# Database Design For Hospital Entity:
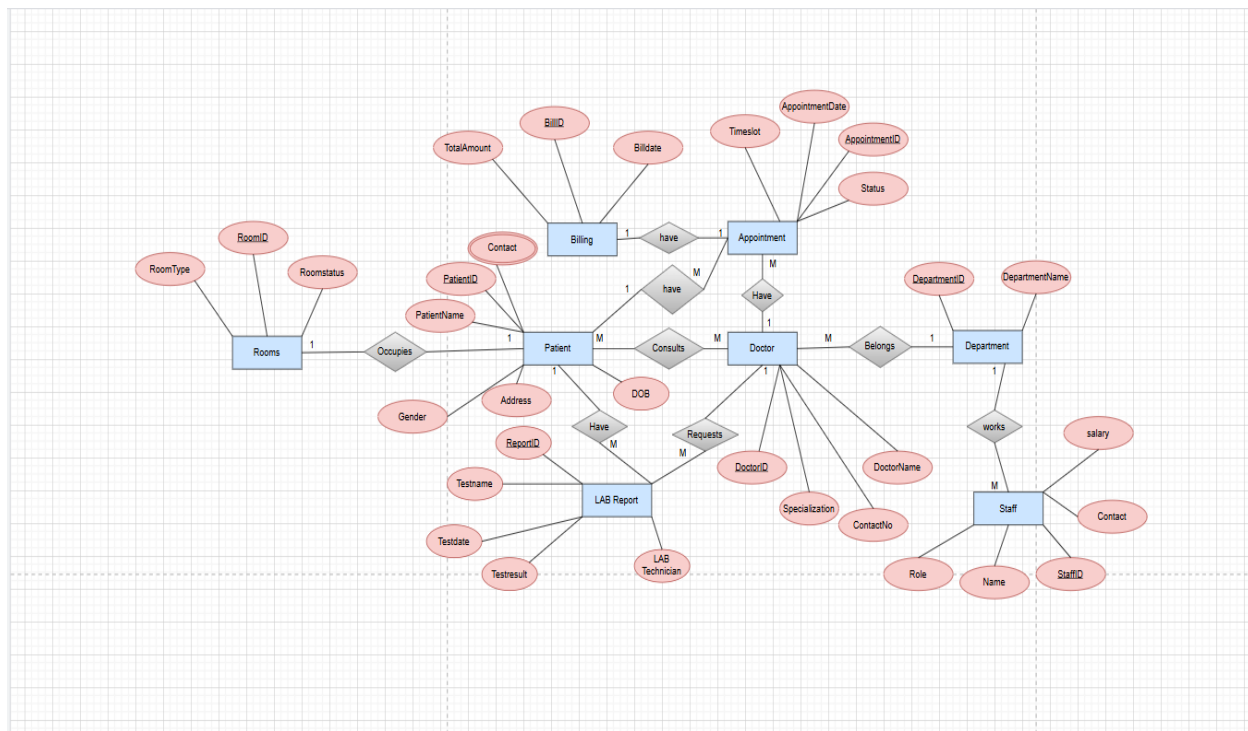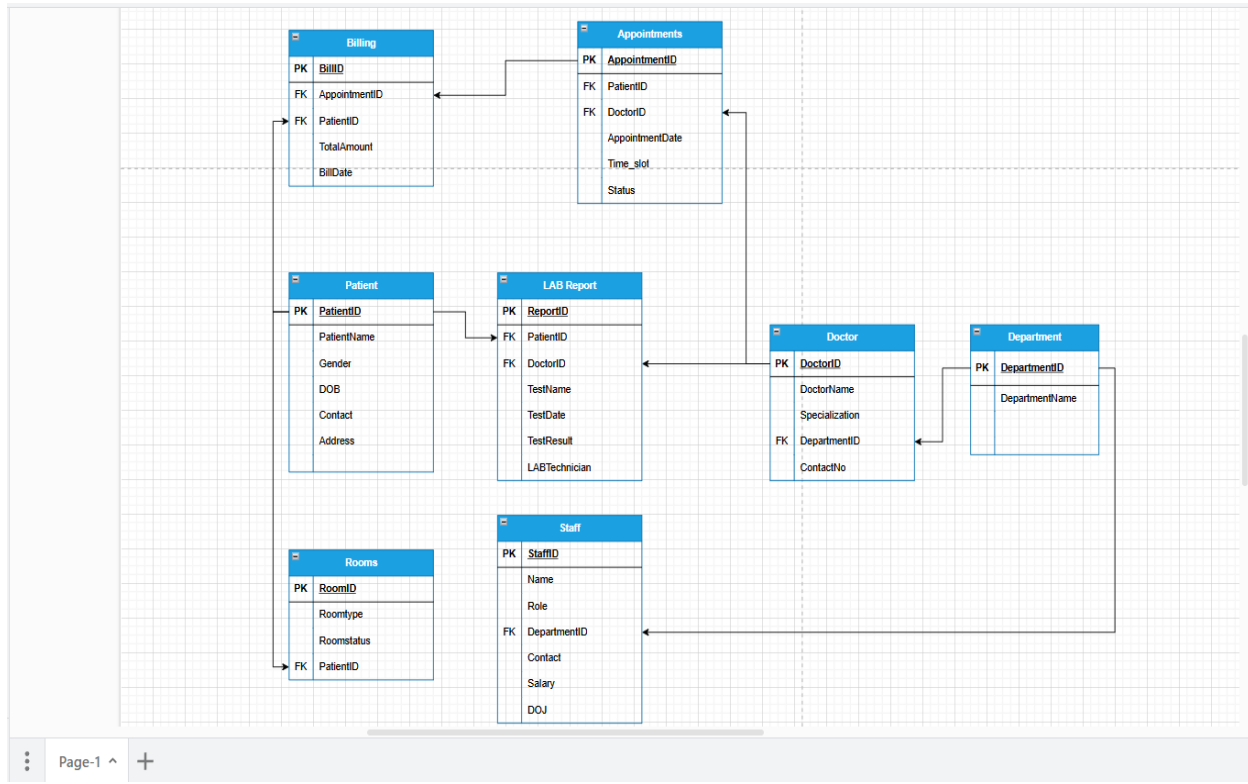
Some of the SQL Queries To Handle Hospital Entity

