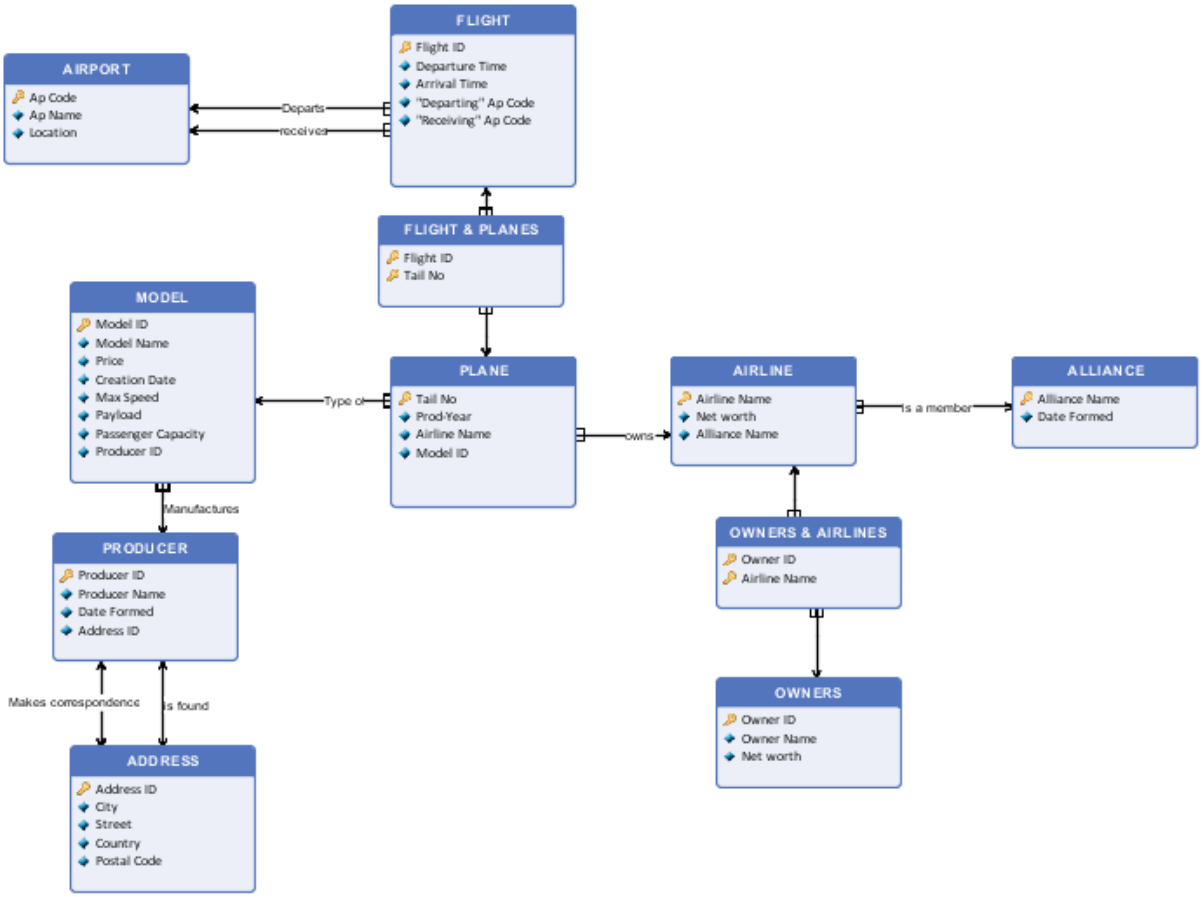
Allan Kirui, 183036 21/12/2021

Planes, Producers, Airlines

1. Krótki opis bazy danych

   Client - A plane/flight and airline tracking system
   Purpose- Storing and searching for information about planes, producers and airplanes
   User- Plane and airline enthusiasts

