Design a relational database for the following Airport E-R diagram and write an SQL script (airport.sql) with all constraints similar to companyscript.sql.

```
create table Plane_Type

( model       varchar(15)    not null,

  capacity    int        not null,

  weight    char(9)        not null,

  constraint Ptypek

    primary key (model)

);


-- create a new empty table Airplane

create table Airplane

( reg        varchar(15)    not null,

model      varchar(15) not null,

Hang_number int    not null,

  constraint airpk

    primary key (reg),

constraint modelfk

    foreign key (model) references Plane_Type(model)

constraint Hangarfk

    foreign key (Hang_number) references Hangar(Hanq_Number)

);


-- create a new empty table dept_locations

create table owns

( airplane_reg    int        not null,

  owner_ssn   char(10)    not null,

  constraint owns_primary

    primary key (airplane_reg, owner_ssn),
```

```
  constraint owns_fk
    foreign key (airplane_reg) references airplane(reg)
    constraint owns_fk2
    foreign key (owner_ssn) references owner(ss)
);

-- create a new empty table Hangar
create table Hangar
( Hang_number   int      not null,
  Capacity    int      not null,
  location    varchar(15),
  dnum      int      not null,
  constraint hangarpk
    primary key (Hang_number)
);

-- create a new empty table works_on
create table works_on
( plane_Model      varchar(15)      not null,
  essn      char(10)      not null,
  constraint Planekey
    primary key (plane_Model,essn),
  constraint workprojfk
    foreign key (plane_Model) references Plane_Type(model)
      constraint workprojfk2
    foreign key (essn) references employee(ssn)
);
```

```
-- create a new empty table dependent
create table flies
( plane_Model     varchar(15)      not null,
  pilot_ssn    char(10)    not null,
  constraint deppk
    primary key (pilot_ssn, plane_Model),
  constraint depempfk
    foreign key (plane_Model) references employee(ssn)
      constraint depempfk2
    foreign key (pilot_ssn) references pilot(ssn)
);
create table Employee
( salary      int       not null,
  shift      varchar(10)       not null,
  ssn      char(10),
  address   varchar(25),
  phone      varchar(10),
  name      varchar(15),
  constraint employeepk
    primary key (ssn)
);
create table Pilot
( Restr      int       not null,
  Lic_Num      varchar(10)      not null,
  ssn      char(10),
  address   varchar(25),
  phone      varchar(10),
  name      varchar(15),
  constraint pilot_pk
```

```
  primary key (ssn)
);
create table Owner
(
 ssn      char(10)    not null,
 address   varchar(25),
 personal_phone      varchar(10),
 owner_name      varchar(15),
 Corporation_name          varchar(15) not null,

 constraint Owner_pk
  primary key (ssn,owner_name)
);
create table Service
(
 Date_code      varchar(9) not null,
 Woodcode   varchar(25) not null,
 hours      varchar(10),

 constraint Service_pk
  primary key (Date_code,Woodcode)
);
create table Maintain
(

  Date_code   varchar(9) not null,
  Woodcode   varchar(25) not null,
  employee_ssn char(10),
```

constraint Maintain_Pk

primary key (Date_code,Woodcode,employee_ssn)

);

2A. PRIMARY KEY

The closure of {A,B,D}

From depedency {A,B} {C},{B,D}{E,F} and {A,D}{G,H}

{A,B,D}+ {A,B,C,D,E,F,G,H}

Dependecy {A} {I} , add attribute I to {A,B,D}+

{A,B,D}+ {A,B,C,D,E,F,G,H,I}

Dependecy {H} {J} adds attribute J to {A,B.D}+

{A,D,C,D,E,F,G,H,I}

This represents T so the primary k is {A,B,D}+

2B DECOMPOSING INTO 2NF

PRIMARY KEY {A,B,D}

{A,B} partial key, functionally determines C

{B,D} partial key, functionally determines {E,F}

{A,D} partial key, functionally determines {G,H}

{A} partial key, functionally determines {I}

T1 {A,B,C} | | KEY: {A,B}

T2 {B,D,E,F} | | KEY: {B,D}

T3 {A,D,G,H,J} | | KEY: {A,D}

T4 {A,I} | | KEY: {A}

T5 {A,B,D} | | KEY {A,B,D}

2C DECOMPOSING INTO 3NF

H (a non attribute) functionallhy determines J

T6 {A,D,G,H} || KEY {A,D}

T7 {H,J} || KEY {H}

3NF IS ALSO IN 2NF, ADD T1->5

Set of relations on 3NF

T1 {A,B,C}

T2 {B,D,E,F}

T3 {A,D,G,H,J}

T4 {A,I}

T5 {A,B,D}

T6 {A,D,G,H}

T7 {H,J}

3A

The table is in 1NF.

We have a partial dependency on the Commission percent, it fully depends on the sales person and the car has nothing to do with the commision percent. The sales person is the one who fully decides the commision percent. Not in 2nf, hence not in 3NF.

I will normalize this by making the commision percent also depending on the car they are selling if we explicitly say that there is a dependency on both then we could have it in 2NF. We have no transative dependencies meaning we will be able to have it in 3nf.

Decomposing

SalesCommision(SalesPerson,CommsionPercent)

#4A

The relation is in 1NF but not in 2NF.

The primary keys are Book_title and Author_Name. The values are unique so it makes it 1NF

We have partial dependency in the relations

, Author_Name functionally determines Author_Affil. We do not need the Book title of the author to get the Author_ Affiliation

Book_title functionally determines Publisher and Book_type we do not need the authors name to know this either.

With part of the key fully determing the attribute, we ca not have it in 2NF.

#4B

By removing this partal relationships we can get the books in 2NF.

We decompose

By doing this

Book_Author (Book_title,authorName)

Book_publisher(Book_title,Publisher,Book_type and List price)

Author(Author_Name,Author_Affil)

Now we have no partial dependencies, meaning we have 2NF.