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ►

```
121    INSERT INTO Staff VALUES (3, 'Sarah Wilson', 'Receptionist', 3, '4321098765', 20000.00, TO_DATE('2023-05-05', 'YYYY-MM-DD'));
122    -- Retrieving All Appointments for a Specific Doctor
123    SELECT
124        a.AppointmentID, p.PatientName, a.AppointmentDate, a.TimeSlot, a.Status
125    FROM
126        Appointments a
127    JOIN
128        Patient p ON a.PatientID = p.PatientID
129    WHERE
130        a.DoctorID = 1;
131    ---------Total Revenue-------
```

APPOINTMENTID	PATIENTNAME	APPOINTMENTDATE	TIMESLOT	STATUS
1	John Doe	01-DEC-24	10:00 AM	Confirmed

Download CSV

TOTALREVENUE
5000

Download CSV

no data found

2024 Oracle · Live SQL 24.4.1. running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

SQL Worksheet   Clear   Find   Actions ∨   Save   Run ►

```
145    -------creating a view for active patients---------
146    CREATE VIEW ActivePatients AS
147    SELECT
148        r.RoomID, r.RoomType, r.RoomStatus, p.PatientName
149    FROM
150        Rooms r
151    JOIN
152        Patient p ON r.PatientID = p.PatientID
153    WHERE
154        r.RoomStatus = 'Occupied';
155    ---------Window function Rank --------Ranking Patients by total bills paid---------
156    SELECT
157        p.PatientID,
158        p.PatientName,
159        SUM(b.TotalAmount) AS TotalBill,
160        RANK() OVER (ORDER BY SUM(b.TotalAmount) DESC) AS RankBySpending
161    FROM
```

View created.

PATIENTID	PATIENTNAME	TOTALBILL	RANKBYSPENDING
2	Jane Smith	2500	1
1	John Doe	1500	2
3	Alice Johnson	1000	3

2024 Oracle · Live SQL 24.4.1. running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

SQL Worksheet   ◇ Clear   ⚲ Find   Actions ▾   💾 Save   Run ▶

```
166        p.PatientID, p.PatientName;
167  ----------------Calculate Running Total of Revenue by Date--------------
168  SELECT
169      BillDate,
170      SUM(TotalAmount) AS DailyRevenue,
171      SUM(SUM(TotalAmount)) OVER (ORDER BY BillDate) AS RunningTotalRevenue
172  FROM
173      Billing
174  GROUP BY
175      BillDate;
176  -----------------------------Find Departments with the Most Doctors--------------------
```

BILLDATE	DAILYREVENUE	RUNNINGTOTALREVENUE
01-DEC-24	1500	1500
02-DEC-24	2500	4000
03-DEC-24	1000	5000

Download CSV

3 rows selected.

