DB ASSIGNMENT

ERD and SQL

CS2005

muhammad rafey and Hamza Aamir

GITHUB LINK : https://github.com/MARafey/Data-Base-Airport-SQL-system.git

This SQL code is a collection of CREATE TABLE statements that define the schema for a database used by an airline company to manage their operations. It also includes INSERT statements to populate the tables with initial data and a set of queries as demanded in the assignment.

The code begins by dropping any existing tables with the same names, using the DROP TABLE IF EXISTS statement. This is a safety measure to ensure that any previous data is removed before creating the new tables.

The first table created is PERSON, which stores personal information about employees and owners. This includes their social security number (SSN), name, address, and phone number.

The next table is Pilot, which stores information about pilots. It includes their license number, any restrictions they may have, and a reference to their corresponding PERSON record.

The Plane_Type table stores information about different types of airplanes, including their model number, passenger capacity, and weight.

The Hangar table stores information about the hangars owned by the airline company. It includes an ID number, the number of planes that can be stored in the hangar, and its location.

The AirPlane table stores information about individual airplanes, including their registration number, the type of airplane (OF_TYPE), the hangar where it is stored (STORED_IN), and whether it is currently undergoing maintenance (UnderMaintenance).

The SERVICE table stores information about maintenance services, including a unique service ID, the date of the service, the number of hours worked, and a work code.

The Employee table stores information about the airline company's employees, including their employee ID, a reference to their corresponding PERSON record, their salary, and the shift they work.

The MAINTAIN table is a junction table that connects Employee and SERVICE. It stores information about which employees worked on which services.

The owners table stores information about the airplane owners, including their SSN, name, address, and phone number.

The Corportation table stores information about the corporations that own the airplane, including the corporation's name, address, phone number, and TIN (Tax Identification Number). The TIN is a foreign key that references the owners table.

The OWNS table is a junction table that connects owners and AirPlane. It stores information about which owners own which airplanes.

The FLIES table is a junction table that connects Plane_Type and Pilot. It stores information about which pilots are certified to fly which types of airplanes.

The PLANE_SERVICE table is a junction table that connects AirPlane and SERVICE. It stores information about which airplanes have received which services.

The WORKS_ON table is a junction table that connects Plane_Type and Employee. It stores information about which employees are trained to work on which types of airplanes.

explanation of queries:

–Q3 This query selects the RegNo column from the AirPlane table, where the RegNo value is not present in the Reg column of the PLANE_SERVICE table. This means it will return the registration numbers of the airplanes that have not been serviced.

–Q4 This query selects the Cname and CAddress columns from the Coporation table, and joins it with three other tables: owners, OWNS, and AirPlane. The join conditions are based on the TIN, SSN, and Reg columns. The query also joins the AirPlane table with another table called Plane_Type, based on the

OF_TYPE and ModelNo columns. The query then filters the results by using a WHERE clause that specifies that the PassengerCapacity column of the Plane_Type table must be greater than 200. This means it will return the names and addresses of the corporations that own airplanes with a passenger capacity of more than 200.

–Q5 This query selects the average value of the Salary column from the Employee table, and groups it by the Shift_Worked column. The query then filters the results by using a WHERE clause that specifies that the Shift_Worked column must be equal to 'Night Shift'. This means it will return the average salary of the employees who work in the night shift.

–Q6 This query selects the top three values of the EmployeeID column from the MAINTAIN table, and sums up the values of the Hours column from the SERVICE table for each employee. The query joins the MAINTAIN table with the SERVICE table based on the ServiceID column. The query then groups the results by the EmployeeID column and orders them by the TotalHours column in descending order. This means it will return the employee IDs and total hours of service of the top three employees who have performed the most maintenance work.

