between Flights and Legs in the ER diagram.

LegInstances (legistid, date, seatsAv, actualArr, actualDepar, number, aid)

- number is a FK to Legs NOT NULL
- aid is FK to Airplanes. NOT NULL

Legistid is the primary key of Leginstances created by us: it let us distinguish every single tuple.

SQL

```
CREATE TABLE Cities(
        cityName varchar(255) NOT NULL,
        PRIMARY KEY (cityName));

CREATE TABLE Weekdays(
        wkname varchar(255) NOT NULL,
        PRIMARY KEY(wkname));

CREATE TABLE Companies(
        cname varchar(255) NOT NULL,
        PRIMARY KEY(cname));

CREATE TABLE Airports(
        acode int NOT NULL,
        airName varchar(255),
        rLength int,
        cityName varchar(255) NOT NULL,
        provinceName varchar(255) NOT NULL,
        PRIMARY KEY (acode),
        FOREIGN KEY (cityName) REFERENCES Cities(cityName) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (provinceName) REFERENCES Cities(cityName) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Employees(
        sin int NOT NULL,
        name varchar(255) NOT NULL,
        acodeWork int NOT NULL,
        PRIMARY KEY (sin),
        FOREIGN KEY(acodeWork) REFERENCES Airports(acode) ON DELETE
        CASCADE ON UPDATE CASCADE);
```

```
CREATE TABLE Technicians(
        sin int NOT NULL,
        PRIMARY KEY (sin),
        FOREIGN KEY (sin) REFERENCES Employees(sin) ON DELETE CASCADE ON
        UPDATE CASCADE);

CREATE TABLE Pilots(
        sin int NOT NULL,
        PRIMARY KEY (sin),
        FOREIGN KEY (sin) REFERENCES Employees(sin) ON DELETE CASCADE ON
        UPDATE CASCADE);

CREATE TABLE AirplaneTypes(
        name varchar(255) NOT NULL,
        maxSeats int,
        cname varchar(255) NOT NULL,
        PRIMARY KEY (name,cname),
        FOREIGN KEY (cname) REFERENCES Companies(cname) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Airplanes(
        aid int NOT NULL,
        occseats int,
        name varchar(255) NOT NULL,
        PRIMARY KEY(aid),
        FOREIGN KEY(name) REFERENCES AirplaneTypes(name) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE PilotsAirplanes(
        sin int NOT NULL,
        aid int NOT NULL,
        PRIMARY KEY(sin,aid),
        FOREIGN KEY(sin) REFERENCES Pilots(sin) ON DELETE CASCADE ON
        UPDATE CASCADE,
        FOREIGN KEY(aid) REFERENCES Airplanes(aid) ON DELETE CASCADE ON
        UPDATE CASCADE);

CREATE TABLE Flights(
        fnumber int NOT NULL,
        wkname varchar(255)NOT NULL,
        PRIMARY KEY(fnumber),
        FOREIGN KEY(wkname) REFERENCES Weekdays(wkname) ON DELETE
        CASCADE ON UPDATE CASCADE);
```

