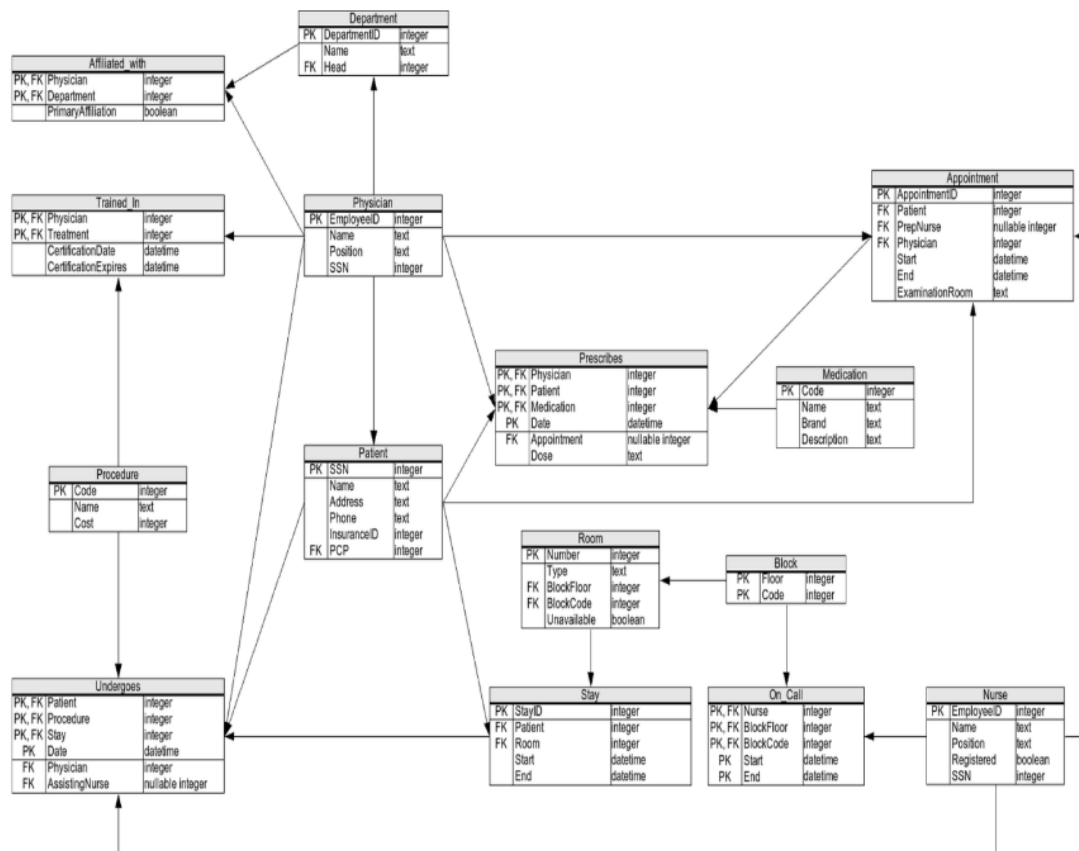


Questions:-



```

1 • Create database dbslab8;
2 • use dbslab8;
3
4 • create table Physician (
5     EmployeeID integer primary key not null,
6     Name text,
7     Position text,
8     SSN integer
9
10 );
    
```

```
12 • ○ create table Department (  
13     DepartmentID integer primary key not null,  
14     Name text,  
15     Head integer,  
16     foreign key (Head) references Physician(EmployeeID)  
17 );  
18  
19 • ○ create table Affiliated_with (  
20     Physician integer not null,  
21     Department integer not null,  
22     PrimaryAffiliation boolean,  
23     primary key(Physician , Department),  
24     foreign key (Physician) references Physician(EmployeeID),  
25     foreign key (Department) references Department(DepartmentID)  
26 );  
27  
28 • ○ create table Procedures (  
29     Code integer primary key not null,  
30     Name Varchar(30),  
31     Cost integer  
32 );  
33  
34 • ○ create table Trained_In (  
35     Physician integer not null,  
36     Treatment integer not null,  
37     certificateDate DATETIME ,  
38     CertifiacteExpires datetime,  
39     primary key(Physician , Treatment).
```

```

3     primary key(Physician , Treatment),
3     foreign key (Physician) references Physician(EmployeeID),
1     foreign key (Treatment) references Procedures(Code)
2 );
3
4 • ○ create table Patient (
5     SSN integer primary key not null,
5     name Varchar(30),
7     Address varchar(100),
3     Phone text,
3     InsuranceID integer ,
3     PCP integer ,
1     foreign key (PCP) references Physician(EmployeeID)
2 );
3
4 • ○ create table Undergoes (
5     Patient integer not null,
5     Procedures integer not null,
7     Stay integer not null,
3     Date datetime not null,
3     Physician integer,
3     AssistingNurse integer,
1     primary key(Patient , Procedures , Stay, Date),
2     foreign key (Patient) references Patient(SSN),
3     foreign key (Procedures) references Procedures(Code),
4     foreign key (Stay) references Stay(StayID),
5     foreign key (Physician) references Physician(EmployeeID),
5     foreign key (AssistingNurse) references Nurse(employeeID)