–Q7 This query selects the RegNo and ModelNo columns from the AirPlane table, and joins it with three other tables: Plane_Type, PLANE_SERVICE, and SERVICE. The join conditions are based on the OF_TYPE, Reg, and ServiceID columns. The query then filters the results by using a WHERE clause that specifies that the Date column of the SERVICE table must be greater than or equal to '2020-09-08'. This means it will return the registration numbers and model numbers of the airplanes that have been serviced after September 8th, 2020.

–Q8 This query selects the o_name and Phone columns from the owners table, and joins it with another table called OWNS based on the SSN column. The query then filters the results by using a WHERE clause that specifies that the Pdate column of the OWNS table must be greater than or equal to '2020-09-08'. This means it will return the names and phone numbers of the owners who have purchased an airplane after September 8th, 2020.

–Q9 This query selects the Liscence column from the Pilot table, and counts the values of the Model column from the FLIES table for each pilot. The query joins the Pilot table with the FLIES table based on the Liscence and Fliscence columns. The query then groups the results by the Liscence column. This means it will return the license numbers and number of planes flown by each pilot.

–Q10 This query selects the top one value of the HLocation column from the Hangar table, and subtracts the count of the RegNo column from the AirPlane table from the PlaneCapacity column of the Hangar table for each hangar. The query left joins the Hangar table with the AirPlane table based on the ID and

STORED_IN columns. The query then groups the results by the ID, HLocation, and PlaneCapacity columns, and orders them by the AvailableSpace column in descending order. This means it will return the location and available space of the hangar with the most empty slots for storing airplanes.

–Q11 This query selects the Cname column from the Coporation table, and counts the values of the Reg column from the OWNS table for each corporation. The query joins the Coporation table with the owners table based on a pattern match between the TIN and SSN columns. The query then joins the owners table with the OWNS table based on the SSN column. The query then groups the results by the Cname column and orders them by the NumberOfPlanes column in descending order. This means it will return the names and number of planes owned by each corporation, sorted by the highest number of planes.

–Q12 This query selects the ModelNo column from the Plane_Type table, and calculates the average value of the Hours column from the SERVICE table for each model. The query joins the Plane_Type table with the AirPlane table based on the ModelNo and OF_TYPE columns. The query then joins the AirPlane table with the PLANE_SERVICE table based on the RegNo and Reg columns. The query then joins the PLANE_SERVICE table with the SERVICE table based on the ServiceID column. The query then groups the results by the ModelNo column. This means it will return the model numbers and average maintenance hours of each plane type.

–Q13 This query selects the o_name column from the owners table, and joins it with four other tables: OWNS, AirPlane, PLANE_SERVICE, and SERVICE. The join conditions are based on the SSN and Reg columns. The query also joins the SERVICE table with the MAINTAIN table based on the ServiceID column, and joins the MAINTAIN table with the Employee table based on the EmployeeID column. The query then left joins the Employee table with the WORKS_ON table based on the EmployeeID and Model columns, and joins the WORKS_ON table with the AirPlane table based on the OF_TYPE and Model columns. The query then filters the results by using a WHERE clause that specifies that the EmployeeID column of the WORKS_ON table must be null. This means it will return the names of the owners whose airplanes have been serviced by employees who do not work on their plane types.

–Q14 This query selects the o_name and Phone columns from the owners table, and joins it with another table called Coporation based on the SSN and TIN columns. The query then joins the Coporation table with another table called OWNS based on the TIN and SSN columns. The query then joins the OWNS table with another table called AirPlane based on the Reg and RegNo columns. The query then joins the AirPlane table with another table called Hangar based on the STORED_IN and ID columns. The query then filters the results by using a WHERE clause that specifies that the Address1 column of the owners table must be equal to the HLocation column of the Hangar table. This means it will return the names and phone numbers of the owners who are corporations and store their airplanes in hangars located at their addresses.