2024 Oracle · Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf · 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym

Built with ❤ using Oracle APEX · Privacy · Terms of Use

SQL Worksheet   ◇ Clear   ⚲ Find   Actions ▾   💾 Save   Run ▶

```
175      BillDate;
176  -----------------------------Find Departments with the Most Doctors--------------------
177  WITH DoctorCounts AS (
178      SELECT
179          d.DepartmentID,
180          dp.DepartmentName,
181          COUNT(d.DoctorID) AS DoctorCount
182      FROM
183          Doctor d
184      JOIN
185          Departments dp ON d.DepartmentID = dp.DepartmentID
186      GROUP BY
187          d.DepartmentID, dp.DepartmentName
188  )
189  SELECT
190      DepartmentName,
191      DoctorCount
192  FROM
```

DEPARTMENTNAME	DOCTORCOUNT
Neurology	1
Cardiology	1
Orthopedics	1

2024 Oracle · Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf · 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym

Built with ❤ using Oracle APEX · Privacy · Terms of Use

Live SQL

Feedback Help sairajkumar2002@gmail.com

SQL Worksheet

Clear Find Actions Save Run

```
195        DoctorCount = (SELECT MAX(DoctorCount) FROM DoctorCounts);
196  --------------------------Average Bill by Department Using Partition------------
197  SELECT
198        dp.DepartmentName,
199        p.PatientName,
200        b.TotalAmount,
201        AVG(b.TotalAmount) OVER (PARTITION BY dp.DepartmentName) AS AvgBillByDepartment
202  FROM
203        Billing b
204  JOIN
205        Appointments a ON b.AppointmentID = a.AppointmentID
206  JOIN
207        Doctor d ON a.DoctorID = d.DoctorID
208  JOIN
209        Departments dp ON d.DepartmentID = dp.DepartmentID
210  JOIN
211        Patient p ON b.PatientID = p.PatientID;
212  -----------Store Active Patients Temporarily----------------------
```

DEPARTMENTNAME	PATIENTNAME	TOTALAMOUNT	AVGBILLBYDEPARTMENT
Cardiology	John Doe	1500	1500
Neurology	Jane Smith	2500	2500
Orthopedics	Alice Johnson	1000	1000

2024 Oracle · Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym

Built with ❤ using Oracle APEX · Privacy · Terms of Use

CODE:

Live SQL

Feedback Help sairajkumar2002@gmail.com

SQL Worksheet

Clear Find Actions Save Run

```
1    -- Patient Table
2    CREATE TABLE Patient (
3        PatientID NUMBER PRIMARY KEY,
4        PatientName VARCHAR2(100),
5        Gender VARCHAR2(10),
6        DOB DATE,
7        Contact VARCHAR2(15),
8        Address VARCHAR2(255)
9    );
10   -- Departments Table
11   CREATE TABLE Departments (
12       DepartmentID NUMBER PRIMARY KEY,
13       DepartmentName VARCHAR2(100)
14   );
15
16   -- Doctor Table
17   CREATE TABLE Doctor (
18       DoctorID NUMBER PRIMARY KEY,
19       DoctorName VARCHAR2(100),
20       Specialization VARCHAR2(50),
21       DepartmentID NUMBER,
22       ContactNo VARCHAR2(15),
23       CONSTRAINT fk_DepartmentID FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
24   );
25
26   -- Rooms Table
27   CREATE TABLE Rooms (
28       RoomID NUMBER PRIMARY KEY,
29       RoomType VARCHAR2(50),
30       RoomStatus VARCHAR2(50),
31       PatientID NUMBER,
32       CONSTRAINT fk_PatientID FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)
33   );
34
35   -- Appointments Table
```