2. Schemat graficzny bazy danych (diagram ERD)



3. Opis zbioru encji (typy, klucze, ...)

ALLIANCE			
Cardinality: around 100 (5 years) Has general information about an alliance like its name and date of creation Created: When an Alliance is created Updated: When its dissolved Deleted: When an alliance is ended			
Name	**Primary Key**	**Type/Limit**	**Description**
AllianceName	Yes	Sequence of letters from 2 up to 20	Name used to identify an air alliance
`AllianceName VARCHAR(20) CHECK(LEN(AllianceName)>=2) PRIMARY KEY,`			
DateFormed	No	Date / YYYY-MM-DD	The creation date of an air alliance
`DateFormed DATE CONSTRAINT DateFormed CHECK (YEAR(DateFormed) >= 1997) --oldest alliance is from 1997`			

AIRLINE			
Cardinality: around 10,000 (5 years) A set of entities that have general information about an airline like its name and net worth Created: When an airline opens Updated: When an airline increases net worth Deleted: When an airline is close down			
Name	**Primary Key**	**Type/Limit**	**Description**
AirlineName	Yes	Sequence of letters from 2 up to 50	Name used to identify an airline
`AirlineName VARCHAR(50)  CHECK(LEN(AirlineName)>=2) PRIMARY KEY,`			
NetWorth	No	int of max 10 digits	The total wealth of an airline, taking account of all financial assets and liabilities.
`NetWorth DECIMAL(10,0), --max value is in billion dollar range`			
AllianceName	No	Sequence of letters up to 20	Name used to identify an air alliance

```
AllianceName VARCHAR(20) REFERENCES ALLIANCE  ON DELETE CASCADE ON UPDATE CASCADE
```

OWNERS

Cardinality: around 1000 (5 years)
A set of entities that have general information about an owner of an airline

Created: When an owner buys a stake in an airline
Updated: When stake changes
Deleted: When owner sells their stake

Name	Primary Key	Type/Limit	Description
OwnerID	Yes	INT	Identification number of owner

```
OwnerID  INT IDENTITY(1,1) PRIMARY KEY,
```

Name	Primary Key	Type/Limit	Description
Owner Name	No	Sequence of letters up to 50	Name of owner

```
OwnerName VARCHAR(50) NOT NULL,
```

Name	Primary Key	Type/Limit	Description
NetWorth	No	Decimal of max 14 digits	The total wealth of an owner, taking account of all financial assets and liabilities.

```
NetWorth DECIMAL(14,0), --max value is in tens trillion dollar range
```

OWNERS & AIRLINES

A set of entities that contain information about airlines and their respective owners

Name	Primary Key	Type/Limit	Description
OwnerID	Yes	INT	Identification number of owner

```
OwnerID INT NOT NULL,  PRIMARY KEY (OwnerID,AirlineName)
CONSTRAINT OwnIdent FOREIGN KEY (OwnerID) REFERENCES OWNERS ON DELETE CASCADE ON UPDATE CASCADE,
```

Airline Name	Yes	Sequence of letters up to 50	Name of an Airline
AirlineName VARCHAR(50) NOT NULL, PRIMARY KEY (OwnerID,AirlineName) CONSTRAINT AirlName FOREIGN KEY (AirlineName) REFERENCES AIRLINE,			

ADDRESSES			
Cardinality:around 1000 (5 years) A set of entities that contain information about the particulars of the place where an organization is situated Created: When manufacturer opens shop Updated: When address changes Deleted: When manufacturer closes shop			
Name	**Primary Key**	**Type/Limit**	**Description**
AddressID	Yes	INT	ID number used to identify the address
AddressID int IDENTITY(1,1) PRIMARY KEY,			
City	No	Sequence of letters up to 85	Name of owner
City VARCHAR(85)NOT NULL,			
Street	No	Sequence of letters up to 85	Public road in city where manufacturer is found
Street VARCHAR(85)NOT NULL,			
Country	No	Sequence of letters up to 60	Nation where address is found
Country VARCHAR(60) NOT NULL,			
Postal Code	No	Sequence of letters up to 6	A series of letters or digits or both, sometimes including spaces or punctuation, included in a postal address for the purpose of sorting mail
PostalCode VARCHAR(6) NOT NULL UNIQUE CHECK (PostalCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]')			

