

-- hospital management

USE hospitaldb;

-- 1. Generate a report: PatientName, DoctorName, DiseaseName

```
SELECT
    P.PatientName,
    D.DoctorName,
    DI.DiseaseName
FROM Allocation A
JOIN Patients P ON A.PatientID = P.PatientID
JOIN Doctors D ON A.DoctorID = D.DoctorID
JOIN Diseases DI ON A.DiseaseID = DI.DiseaseID;
```

-- 2. Generate a report: Hospital Name, Number of doctors

```
SELECT
    H.HospitalName,
    COUNT(D.DoctorID) AS NumberOfDoctors
FROM Hospitals H
LEFT JOIN Doctors D ON H.HospitalID = D.HospitalID
GROUP BY H.HospitalName;
```

-- 3. Generate a report: Hospital Name, Doctor Name, Patient Name

```
SELECT
    H.HospitalName,
    D.DoctorName,
    P.PatientName
FROM Allocation A
JOIN Doctors D ON A.DoctorID = D.DoctorID
JOIN Patients P ON A.PatientID = P.PatientID
JOIN Hospitals H ON D.HospitalID = H.HospitalID;
```

-- 4. Using ORDER BY clause: Display doctor details with respect to salary (descending)

```
SELECT
    DoctorName,
    Salary
FROM Doctors
ORDER BY Salary DESC;
```

-- 5. Select the patient details who has paid maximum bills so far.

```
SELECT
    PatientID,
    PatientName,
    DOB,
    PhoneNo,
    ArealD,
    TotalBill
FROM (
    SELECT
        P.PatientID,
        P.PatientName,
        P.DOB,
        P.PhoneNo,
        P.ArealD,
        SUM(A.FeesPerDay * A.NoOfDays) AS TotalBill,
        RANK() OVER (ORDER BY SUM(A.FeesPerDay * A.NoOfDays) DESC) AS rnk
    FROM Allocation A
    JOIN Patients P ON A.PatientID = P.PatientID
    GROUP BY P.PatientID, P.PatientName, P.DOB, P.PhoneNo, P.ArealD
) BillRanked
WHERE rnk = 1;
```

-- 6. Select the patient details who has stayed for maximum number of days in the hospital.

```
SELECT
    PatientID,
    PatientName,
    DOB,
    PhoneNo,
    ArealD,
    TOTAL_NO_DAYS
FROM (
    SELECT
        P.PatientID,
        P.PatientName,
        P.DOB,
```

```

        P.PhoneNo,
        P.ArealD,
        SUM(A.NoOfDays) AS TOTAL_NO_DAYS,
        RANK() OVER (ORDER BY SUM(A.NoOfDays) DESC) AS Max_days
FROM Patients P
JOIN Allocation A ON P.PatientID = A.PatientID
GROUP BY P.PatientID, P.PatientName, P.DOB, P.PhoneNo, P.ArealD
) StayRanked
WHERE Max_days = 1;

```

-- 7. Generate a report: PatientName, DiseaseID, DoctorID, No of times visited.

```

SELECT
    P.PatientName,
    A.DiseaseID,
    A.DoctorID,
    COUNT(*) AS No_Of_Times_Visited
FROM Allocation A
JOIN Patients P ON A.PatientID = P.PatientID
GROUP BY P.PatientName, A.DiseaseID, A.DoctorID
ORDER BY No_Of_Times_Visited DESC;

```

-- 9. Display the maximum salary earner for each hospital using RANK()

```

SELECT HospitalName, DoctorName, Salary AS Maximum_Salary
FROM (
    SELECT
        H.HospitalName,
        D.DoctorName,
        D.Salary,
        RANK() OVER (PARTITION BY D.HospitalID ORDER BY D.Salary DESC) AS rnk
    FROM Doctors D
    JOIN Hospitals H ON D.HospitalID = H.HospitalID
) AS RankedDoctors
WHERE rnk = 1;

```

-- 10. For each disease, display the patients who have visited the maximum number of times

```

SELECT DiseaseID, PatientID, Visit_Count
FROM (
    SELECT