–Q15 This query selects the Name column from the PERSON table, and joins it with another table called Pilot based on the SSN and PERSON2 columns. The query then joins the Pilot table with another table called FLIES based on the Liscence and Fliscence columns. The query then joins the FLIES table with another table called AirPlane based on the Model and OF_TYPE columns. The query then filters the results by using a WHERE clause that specifies that the UnderMaintenance column of the AirPlane table must be equal to 1. This means it will return the names of the pilots who have flown airplanes that are under maintenance

–Q16 This query selects the Name column from the PERSON table, and sums up the values of the Hours column from the SERVICE table for each person. The query joins the PERSON table with the Employee table based on the SSN and PERSON columns. The query then joins the Employee table with the MAINTAIN table based on the EmployeeID column. The query then joins the MAINTAIN table with the SERVICE table based on the ServiceID column. The query then joins the SERVICE table with the PLANE_SERVICE table based on the ServiceID column. The query then joins the PLANE_SERVICE table with the OWNS table based on the Reg column. The query then joins the OWNS table with the Coporation table based on the SSN and TIN columns. The query then filters the results by using a WHERE clause that specifies that the Cname column of the Coporation table must be equal to ‘Acme Corporation’. The query then groups the results by the Name column, and orders them by the TotalHours column in descending order. This means it will return the names and total hours of service of the employees who work for Acme Corporation and maintain their planes.

–Q17 This query selects the RegNo and ModelNo columns from the AirPlane table, and left joins it with four other tables: OWNS, Coporation, PLANE_SERVICE, and SERVICE. The join conditions are based on the RegNo and Reg columns, and the SSN and TIN columns. The query also left joins the SERVICE table with the MAINTAIN table based on the ServiceID column, and left joins the MAINTAIN table with the Employee table based on the EmployeeID column. The query then filters the results by using a WHERE clause that specifies that the Cname column of the Coporation table must be null, and either the Shift_Worked column of the Employee table must not be equal to ‘Day Shift’, or it must be null. The query then groups the results by the RegNo and ModelNo columns. This means it will return the registration numbers and model numbers of the airplanes that are not owned by any corporation, and have been serviced by employees who do not work in the day shift or have not been serviced at all.

–Q18 This query selects the owner name and address from the owners table for those owners who also own a corporation and have an airplane of the same type as another corporation owner who bought an airplane after 2020-01-08. It uses aliases (AS) to rename the tables and columns for convenience, joins (JOIN) to combine data from multiple tables based on matching conditions, and a subquery (EXISTS) to check for the existence of another record that satisfies the criteria.

–Q19 This query selects the hangar ID and the count of airplanes stored in each hangar from the hangar and airplane tables. It uses a left join (LEFT JOIN) to include all records from the hangar table even if they have no matching records in the airplane table, and a group by clause (GROUP BY) to group the records by hangar ID and apply an aggregate function (COUNT) to each group.