TOTALREVENUE

2024 Oracle · Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym

Built with ❤ using Oracle APEX · Privacy · Terms of Use

```sql
34
35    -- Appointments Table
36  , CREATE TABLE Appointments (
37      AppointmentID NUMBER PRIMARY KEY,
38      PatientID NUMBER,
39      DoctorID NUMBER,
40      AppointmentDate DATE,
41      TimeSlot VARCHAR2(50),
42      Status VARCHAR2(50),
43      CONSTRAINT fk_PatientID_App FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
44      CONSTRAINT fk_DoctorID_App FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)
45  );
46
47    -- Billing Table
48  , CREATE TABLE Billing (
49      BillID NUMBER PRIMARY KEY,
50      PatientID NUMBER,
51      AppointmentID NUMBER,
52      TotalAmount NUMBER(10, 2),
53      BillDate DATE,
54      CONSTRAINT fk_PatientID_Bill FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
55      CONSTRAINT fk_AppointmentID FOREIGN KEY (AppointmentID) REFERENCES Appointments(AppointmentID)
56  );
57
58    -- Lab Report Table
59  , CREATE TABLE LabReport (
60      ReportID NUMBER PRIMARY KEY,
61      PatientID NUMBER,
62      DoctorID NUMBER,
63      TestName VARCHAR2(100),
64      TestDate DATE,
65      TestResult VARCHAR2(255),
66      LabTechnician VARCHAR2(100),
67      CONSTRAINT fk_PatientID_Lab FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
68      CONSTRAINT fk_DoctorID_Lab FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)
```

```sql
67      CONSTRAINT fk_PatientID_Lab FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
68      CONSTRAINT fk_DoctorID_Lab FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)
69  );
70
71    -- Staff Table
72  , CREATE TABLE Staff (
73      StaffID NUMBER PRIMARY KEY,
74      StaffName VARCHAR2(100),
75      Role VARCHAR2(50),
76      DepartmentID NUMBER,
77      Contact VARCHAR2(15),
78      Salary NUMBER(10, 2),
79      DOJ DATE,
80      CONSTRAINT fk_DepartmentID_Staff FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
81  );
82    -- Insert Data into Patient Table
83  INSERT INTO Patient VALUES (1, 'John Doe', 'Male', TO_DATE('1985-06-15', 'YYYY-MM-DD'), '1234567890', '123 Main St');
84  INSERT INTO Patient VALUES (2, 'Jane Smith', 'Female', TO_DATE('1990-08-22', 'YYYY-MM-DD'), '0987654321', '456 Elm St');
85  INSERT INTO Patient VALUES (3, 'Alice Johnson', 'Female', TO_DATE('2000-11-01', 'YYYY-MM-DD'), '5678901234', '789 Pine St');
86
87    -- Insert Data into Departments Table
88  INSERT INTO Departments VALUES (1, 'Cardiology');
89  INSERT INTO Departments VALUES (2, 'Neurology');
90  INSERT INTO Departments VALUES (3, 'Orthopedics');
91
92    -- Insert Data into Doctor Table
93  INSERT INTO Doctor VALUES (1, 'Dr. Brown', 'Cardiologist', 1, '9876543210');
94  INSERT INTO Doctor VALUES (2, 'Dr. Green', 'Neurologist', 2, '8765432109');
95  INSERT INTO Doctor VALUES (3, 'Dr. White', 'Orthopedic', 3, '7654321098');
96
97    -- Insert Data into Rooms Table
98  INSERT INTO Rooms VALUES (101, 'Single', 'Occupied', 1);
99  INSERT INTO Rooms VALUES (102, 'Double', 'Available', NULL);
100 INSERT INTO Rooms VALUES (103, 'ICU', 'Occupied', 2);
101
```

TOTALREVENUE

```sql
99  INSERT INTO Rooms VALUES (102, 'Double', 'Available', NULL);
100 INSERT INTO Rooms VALUES (103, 'ICU', 'Occupied', 2);
101
102 -- Insert Data into Appointments Table
103 INSERT INTO Appointments VALUES (1, 1, 1, TO_DATE('2024-12-01', 'YYYY-MM-DD'), '10:00 AM', 'Confirmed');
104 INSERT INTO Appointments VALUES (2, 2, 2, TO_DATE('2024-12-02', 'YYYY-MM-DD'), '11:30 AM', 'Pending');
105 INSERT INTO Appointments VALUES (3, 3, 3, TO_DATE('2024-12-03', 'YYYY-MM-DD'), '09:00 AM', 'Cancelled');
106
107 -- Insert Data into Billing Table
108 INSERT INTO Billing VALUES (1, 1, 1, 1500.00, TO_DATE('2024-12-01', 'YYYY-MM-DD'));
109 INSERT INTO Billing VALUES (2, 2, 2500.00, TO_DATE('2024-12-02', 'YYYY-MM-DD'));
110 INSERT INTO Billing VALUES (3, 3, 3, 1000.00, TO_DATE('2024-12-03', 'YYYY-MM-DD'));
111
112 -- Insert Data into LabReport Table
113 INSERT INTO LabReport VALUES (1, 1, 1, 'Blood Test', TO_DATE('2024-11-28', 'YYYY-MM-DD'), 'Normal', 'Technician A');
114 INSERT INTO LabReport VALUES (2, 2, 2, 'MRI', TO_DATE('2024-11-29', 'YYYY-MM-DD'), 'Abnormal', 'Technician B');
115 INSERT INTO LabReport VALUES (3, 3, 3, 'X-Ray', TO_DATE('2024-11-30', 'YYYY-MM-DD'), 'Normal', 'Technician C');
116
117 -- Insert Data into Staff Table
118 INSERT INTO Staff VALUES (1, 'Emily Davis', 'Nurse', 1, '6543210987', 30000.00, TO_DATE('2023-01-15', 'YYYY-MM-DD'));
119 INSERT INTO Staff VALUES (2, 'Michael Brown', 'Technician', 2, '5432109876', 25000.00, TO_DATE('2023-03-10', 'YYYY-MM-DD'));
120 INSERT INTO Staff VALUES (3, 'Sarah Wilson', 'Receptionist', 3, '4321098765', 20000.00, TO_DATE('2023-05-05', 'YYYY-MM-DD'));
121 -- Retrieving All Appointments for a Specific Doctor
122 SELECT
123     a.AppointmentID, p.PatientName, a.AppointmentDate, a.TimeSlot, a.Status
124 FROM
125     Appointments a
126 JOIN
127     Patient p ON a.PatientID = p.PatientID
128 WHERE
129     a.DoctorID = 1;
130 ---------Total Revenue-----
131 SELECT
132     SUM(TotalAmount) AS TotalRevenue
133 FROM
```