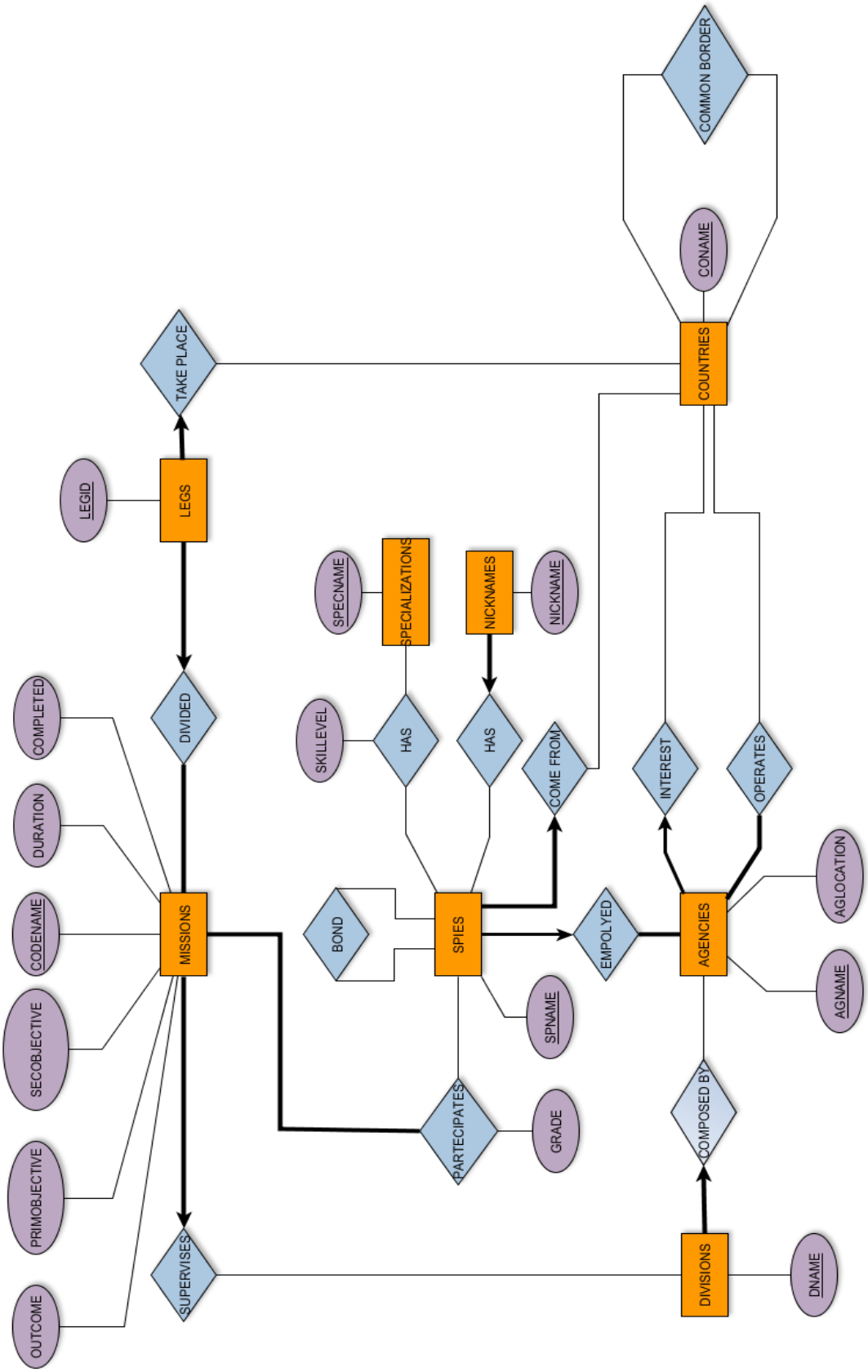
```
CREATE TABLE TechniciansAT(
        name varchar(255) NOT NULL,
        sin int NOT NULL,
        cname varchar(255),
        PRIMARY KEY(name,cname,sin),
        FOREIGN KEY(sin) REFERENCES Technicians(sin) ON DELETE CASCADE ON
        UPDATE CASCADE,
        FOREIGN KEY(name,cname) REFERENCES AirplaneTypes(name,cname) ON
        DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE Legs(
        number int NOT NULL,
        fnumber int NOT NULL,
        acodeDep int NOT NULL,
        timeDep DATETIME,
        acodeArr int NOT NULL,
        timeArr DATETIME,
        PRIMARY KEY(number,fnumber),
        FOREIGN KEY(acodeDep) REFERENCES Airports(acode) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(acodeArr) REFERENCES Airports(acode) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(fnumber) REFERENCES Flights(fnumber) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE LegInstances(
        legistid int NOT NULL,
        date DATE,
        seatsAv int,
        actualArr DATETIME,
        actualDepar DATETIME,
        number int NOT NULL,
        aid int NOT NULL,
        PRIMARY KEY(legistid),
        FOREIGN KEY(number) REFERENCES Legs(number) ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(aid) REFERENCES Airplanes(aid) ON DELETE CASCADE ON
        UPDATE CASCADE);
```

Part B - IMF

ER schema



Relational Model

Countries (coName)

CommonBorder (coName1, coName2)
- coName1 is FK to Countries NOT NULL
- coName2 is FK to Countries NOT NULL

Operates (agName, coName)
- agName is FK to Agencies NOT NULL
- coName is FK to Countries NOT NULL

We cannot fully capture the constraints of the "operates" relationship between Agencies and Countries.

Agencies (agName, agLocation, coNameInterest)
- coNameInterest is a FK to Countries NOT NULL

Divisions (dName, agName)
- agName is a FK to Agencies NOT NULL

Missions (codeName, duration, completed, primObjective, secObjective, outcome, dName)
- dName is FK to Divisions NOT NULL

Legs (legid, codeName, coName)
- codeName is a FK to Missions NOT NULL
- coName is FK to Countries NOT NULL

We can't fully capture the constraints of the "divided" relationship between Missions and Legs.