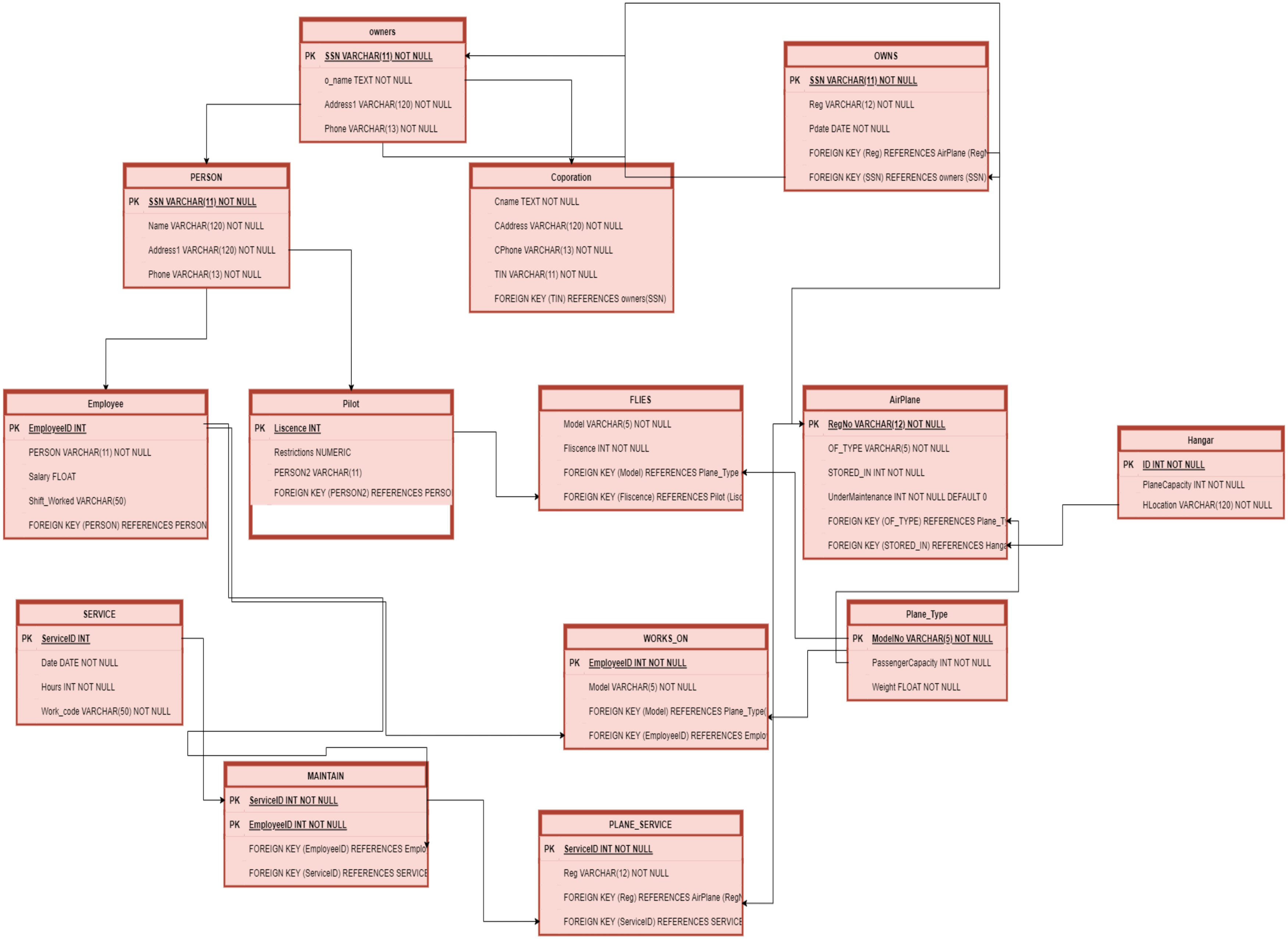
–Q20 This query selects the plane model number and the count of airplanes of each model type from the plane_type and airplane tables. It uses a left join (LEFT JOIN) to include all records from the plane_type table even if they have no matching records in the airplane table, and a group by clause (GROUP BY) to group the records by model number and apply an aggregate function (COUNT) to each group.

–Q21 This query selects the airplane registration number and the count of services performed on each airplane from the airplane, plane_service and service tables. It uses left joins (LEFT JOIN) to include all records from the airplane table even if they have no matching records in the other tables, and a group by clause (GROUP BY) to group the records by registration number and apply an aggregate function (COUNT) to each group.

–Q22 This query selects the shift worked and the average salary of employees for each shift from the employee table. It uses a group by clause (GROUP BY) to group the records by shift worked and apply an aggregate function (AVG) to each group.

–Q23 This query selects the social security number of owners and the count of airplanes owned by each owner from the owns table. It uses a group by clause (GROUP BY) to group the records by social security number and apply an aggregate function (COUNT) to each group.