PRODUCER			
Cardinality: around 20 (5 years) A set of entities that have general information about a plane manufacture for example their name Created: When a manufacturer opens shop Updated: When they change their company name Deleted: When manufacturer goes out of business			
Name	**Primary Key**	**Type/Limit**	**Description**
ProducerID	Yes	INT	Identification number of plane manufacturer
`ProducerID int IDENTITY(1,1) PRIMARY KEY,`			
ProducerName	No	Sequence of letters from 2 up to 50	Name used to identify plane manufacturer
`ProducerName VARCHAR(50) CHECK(LEN(ProducerName)>=2) UNIQUE,`			
DateFormed	No	Date / YYYY-MM-DD	The creation date of the plane manufacturer
`DateFormed DATE,`			
AddressID	No	INT	ID number used to identify the address
`AddressID int REFERENCES ADDRESSES`			

MODEL			
Cardinality: around 1000 (5 years) A set of entities that have general information about a plane model, like its price , its manufacturer, its specs Created: When a model is created Updated: When price is changed Deleted: When the model is retired and all planes are destroyed of said model			
Name	**Primary Key**	**Type/Limit**	**Description**
ModelID	Yes	Sequence of letters from 4 up to 50	Used to identify the model of a plane

ModelID VARCHAR(50) CHECK(LEN(ModelID)>=4) PRIMARY KEY ,			
Model Name	No	Sequence of letters from 2 up to 50	An alphanumeric code between two and six characters in length used to identify a specific model
ModelName VARCHAR(50) CHECK(LEN(ModelName)>=2) UNIQUE,			
Price	No	Decimal with max of 11 digits	A number used to show the cost of buying a plane in USD
Price DECIMAL(11,2),			
CreationDate	No	Date / YYYY-MM-DD	The creation date of the plane model
CreationDate DATE NOT NULL,			
MaxSpeed	No	Decimal with max 3 digits	A number showing the maximum velocity a plane model can reach
MaxSpeed DECIMAL(3,0) NOT NULL,			
Payload	No	Decimal with max 3 digits	Max load of a plane in tonnes
Payload DECIMAL(3,1) NOT NULL,			
PassengerCap	No	Decimal with max 3 digits	Number of passenger seats on a plane
PassengerCap DECIMAL(3,0) NOT NULL,			
ProducerID	No	INT	Identification number of plane manufacturer
ProducerID int REFERENCES PRODUCER ON DELETE CASCADE ON UPDATE CASCADE			

PLANE
Cardinality: around 100,000 (5 years) A set of entities that have general information about a plane, like its flight status, its production year Created: When a plane is manufactured Updated: When a plane is sold Deleted: When a plane is destroyed/put out of service

Name	Primary Key	Type/Limit	Description
TailNo	Yes	Sequence of letters up to 6	An alphanumeric code between two and six characters in length used to identify a specific airplane
`TailNo VARCHAR(6) NOT NULL PRIMARY KEY,`			
ProdYear	No	Date in years/ YYYY	The 12 calendar month period during which a plane was first produced
`ProdYear DECIMAL(4,0) DEFAULT '9999',`			
Airline Name	No	Sequence of letters from 2 up to 50	Name of an Airline
`AirlineName VARCHAR(50) REFERENCES AIRLINE ON DELETE CASCADE ON UPDATE CASCADE,`			
ModelID	No	Sequence of letters from 4 up to 50	Used to identify the model of a plane
`ModelID VARCHAR(50) REFERENCES MODEL ON DELETE CASCADE ON UPDATE CASCADE`			