TOTALREVENUE

```sql
131 SELECT
132     SUM(TotalAmount) AS TotalRevenue
133 FROM
134     Billing;
135 ------------Doctors without Appointments---------
136 SELECT
137     d.DoctorName
138 FROM
139     Doctor d
140 LEFT JOIN
141     Appointments a ON d.DoctorID = a.DoctorID
142 WHERE
143     a.DoctorID IS NULL;
144 -------creating a view for active patients---------
145 CREATE VIEW ActivePatients AS
146 SELECT
147     r.RoomID, r.RoomType, r.RoomStatus, p.PatientName
148 FROM
149     Rooms r
150 JOIN
151     Patient p ON r.PatientID = p.PatientID
152 WHERE
153     r.RoomStatus = 'Occupied';
154 ---------Window function Rank -------Ranking Patients by total bills paid---------
155 SELECT
156     p.PatientID,
157     p.PatientName,
158     SUM(b.TotalAmount) AS TotalBill,
159     RANK() OVER (ORDER BY SUM(b.TotalAmount) DESC) AS RankBySpending
160 FROM
161     Patient p
162 JOIN
163     Billing b ON p.PatientID = b.PatientID
164 GROUP BY
165     p.PatientID, p.PatientName;
166     ------Calculate Running Total of Revenue by Date
```

TOTALREVENUE

```sql
        Patient p
    JOIN
        Billing b ON p.PatientID = b.PatientID
GROUP BY
    p.PatientID, p.PatientName;
----------------Calculate Running Total of Revenue by Date--------------
SELECT
    BillDate,
    SUM(TotalAmount) AS DailyRevenue,
    SUM(SUM(TotalAmount)) OVER (ORDER BY BillDate) AS RunningTotalRevenue
FROM
    Billing
GROUP BY
    BillDate;
-----------------------------Find Departments with the Most Doctors------------------
WITH DoctorCounts AS (
    SELECT
        d.DepartmentID,
        dp.DepartmentName,
        COUNT(d.DoctorID) AS DoctorCount
    FROM
        Doctor d
    JOIN
        Departments dp ON d.DepartmentID = dp.DepartmentID
    GROUP BY
        d.DepartmentID, dp.DepartmentName)
SELECT
    DepartmentName,
    DoctorCount
FROM
    DoctorCounts
WHERE
    DoctorCount = (SELECT MAX(DoctorCount) FROM DoctorCounts);
-----------------------------Average Bill by Department Using Partition-----------
SELECT
```

```sql
WITH DoctorCounts AS (
    SELECT
        d.DepartmentID,
        dp.DepartmentName,
        COUNT(d.DoctorID) AS DoctorCount
    FROM
        Doctor d
    JOIN
        Departments dp ON d.DepartmentID = dp.DepartmentID
    GROUP BY
        d.DepartmentID, dp.DepartmentName)
SELECT
    DepartmentName,
    DoctorCount
FROM
    DoctorCounts
WHERE
    DoctorCount = (SELECT MAX(DoctorCount) FROM DoctorCounts);
-----------------------------Average Bill by Department Using Partition-----------
SELECT
    dp.DepartmentName,
    p.PatientName,
    b.TotalAmount,
    AVG(b.TotalAmount) OVER (PARTITION BY dp.DepartmentName) AS AvgBillByDepartment
FROM
    Billing b
JOIN
    Appointments a ON b.AppointmentID = a.AppointmentID
JOIN
    Doctor d ON a.DoctorID = d.DoctorID
JOIN
    Departments dp ON d.DepartmentID = dp.DepartmentID
JOIN
    Patient p ON b.PatientID = p.PatientID;
```

TOTALREVENUE