–Q24 This query selects the social security number, name and count of authorized planes for each pilot from the pilot, person and flies tables. It uses joins (JOIN) to combine data from multiple tables based on matching conditions, and a group by clause (GROUP BY) to group the records by social security number and name and apply an aggregate function (COUNT) to each group.



owners
PK SSN VARCHAR(11) NOT NULL
o_name TEXT NOT NULL
Address1 VARCHAR(120) NOT NULL
Phone VARCHAR(13) NOT NULL

OWNS
PK SSN VARCHAR(11) NOT NULL
Reg VARCHAR(12) NOT NULL
Pdate DATE NOT NULL
FOREIGN KEY (Reg) REFERENCES AirPlane (RegN
FOREIGN KEY (SSN) REFERENCES owners (SSN)

PERSON
PK SSN VARCHAR(11) NOT NULL
Name VARCHAR(120) NOT NULL
Address1 VARCHAR(120) NOT NULL
Phone VARCHAR(13) NOT NULL

Coporation
Cname TEXT NOT NULL
CAddress VARCHAR(120) NOT NULL
CPhone VARCHAR(13) NOT NULL
TIN VARCHAR(11) NOT NULL
FOREIGN KEY (TIN) REFERENCES owners(SSN)

Employee
PK EmployeeID INT
PERSON VARCHAR(11) NOT NULL
Salary FLOAT
Shift_Worked VARCHAR(50)
FOREIGN KEY (PERSON) REFERENCES PERSON

Pilot
PK Liscence INT
Restrictions NUMERIC
PERSON2 VARCHAR(11)
FOREIGN KEY (PERSON2) REFERENCES PERSO

FLIES
Model VARCHAR(5) NOT NULL
Fliscence INT NOT NULL
FOREIGN KEY (Model) REFERENCES Plane_Type
FOREIGN KEY (Fliscence) REFERENCES Pilot (Lisc

AirPlane
PK RegNo VARCHAR(12) NOT NULL
OF_TYPE VARCHAR(5) NOT NULL
STORED_IN INT NOT NULL
UnderMaintenance INT NOT NULL DEFAULT 0
FOREIGN KEY (OF_TYPE) REFERENCES Plane_T
FOREIGN KEY (STORED_IN) REFERENCES Hanga

Hangar
PK ID INT NOT NULL
PlaneCapacity INT NOT NULL
HLocation VARCHAR(120) NOT NULL