AIRPORT			
Cardinality: around 1000 (5 years) A set of entities that have the name of airports Created: When an airport is first opened Updated: When the name of an airport changes Deleted: When an airport is demolished			
Name	**Primary Key**	**Type/Limit**	**Description**
Airport Code	Yes	Sequence of letters up to 4	ICAO code used to idenify an airport
`AirportCode VARCHAR(4) CHECK (AirportCode LIKE '[A-Z][A-Z][A-Z][A-Z]') PRIMARY KEY, --ICAO code`			
Airport Name	No	Sequence of letters up to 50	Name of an airport
`AirportName VARCHAR(50) NOT NULL,`			
Airport Location	No	Sequence of letters up to 85	Location of the airport

`AirportLocation VARCHAR(85) NOT NULL`

FLIGHT			
Cardinality: 1,000,000 (1 year) A set of entities that have information about flights, like its destination Created: When a flight takes off Updated: When a flight lands Deleted: When a flight lands			
Name	**Primary Key**	**Type/Limit**	**Description**
FlightID	Yes	Sequence of letters up to 7	A code for an airline service consisting of two-character airline designator and a 1 to 4 digit number.
`FlightID VARCHAR(7) CHECK (FlightID LIKE '[A-Z][A-Z] [0-9][0-9][0-9][0-9]') PRIMARY KEY,`			
DepartingApCode	No	Sequence of letters up to 4	ICAO code used to idenify the departing airport
`DepartingApCode VARCHAR(4),` `CONSTRAINT DepApCode FOREIGN KEY (DepartingApCode) REFERENCES AIRPORT ON DELETE CASCADE ON UPDATE CASCADE,`			
ReceivingApCode	No	Sequence of letters up to 4	ICAO code used to idenify the receiving airport
`ReceivingApCode VARCHAR(4),` `CONSTRAINT RecApCode FOREIGN KEY (ReceivingApCode) REFERENCES AIRPORT ON DELETE NO ACTION ON UPDATE NO ACTION,`			
DepartureTime	No	TIME / HH:MM:SS	The time the flight started
`DepartureTime TIME NOT NULL,`			
Arrival Time	No	TIME / HH:MM:SS	The expected arrival time
`ArrivalTime TIME NOT NULL,`			

FLIGHT & PLANES			
A set of entities that contain information about flights and planes on said flights			
Name	**Primary Key**	**Type/Limit**	**Description**
TailNo	Yes	Sequence of letters up to 6	A code for an airline service consisting of two-character airline designator and a 1 to 4 digit number.
`TailNo VARCHAR(6) NOT NULL ,` `CONSTRAINT TailNum FOREIGN KEY (TailNo) REFERENCES PLANE ON DELETE CASCADE ON UPDATE CASCADE,`			
FlightID	Yes	Sequence of letters up to 7	A code for an airline service consisting of two-character airline designator and a 1 to 4 digit number.
`FlightID VARCHAR(7) NOT NULL ,` `CONSTRAINT FlightIdent FOREIGN KEY (FlightID) REFERENCES FLIGHT ON DELETE CASCADE ON UPDATE CASCADE,`			

4. Schemat relacyjnej bazy danych

AIRLINE (Airline Name, Net Worth, FK_ALLIANCE_NAME)
(Airline Name) KEY
(FK_ALLIANCE_NAME) REF ALLIANCE

ALLIANCE (Alliance Name, Date Formed)
(Alliance Name) KEY

OWNERS (Owner ID, Owner Name, NetWorth)
(Owner ID) KEY

OWNERS & AIRLINES (Airline Name, Owner ID)
(Owner ID) KEY REF OWNER
(Airline Name) KEY REF AIRLINE

ADDRESSES (Address ID, City, Street, Country, Postal Code)
(Address ID) KEY

PRODUCER (Producer ID, Producer Name, Date formed, FK_ADDRESS_ID)
(Producer ID) KEY
(FK_ADDRESS_ID) REF ADDRESS