Spies (spName, coName)
- coName is a FK to Countries NOT NULL

We suppose that there aren't spies with the same name.

Bond (spName1, spName2)
- spName1 is FK to Spies NOT NULL
- spName2 is FK to Spies NOT NULL

Specializations (specName)

SpiesSpecialization (specName, spName, skillLevel)
- specName is FK to Specializations NOT NULL
- spName is FK to Spies NOT NULL

Nicknames (nickName, spName)
- spName is FK to Spies NOT NULL

Partecipates (spName, codeName, grade)

- spName is FK to Spies NOT NULL
- codeName is FK to Missions. NOT NULL

We consider different spies that partecipate in one mission as one group only.
We can't fully capture the constraints of the "partecipates" relationship between Spies and Missions.

SQL

```
CREATE TABLE Countries (
        coName varchar(255) NOT NULL,
        PRIMARY KEY(coName));

CREATE TABLE CommonBorder (
        coName1 varchar(255) NOT NULL,
        coName2 varchar(255) NOT NULL,
        PRIMARY KEY (coName1,coName2),
        FOREIGN KEY (coName1) REFERENCES Countries(coName) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (coName2) REFERENCES Countries(coName) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Agencies(
        agName varchar(255) NOT NULL,
        agLocation varchar(255),
        coNameInterest varchar(255)NOT NULL,
        PRIMARY KEY(agName),
        FOREIGN KEY(coNameInterest) REFERENCES Countries(coName) ON
        DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE Operates(
        agName varchar(255)NOT NULL,
        coName varchar(255) NOT NULL,
        PRIMARY KEY(agName,coName),
        FOREIGN KEY(agName) REFERENCES Agencies(agName) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(coName) REFERENCES Agencies(agName) ON DELETE
        CASCADE ON UPDATE CASCADE);
```

```
CREATE TABLE Divisions (
        dName varchar(255) NOT NULL,
        agName varchar(255) NOT NULL,
        PRIMARY KEY(dName),
        FOREIGN KEY(agName) REFERENCES Agencies(agName) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Missions(
        codeName varchar(255) NOT NULL,
        duration int,
        completed boolean,
        primObjective varchar(255) NOT NULL,
        secObjective varchar(255),
        outcome varchar(255),
        dName varchar(255) NOT NULL,
        PRIMARY KEY(codeName),
        FOREIGN KEY(dName) REFERENCES Divisions(dName) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Legs(
        legid int NOT NULL,
        codeName varchar(255) NOT NULL,
        coName varchar(255) NOT NULL,
        PRIMARY KEY(legid),
        FOREIGN KEY(codeName) REFERENCES Missions(codeName) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(coName) REFERENCES Countries(coName) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Spies(
        spName varchar(255) NOT NULL,
        coName varchar(255) NOT NULL,
        PRIMARY KEY(spName),
        FOREIGN KEY(coName) REFERENCES Countries(coName) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Bond(
        spName1 varchar(255) NOT NULL,
        spName2 varchar(255) NOT NULL,
        PRIMARY KEY(spName1,spName2),
        FOREIGN KEY(spName1) REFERENCES Spies(spName) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(spName2) REFERENCES Spies(spName) ON DELETE
        CASCADE ON UPDATE CASCADE);
```

```
CREATE TABLE Specializations(
        specName varchar(255) NOT NULL,
        PRIMARY KEY(specName));

CREATE TABLE SpiesSpecializaton(
        specName varchar(255) NOT NULL,
        spName varchar(255) NOT NULL,
        skillLevel int,
        PRIMARY KEY(specName,spName),
        FOREIGN KEY(specName) REFERENCES Specializations(specName) ON
        DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(spName) REFERENCES Spies(spName) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Nicknames(
        nickName varchar(255) NOT NULL,
        spName varchar(255) NOT NULL,
        PRIMARY KEY(nickName),
        FOREIGN KEY(spName) REFERENCES Spies(spName) ON DELETE
        CASCADE ON UPDATE CASCADE);

CREATE TABLE Partecipates(
        spName varchar(255),
        codeName varchar(255),
        grade int,
        PRIMARY KEY(spName,codeName),
        FOREIGN KEY(codeName) REFERENCES Missions(codeName) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY(spName) REFERENCES Spies(spName) ON DELETE
        CASCADE ON UPDATE CASCADE);
```