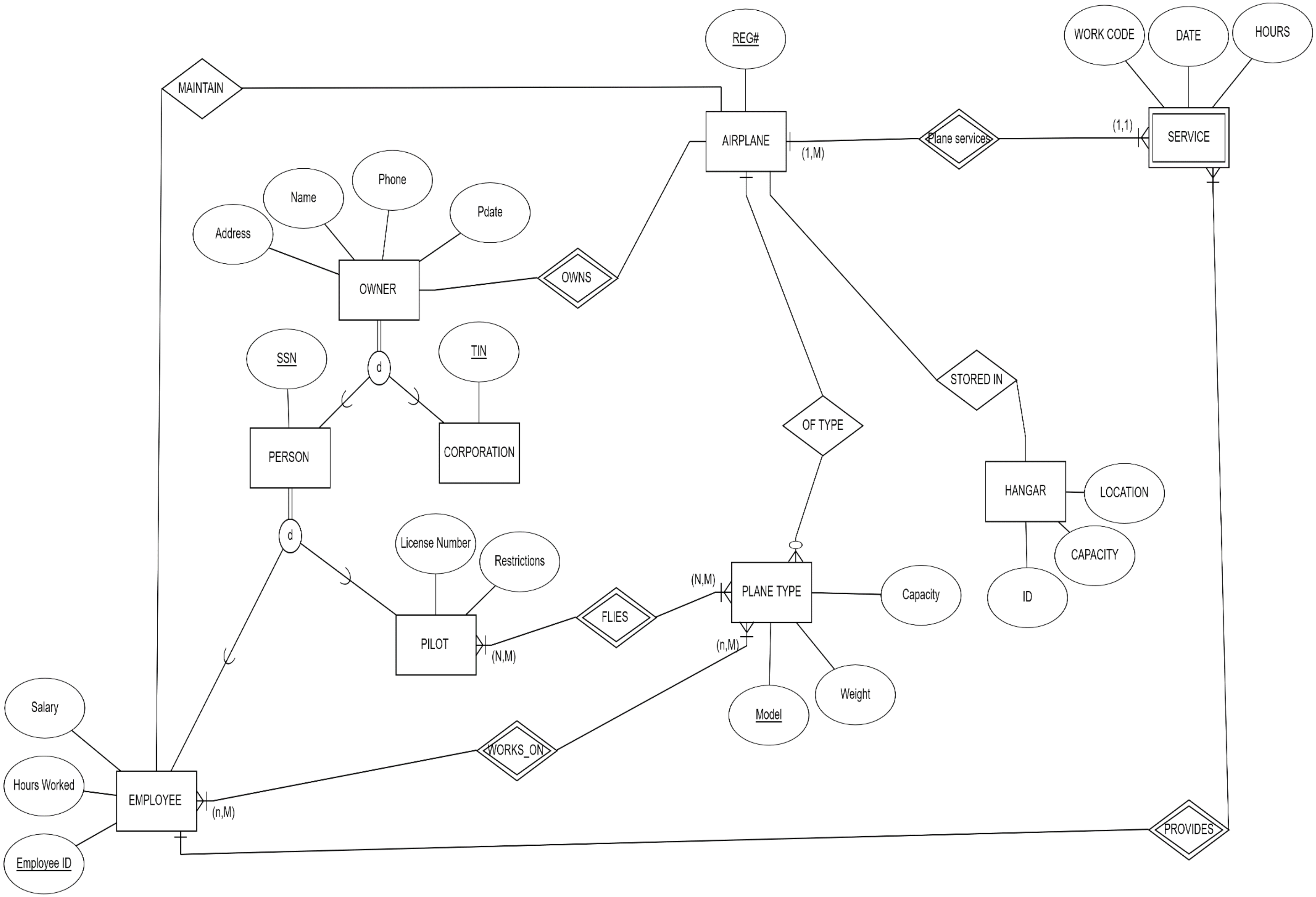
SERVICE
PK ServiceID INT
Date DATE NOT NULL
Hours INT NOT NULL
Work_code VARCHAR(50) NOT NULL

WORKS_ON
PK EmployeeID INT NOT NULL
Model VARCHAR(5) NOT NULL
FOREIGN KEY (Model) REFERENCES Plane_Type(
FOREIGN KEY (EmployeeID) REFERENCES Emplo

Plane_Type
PK ModelNo VARCHAR(5) NOT NULL
PassengerCapacity INT NOT NULL
Weight FLOAT NOT NULL

MAINTAIN
PK ServiceID INT NOT NULL
PK EmployeeID INT NOT NULL
FOREIGN KEY (EmployeeID) REFERENCES Emplo
FOREIGN KEY (ServiceID) REFERENCES SERVICE

PLANE_SERVICE
PK ServiceID INT NOT NULL
Reg VARCHAR(12) NOT NULL
FOREIGN KEY (Reg) REFERENCES AirPlane (RegN
FOREIGN KEY (ServiceID) REFERENCES SERVICE


MAINTAIN
REG#
WORK CODE
DATE
HOURS
AIRPLANE
(1,M)
Plane services
(1,1)
SERVICE
Name
Phone
Address
Pdate
OWNER
OWNS
SSN
d
TIN
PERSON
CORPORATION
OF TYPE
STORED IN
HANGAR
LOCATION
CAPACITY
ID
d
License Number
Restrictions
(N,M)
PLANE TYPE
Capacity
FLIES
PILOT
(N,M)
(n,M)
Weight
Model
Salary
Hours Worked
EMPLOYEE
(n,M)
WORKS_ON
Employee ID
PROVIDES