```

```

        A.DiseaseID,
        A.PatientID,
        COUNT(*) AS Visit_Count,
        RANK() OVER (PARTITION BY A.DiseaseID ORDER BY COUNT(*) DESC) AS rnk
FROM Allocation A
GROUP BY A.DiseaseID, A.PatientID
) AS RankedVisits
WHERE rnk = 1;

```

-- 11. Generate a report: Doctor Name, Dean Name

```

SELECT
    D.DoctorName,
    Dean.DoctorName AS DeanName
FROM Doctors D
JOIN Doctors Dean ON D.Dean = Dean.DoctorID;

```

-- 12. Display the doctors who earns more than his dean

```

SELECT D.DOCTORNAME , DEAN.DOCTORNAME AS DEAN_NAME
FROM DOCTORS D
JOIN DOCTORS DEAN ON D.DEAN = DEAN.DOCTORID
WHERE D.SALARY > DEAN.SALARY ;

```

-- 13. Display the doctors whose salary is greater than the average salary of the hospital

```

SELECT
    D.DoctorID,
    D.DoctorName,
    D.Salary,
    H.HospitalName,
    AVG(D2.Salary) AS AvgHospitalSalary
FROM Doctors D
JOIN Hospitals H ON D.HospitalID = H.HospitalID
JOIN Doctors D2 ON D2.HospitalID = D.HospitalID
GROUP BY D.DoctorID, D.DoctorName, D.Salary, H.HospitalName
HAVING D.Salary > AVG(D2.Salary);

```

-- 14. Using CASE/DECODE

```

-- Patient ID
-- Number of days stayed
-- Description

```

-- If no of days stayed is between 1 and 10 : Description is "MILD"

-- If no of days stayed is between 11 and 20 : Description is "MINOR"

-- If no of days stayed is above 30 : Description is "SEVERE"

SELECT PATIENTID , NOOFDAYS ,

CASE

WHEN NOOFDAYS BETWEEN 1 AND 10 THEN "MILD"

WHEN NOOFDAYS BETWEEN 11 AND 20 THEN "MINOR"

WHEN NOOFDAYS > 30 THEN "SEVERE"

ELSE "MODERATE"

END AS DESCRIPTION

FROM ALLOCATION ;

-- 15. Generate a report:

-- Disease ID and severity description based on number of visits (attempts)

If no of attempts is lesser than 10, Description is "Least bothered"

If no of attempts is greater than 10 and lesser than 30 , Description is "To be taken special care"

If no of attempts is greater than 30, Description is "Spreading dangerous"

SELECT

A.DiseaseID,

COUNT(*) AS NoOfAttempts,

CASE

WHEN COUNT(*) < 10 THEN 'Least bothered'

WHEN COUNT(*) BETWEEN 11 AND 30 THEN 'To be taken special care'

WHEN COUNT(*) > 30 THEN 'Spreading dangerous'

ELSE 'Moderate'

END AS Description

FROM Allocation A

GROUP BY A.DiseaseID;

-- 16. In which month of 2012 no patient has visited the hospital "Sankara Nethralaya"

-- 16. Find the months in 2012 when no patient visited 'APOLLO'

SELECT

M.MONTH AS NoVisitMonth

FROM (

SELECT 1 AS MONTH UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION

SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION

SELECT 9 UNION SELECT 10 UNION SELECT 11 UNION SELECT 12

) M

WHERE M.MONTH NOT IN (

SELECT DISTINCT MONTH(A.AdmittedDate)

FROM Allocation A

JOIN Doctors D ON A.DoctorID = D.DoctorID

JOIN Hospitals H ON D.HospitalID = H.HospitalID

WHERE H.HospitalName = 'APOLLO'

AND YEAR(A.AdmittedDate) = 2012

);

-- 17. Prepare a report for the hospital "Madras Medical Mission":

<i>Year</i>	<i>Month</i>	<i>Number of patients visited</i>
-------------	--------------	-----------------------------------

SELECT YEAR(A.ADMITTEDDATE) AS YEAR , MONTH(A.ADMITTEDDATE) AS MONTH ,

COUNT(DISTINCT A.PATIENTID) AS NO_OF_PATIENTS_VISITED

FROM ALLOCATION A

JOIN DOCTORS D ON A.DOCTORID = D.DOCTORID

JOIN HOSPITALS H ON D.HOSPITALID = H.HOSPITALID

WHERE H.HOSPITALNAME = 'Madras Medical Mission'