```

```
66     foreign key (AssistingNurse) references Nurse(employeeID)
67 );
68
69 • ⊖ create table Nurse (
70     EmployeeID integer primary key not null,
71     Name varchar(30),
72     Position varchar(100),
73     Registered boolean,
74     SSN integer
75 );
76
77 • ⊖ create table Block (
78     Floor integer primary key not null,
79     Code integer primary key not null
80 );
81
82 • ⊖ create table Room (
83     Number integer not null primary key,
84     Type varchar(100),
85     BlockFloor integer ,
86     BlockCode integer,
87     Unavailable boolean,
88     foreign key (BlockFloor) references Block(Floor),
89     foreign key (BlockCode) references Block(Code)
90 );
91
```

```
92 • ⊖ create table On_Call (  
93     Nurse integer,  
94     BlockFloor integer ,  
95     BlockCode integer,  
96     Start datetime,  
97     End datetime,  
98     primary key(Nurse , BlockFloor , BlockCode , Start , End),  
99     foreign key (Nurse) references Nurse (EmployeeID),  
100    foreign key (BlockFloor) references Block(Floor),  
101    foreign key (BlockCode) references Block(Code)  
102  
103    );  
104  
105 • ⊖ create table Stay (  
106     StayID integer primary key not null,  
107     Patient integer,  
108     Room integer ,  
109     Start datetime,  
110     End datetime,  
111     foreign key (Patient) references Patient(SSN),  
112     foreign key (Room) references Room(Number)  
113    );  
114  
115 • ⊖ create table Medication (  
116     Code integer primary key not null,  
117     Name varchar(30),  
118     Brand varchar(50),
```

```

9      Description varchar(100)
0  );
1
2  ● ○ create table Appointment (
3      AppointmentID integer primary key not null,
4      Patient integer,
5      PrepNurse integer ,
6      Physician integer,
7      Start datetime,
8      End datetime,
9      ExaminationRoom varchar(30),
0      foreign key (Patient) references Patient(SSN),
1      foreign key (PrepNurse) references Nurse(EmployeeID),
2      foreign key (Physician) references Physician(EmployeeID)
3  );
4
5  ● ○ create table Prescribes (
6      Physician integer ,
7      Patient integer,
8      Medication integer,
9      Date datetime,
0      Appointment integer,
1      Dose Varchar(30),
2      primary key ( Physician , Patient , Medication , Date),
3      foreign key ( Physician) references Physician(EmployeeID),
4      foreign key (Patient) references Patient(SSN),
5      foreign key (Medication) references Medication(Code),
6      foreign key (Appointment) references Appointment(AppointmentID)

```

Above was schema

**8.1 Obtain the names of all physicians that have performed a medical procedure they have never been certified to perform.**

```
309 • SELECT Name
310     FROM Physician
311     WHERE EmployeeID IN
312         (
313             SELECT Physician FROM Undergoes U WHERE NOT EXISTS
314                 (
315                     SELECT * FROM Trained_In
316                     WHERE Treatment = Procedures
```

Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

	Name
▶	Christopher Turk

**8.2 Same as the previous query, but include the following information in the results: Physician name, name of procedure, date when the procedure was carried out, name of the patient the procedure was carried out on.**

```

317 • SELECT P.Name AS Physician, Pr.Name AS Procedures,
318      Pt.Name AS Patient
319      FROM Physician P, Undergoes U, Patient Pt, Procedures Pr
320      WHERE U.Patient = Pt.SSN
321            AND U.Procedures = Pr.Code
322            AND U.Physician = P.EmployeeID
323            AND NOT EXISTS
324                (
325                  SELECT * FROM Trained_In T
326                  WHERE T.Treatment = U.Procedures
327                        AND T.Physician = U.Physician
328                );

```

Result Grid |  Filter Rows:  | Export:  Wrap Cell Content: 

	Physician	Procedures	Patient
▶	Christopher Turk	Complete Walleectomy	Dennis Doe

**8.3 Obtain the names of all physicians that have performed a medical procedure that they are certified to perform, but such that the procedure was done at a date (Undergoes.Date) after the physician's certification expired (Trained\_In.CertificationExpires).**



```

329 • SELECT P.Name FROM
330     Physician AS P,
331     Trained_In T,
332     Undergoes AS U
333     WHERE T.Physician=U.Physician AND T.Treatment=U.Procedures
334     AND U.DateUndergoes>T.CertificationExpires AND
335     P.EmployeeID=U.Physician

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Name			
Todd Quinlan			


**8.4** Same as the previous query, but include the following information in the results: Physician name, name of procedure, date when the procedure was carried out, name of the patient the procedure was carried out on, and date when the certification expired.