MODEL(Model ID, Model Name, Price, Creation Date, Max Speed, Payload, Passenger Cap, FK_PRODUCER_ID)
(Model ID) KEY
(FK_PRODUCER_ID) REF PRODUCER

PLANE (Tail No, Prod-Year, FK_AIRLINE_NAME, FK_MODEL_ID)
(Tail No) KEY
(FK_MODEL_ID) REF MODEL
(FK_AIRLINE_NAME) REF AIRLINE

AIRPORT (Ap Code, Ap Name)
(Ap Code) KEY

FLIGHT (Flight ID,Destination, Departure location, Departure Time, Arrival time, FK_"Departing"_AP_CODE,FK_"Receiving"_AP_CODE)
(Flight ID) KEY
(FK_"Departing"_AP_CODE) REF "DEPARTING" AIRPORT
(FK_"Receiving"_AP_CODE) REF "RECEIVING" AIRPORT

FLIGHT & PLANES (Flight ID, Tail No)
(Flight ID) KEY REF FLIGHT
(Tail No) KEY REF PLANE

5. Szczegółowy opis utworzonych tabel pod kątem zastosowanych ograniczeń np. NOT NULL, UNIQUE, CHECK, DEFAULT, klucze …

```
CREATE TABLE ALLIANCE
(
    AllianceName VARCHAR(20) CHECK(LEN(AllianceName)>=2) PRIMARY KEY,
    DateFormed DATE CONSTRAINT DateFormed CHECK (YEAR(DateFormed) >= 1997) --oldest alliance is from 1997
);
```

AllianceName has to have at least two characters
DateFormed oldest alliance in existence is from 1997

```
CREATE TABLE AIRLINE
(
    AirlineName VARCHAR(50)  CHECK(LEN(AirlineName)>=2) PRIMARY KEY,
    NetWorth DECIMAL(10,0), --max value is in billion dollar range
    AllianceName VARCHAR(20) REFERENCES ALLIANCE  ON DELETE CASCADE ON UPDATE CASCADE
);
```

AirlineName has to have at least two characters

```
CREATE TABLE OWNERS
(
    OwnerID  INT IDENTITY(1,1) PRIMARY KEY,
    OwnerName VARCHAR(50) NOT NULL,
    NetWorth DECIMAL(14,0), --max value is in tens trillion dollar range
);
```

OwnerName Name of the owner is important and required, so it can't be null

```
CREATE TABLE OWNERS_AND_AIRLINES
(
    OwnerID INT NOT NULL,
    AirlineName VARCHAR(50) NOT NULL,
    CONSTRAINT OwnIdent FOREIGN KEY (OwnerID) REFERENCES OWNERS ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT AirlName FOREIGN KEY (AirlineName) REFERENCES AIRLINE,
    PRIMARY KEY (OwnerID,AirlineName) --To prevent repitition
);
```

OwnerID is a primary key that is constrained to a foreign key from the OWNERS table therefore it can't be empty
AirlineName is a primary key that is constrained to a foreign key from the AIRLINE table therefore it can't be empty

```
CREATE TABLE ADDRESSES
(
    AddressID  int IDENTITY(1,1) PRIMARY KEY,
    City VARCHAR(85)NOT NULL,
    Street VARCHAR(85)NOT NULL,
    Country VARCHAR(60) NOT NULL,
    PostalCode VARCHAR(6) NOT NULL UNIQUE CHECK (PostalCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]')
);
```

City City name in an address is mandatory
Street street name is mandatory in an address
Country country name is needed in an address
PostalCode Format of postal code is 5 digits and all are different

```
CREATE TABLE PRODUCER
(
    ProducerID int IDENTITY(1,1) PRIMARY KEY,
    ProducerName VARCHAR(50)  CHECK(LEN(ProducerName)>=2) UNIQUE,
    DateFormed DATE,
    AddressID int REFERENCES ADDRESSES --ON DELETE CASCADE ON UPDATE CASCADE
);
```