GROUP BY YEAR(A.ADMITTEDDATE) , MONTH(A.ADMITTEDDATE) ;

-- 18. Generate a report: Doctor Name Age

SELECT DOCTORNAME , TIMESTAMPDIFF(YEAR , DOB , CURDATE()) AS AGE

FROM DOCTORS D ;

-- 19 .Generate a report:

<i>Disease ID</i>	<i>Patient ID</i>	<i>From Date</i>	<i>Discharged date</i>
-------------------	-------------------	------------------	------------------------

SELECT DiseaseID ,PatientID , AdmittedDate ,

(Admitteddate + INTERVAL NoOfDays DAY) as Discharge_date

FROM Allocation ;

-- 20. Generate a report: Disease ID, Patient ID, Doctor ID, No of weeks stayed

SELECT

DiseaseID,

PatientID,

DoctorID,

ROUND(NoOfDays / 7.0, 2) AS NoOfWeeksStayed -- Convert days to weeks (2 decimal precision)

FROM

Allocation;

-- Q21. Find TOP 5 salaried doctors

```
SELECT DoctorID, DoctorName, Salary
FROM (
    SELECT DoctorID, DoctorName, Salary,
           RANK() OVER (ORDER BY Salary DESC) AS SalaryRank
    FROM Doctors
) RankedDoctors
WHERE SalaryRank <= 5;
```

-- Q22. List TOP 10 hospitals (with respect to maximum no of patients visiting the hospital)

```
SELECT hospital_id, hospital_name, total_patients
FROM (
    SELECT h.hospital_id, h.hospital_name, COUNT(a.patient_id) AS total_patients,
           RANK() OVER (ORDER BY COUNT(a.patient_id) DESC) AS rnk
    FROM Hospitals h
    LEFT JOIN Allocation a ON h.hospital_id = a.hospital_id
    GROUP BY h.hospital_id, h.hospital_name
) sub
WHERE rnk <= 10
ORDER BY total_patients DESC;
```

-- Q23. For each hospital display the 3rd maximum salary earner

```
SELECT HospitalID, DoctorID, DoctorName, Salary
FROM (
    SELECT HospitalID, DoctorID, DoctorName, Salary,
           RANK() OVER (PARTITION BY HospitalID ORDER BY Salary DESC) AS SalaryRank
    FROM Doctors
) RankedDoctors
WHERE SalaryRank = 3;
```

/* Q24. How many diseases are cured by senior most doctor in Fortis hospital */

```
SELECT COUNT(DISTINCT DiseaseID) AS DiseasesCured
FROM (
    SELECT A.DoctorID, A.DiseaseID,
           RANK() OVER (ORDER BY D.DOB ASC) AS SeniorMost
    FROM Doctors D
    JOIN Allocation A ON D.DoctorID = A.DoctorID
    JOIN Hospitals H ON D.HospitalID = H.HospitalID
)
```

```

WHERE H.HospitalName = 'FORTIS'

) RankedDoctors

WHERE SeniorMost = 1;

/* Q25. List out the frequently cured diseases by minimum salary earner */

SELECT DiseaseID, DiseaseName, COUNT(*) AS TotalCures

FROM (

    SELECT A.DiseaseID, D2.DiseaseName

    FROM Doctors D

    JOIN Allocation A ON D.DoctorID = A.DoctorID

    JOIN Diseases D2 ON A.DiseaseID = D2.DiseaseID

    WHERE D.Salary = (SELECT MIN(Salary) FROM Doctors)

) DiseaseData

GROUP BY DiseaseID, DiseaseName

ORDER BY TotalCures DESC;

/* Q26. Generate a report: DoctorID, DoctorName, No of Patients cured */

SELECT DoctorID, DoctorName, COUNT(DISTINCT PatientID) AS NoOfPatientsCured

FROM (

    SELECT D.DoctorID, D.DoctorName, A.PatientID

    FROM Doctors D

    JOIN Allocation A ON D.DoctorID = A.DoctorID

) DoctorPatients

GROUP BY DoctorID, DoctorName

ORDER BY NoOfPatientsCured DESC;

/* Q27. Which doctor has cured all the diseases */

SELECT DoctorID, DoctorName

FROM (

    SELECT D.DoctorID, D.DoctorName,

           COUNT(DISTINCT A.DiseaseID) AS DiseasesCured

    FROM Doctors D

    JOIN Allocation A ON D.DoctorID = A.DoctorID

    GROUP BY D.DoctorID, D.DoctorName

) DoctorDiseaseCount

WHERE DiseasesCured = (SELECT COUNT(DISTINCT DiseaseID) FROM Diseases);

/* Q28. Which patient has not visited hospital so far (ALL POSSIBILITIES) */

SELECT PatientID, PatientName