```

336 • SELECT P.Name AS Physician, Pr.Name AS Procedures, U.DateUndergoes,
337     Pt.Name AS Patient, T.CertificationExpires
338     FROM Physician P, Undergoes U, Patient Pt, Procedures Pr,
339     Trained_In T
340     WHERE U.Patient = Pt.SSN
341     AND U.Procedures = Pr.Code
342     AND U.Physician = P.EmployeeID
343     AND Pr.Code = T.Treatment
344     AND P.EmployeeID = T.Physician
345     AND U.DateUndergoes > T.CertificationExpires;


```

Result Grid




Filter Rows:

Export:



Wrap Cell Content:



	Physician	Procedures	DateUndergoes	Patient	CertificationExpires
▶	Todd Quinlan	Obfuscated Dermogastrotomy	2008-05-09 00:00:00	Dennis Doe	2007-12-31 00:00:00

**8.5 Obtain the information for appointments where a patient met with a physician other than his/her primary care physician. Show the following information: Patient name, physician name, nurse name (if any), start and end time of appointment, examination room, and the name of the patient's primary care physician.**

```

346 • SELECT Pt.Name AS Patient, Ph.Name AS Physician, N.Name AS Nurse,
347       A.Endo, A.ExaminationRoom, PhPCP.Name AS PCP
348   FROM Patient Pt, Physician Ph, Physician PhPCP,
349        Appointment A LEFT JOIN Nurse N ON A.PrepareNurse = N.EmployeeID
350  WHERE A.Patient = Pt.SSN
351        AND A.Physician = Ph.EmployeeID
352        AND Pt.PCP = PhPCP.EmployeeID
353        AND A.Physician <> Pt.PCP;

```

Result Grid						
		Filter Rows:	Export:		Wrap Cell Content:	
	Patient	Physician	Nurse	Endo	ExaminationRoom	PCP
▶	John Smith	Christopher Turk	NULL	2008-04-26 13:00:00	C	John Dorian
	Dennis Doe	Percival Cox	Paul Flowers	2008-04-25 11:00:00	B	Christopher Turk
	Dennis Doe	Percival Cox	NULL	2008-04-26 11:00:00	C	Christopher Turk
	Dennis Doe	Molly Clock	Laverne Roberts	2008-04-21 11:00:00	A	Christopher Turk

**8.6 The Patient field in Undergoes is redundant, since we can obtain it from the Stay table. There are no constraints in force to prevent inconsistencies between these two tables. More specifically, the Undergoes table may include a row where the patient ID does not match the one we would obtain from the Stay table through the Undergoes.Stay foreign key. Select all rows from Undergoes that exhibit this inconsistency.**

```

354 • SELECT * FROM Undergoes U
355     WHERE Patient <>
356     (
357         SELECT Patient FROM Stay S
358         WHERE U.Stay = S.StayID
359     );

```

Result Grid    Filter Rows: <input type="text"/>   Edit:      Export/Import:						
	Patient	Procedures	Stay	DateUndergoes	Physician	AssistingNurse
▶	100000001	7	3217	2008-05-10 00:00:00	7	101
•	NULL	NULL	NULL	NULL	NULL	NULL

8.7 Obtain the names of all the nurses who have ever been on call for room 123.

```

360 • SELECT N.Name FROM Nurse N
361     WHERE EmployeeID IN
362     (
363         SELECT OC.Nurse FROM On_Call OC, Room R
364         WHERE OC.BlockFloor = R.BlockFloor
365             AND OC.BlockCode = R.BlockCode
366             AND R.RoomNumber = 123

```

Result Grid    Filter Rows: <input type="text"/>   Export:    Wri	
	Name
▶	Laverne Roberts
	Paul Flowers

**8.8 The hospital has several examination rooms where appointments take place. Obtain the number of appointments that have taken place in each examination room.**

```
367 • SELECT ExaminationRoom, COUNT(AppointmentID) AS Number
368 FROM Appointment
369 GROUP BY ExaminationRoom;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	ExaminationRoom	Number			
▶	A	3			
	B	3			
	C	3			

**8.9 Obtain the names of all patients (also include, for each patient, the name of the patient's primary care physician), such that the following are true:**

**The patient has been prescribed some medication by his/her primary care physician.**

**The patient has undergone a procedure with a cost larger than \$5,000**


**The patient has had at least two appointments where the nurse who prepped the appointment was a registered nurse.**

**The patient's primary care physician is not the head of any department.**

```

370 • SELECT Pt.Name, PhPCP.Name FROM Patient Pt, Physician PhPCP
371     WHERE Pt.PCP = PhPCP.EmployeeID
372     AND EXISTS
373     (
374         SELECT * FROM Prescribes Pr
375         WHERE Pr.Patient = Pt.SSN
376             AND Pr.Physician = Pt.PCP
377     )
378     AND EXISTS
379     (
380         SELECT * FROM Undergoes U, Procedures Pr
381         WHERE U.Procedures = Pr.Code
382             AND U.Patient = Pt.SSN
383             AND Pr.Cost > 5000
384     )
385     AND 2 <=
386     (
387         SELECT COUNT(A.AppointmentID) FROM Appointment A, Nurse N
388         WHERE A.PrepareNurse = N.EmployeeID
389             AND N.Registered = 1
390     )
391     AND NOT Pt.IsPCP

```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	Name	Name
▶	John Smith	John Dorian