ProducerName a producer name has to have at least 2 characters

```
CREATE TABLE MODEL
(
    ModelID VARCHAR(50) CHECK(LEN(ModelID)>=4) PRIMARY KEY ,
    ModelName VARCHAR(50)  CHECK(LEN(ModelName)>=2) UNIQUE,
    Price DECIMAL(11,2) NOT NULL,
    CreationDate DATE NOT NULL,
    MaxSpeed DECIMAL(3,0) NOT NULL,
    Payload DECIMAL(3,1) NOT NULL,
    PassengerCap DECIMAL(3,0) NOT NULL,
    ProducerID int REFERENCES PRODUCER ON DELETE CASCADE ON UPDATE CASCADE
);
```

Price Model has to have a price
CreationDate A model has a creation date and I don't think it should be empty
MaxSpeed Each plane model has a maximum speed
Payload Each plane has a maximum payload
PassengerCap Each plane has a maximum passenger capacity

```
CREATE TABLE PLANE
(
    TailNo VARCHAR(6) NOT NULL PRIMARY KEY,
    ProdYear DECIMAL(4,0) DEFAULT '9999', --How to make sure value here is >= CreationDate in Model?
    AirlineName VARCHAR(50) REFERENCES AIRLINE  ON DELETE CASCADE ON UPDATE CASCADE,
    ModelID VARCHAR(50) REFERENCES MODEL ON DELETE CASCADE ON UPDATE CASCADE
);
```

TailNo All commercial planes have a tail number
ProdYear If no input in here, it defaults to 9999 which is obviously not yet the year

```
CREATE TABLE AIRPORT
(
    AirportCode VARCHAR(4) CHECK (AirportCode LIKE '[A-Z][A-Z][A-Z][A-Z]') PRIMARY KEY, --ICAO code
    AirportName VARCHAR(50) NOT NULL,
    AirportLocation VARCHAR(85) NOT NULL
);
```

AirportCode All airports and airstrips have an ICAO code which consists of 4 letters
AirportName Each airport has a name
AirportLocation Each airport is located somewhere on the world

```
CREATE TABLE FLIGHT
(
    FlightID VARCHAR(7) CHECK (FlightID LIKE '[A-Z][A-Z] [0-9][0-9][0-9][0-9]') PRIMARY KEY,
    DepartingApCode VARCHAR(4),
    ReceivingApCode VARCHAR(4),
    DepartureTime TIME NOT NULL,
    ArrivalTime TIME NOT NULL,
    CONSTRAINT DepApCode FOREIGN KEY (DepartingApCode) REFERENCES AIRPORT ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT RecApCode FOREIGN KEY (ReceivingApCode) REFERENCES AIRPORT ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT checkRecApCode CHECK
    (
        ReceivingApCode != DepartingApCode
    )
);
```

FlightID Each flight consists of 2 letters and 4 numbers
DepartingApCode & ReceivingApCode with an assumption that a flight can depart from an airport and land in the same airport

DepartureTime Each flight has a departure time if it's happening
ArrivalTime Each flight has an expected arrival time

```
CREATE TABLE FLIGHT_AND_PLANES
(
    TailNo VARCHAR(6) NOT NULL ,
    FlightID VARCHAR(7) NOT NULL ,
    CONSTRAINT TailNum FOREIGN KEY (TailNo) REFERENCES PLANE ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FlightIdent FOREIGN KEY (FlightID) REFERENCES FLIGHT ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (TailNo,FlightID) --To prevent repitition
);
```

TailNo is a primary key that is constrained to a foreign key from the Plane table therefore it can't be empty
FlightID is a primary key that is constrained to a foreign key from the Flight table therefore it can't be empty