```



```
FROM Patients
```

```
WHERE PatientID NOT IN (
```

```
    SELECT DISTINCT PatientID FROM Allocation
```

```
)
```

```
ORDER BY PatientID;
```

```
/* Q29. (Using ROLLUP) Generate a report: DoctorID, PatientID, No of visits */
```

```
SELECT
```

```
    DoctorID,
```

```
    PatientID,
```

```
    COUNT(*) AS NoOfVisits
```

```
FROM Allocation
```

```
GROUP BY ROLLUP(DoctorID, PatientID)
```

```
ORDER BY DoctorID, PatientID;
```

```
-- Q30. Generate a report showing Patient ID and Disease ID with subtotals and grand totals.
```

```
SELECT PatientID, DiseaseID, COUNT(*) AS NoOfAllocations
```

```
FROM Allocation
```

```
GROUP BY ROLLUP(PatientID, DiseaseID);
```

```
-- Q31. Display Doctor ID, Patient ID, Disease ID, and Comments where the comment is "Treatment going on" if NoOfDays is NULL.
```

```
-- If number of days stayed is NULL, then the comment must be "Treatment going on"
```

```
SELECT
```

```
    DoctorID,
```

```
    PatientID,
```

```
    DiseaseID,
```

```
    NVL(NoOfDays, 'Treatment going on') AS Comments
```

```
FROM Allocation;
```

```
-- Q31. Using NVL2 to handle NULL values in NoOfDays
```

```
SELECT
```

```
    DoctorID,
```

```
    PatientID,
```

```
    DiseaseID,
```

```
    NVL2(NoOfDays, NoOfDays, 'Treatment going on') AS Comments
```

```
FROM Allocation;
```

```
-- Q32. Count the number of patients undergoing treatment in Apollo hospital for coronary-related diseases.
```

```
SELECT COUNT(DISTINCT A.PatientID) AS TotalPatients
```

```
FROM Allocation A
JOIN Doctors D ON A.DoctorID = D.DoctorID
JOIN Diseases DS ON A.DiseaseID = DS.DiseaseID
JOIN Hospitals H ON D.HospitalID = H.HospitalID
WHERE H.HospitalName = 'APOLLO'
AND DS.CategoryID = 'C06'; -- Assuming 'C06' is for Coronary diseases

-- Q33. Find the category with the maximum number of diseases using subquery and RANK()
```

```
SELECT CategoryID, DiseaseCount
FROM (
    SELECT
        CategoryID,
        COUNT(DiseaseID) AS DiseaseCount,
        RANK() OVER (ORDER BY COUNT(DiseaseID) DESC) AS rnk
    FROM Diseases
    GROUP BY CategoryID
) RankedCategories
WHERE rnk = 1;
```

-- Q34. Find the hospital with the minimum number of doctors using subquery and RANK()

```
SELECT HospitalID, HospitalName, DoctorCount
FROM (
    SELECT
        HospitalID,
        HospitalName,
        COUNT(DoctorID) AS DoctorCount,
        RANK() OVER (ORDER BY COUNT(DoctorID) ASC) AS rnk
    FROM Hospitals H
    JOIN Doctors D ON H.HospitalID = D.HospitalID
    GROUP BY HospitalID, HospitalName
) RankedHospitals
WHERE rnk = 1;
```

-- Q35. Prepare a hierarchical report: Dean Name, Doctor Name

```
SELECT
    D1.DoctorName AS DeanName,
    D2.DoctorName AS DoctorName
FROM
```

Doctors D1

JOIN

Doctors D2 ON D1.DoctorID = D2.DoctorID

ORDER BY

D1.DoctorName, D2.DoctorName;

-- Q36. How many patients are allocated to the senior most doctors in June 2013?

SELECT D.DoctorID,D.DoctorName,COUNT(A.PatientID) AS TotalPatients

FROM Doctors D

JOIN Allocation A ON D.DoctorID = A.DoctorID

WHERE A.AdmittedDate BETWEEN '2013-06-01' AND '2013-06-30'

AND D.DOB = (SELECT MIN(DOB) FROM Doctors)

GROUP BY D.DoctorID,D.DoctorName;

-- Q37. List out the senior most patient details who had stayed for more than the expected days for the disease "Jaundice".

SELECT P.PatientID, P.PatientName, P.DOB, P.PhoneNo, A.NoOfDays, D.DiseaseName

FROM Patients P

JOIN Allocation A ON P.PatientID = A.PatientID

JOIN Diseases D ON A.DiseaseID = D.DiseaseID

WHERE D.DiseaseName = 'JAUNDICE'

AND A.NoOfDays > 7 -- Adjust this based on the expected number of days for "Jaundice"

AND P.DOB = (SELECT MIN(DOB) FROM Patients);

-- Q38. Prepare a report: Patient Name, Doctor Name, Disease Name, Total Amount Paid

-- Sort the report with respect to the Total Amount paid in Descending Order.

-- Total amount paid should be 0 if patient has not paid any amount.

SELECT

P.PatientName,

D.DoctorName,

DI.DiseaseName,

NVL(SUM(A.FeesPerDay * A.NoOfDays), 0) AS TotalAmountPaid

FROM Patients P

JOIN Allocation A ON P.PatientID = A.PatientID

JOIN Doctors D ON A.DoctorID = D.DoctorID

JOIN Diseases DI ON A.DiseaseID = DI.DiseaseID

GROUP BY P.PatientName, D.DoctorName, DI.DiseaseName

ORDER BY TotalAmountPaid DESC;

-- Q39. Display TOP 10 Patient Details with respect to the Bill Amount paid so far

```
SELECT PatientID, PatientName, TotalAmountPaid
FROM (
    SELECT
        P.PatientID,
        P.PatientName,
        NVL(SUM(A.FeesPerDay * A.NoOfDays), 0) AS TotalAmountPaid,
        RANK() OVER (ORDER BY NVL(SUM(A.FeesPerDay * A.NoOfDays), 0) DESC) AS rnk
    FROM Patients P
    JOIN Allocation A ON P.PatientID = A.PatientID
    GROUP BY P.PatientID, P.PatientName
) RankedPatients
WHERE rnk <= 10;
```

-- Q40. Display doctors who had joined before their dean.

```
SELECT D.DoctorID, D.DoctorName
FROM Doctors D
JOIN Doctors Dean ON D.Dean = Dean.DoctorID
WHERE D.DOJ < Dean.DOJ;
```

-- Q41. Prepare a report showing Doctor ID and the number of patients attended.

```
SELECT
    D.DoctorID,
    COUNT(DISTINCT A.PatientID) AS NoOfPatientsAttended
FROM Doctors D
JOIN Allocation A ON D.DoctorID = A.DoctorID
GROUP BY D.DoctorID;
```

-- Q42. Count the number of patients allocated to doctor Ashok during the first quarter of 2012.

```
SELECT COUNT(DISTINCT A.PatientID) AS TotalPatients
FROM Allocation A
JOIN Doctors D ON A.DoctorID = D.DoctorID
WHERE D.DoctorName = 'Ashok'
AND A.AdmittedDate BETWEEN '2012-01-01' AND '2012-03-31';
```

-- Q43. Count the number of patients allocated to doctor Ashok during the third quarter of 2011.

```
SELECT COUNT(DISTINCT A.PatientID) AS TotalPatients
FROM Allocation A
JOIN Doctors D ON A.DoctorID = D.DoctorID
```

WHERE D.DoctorName = 'Ashok'

AND A.AdmittedDate BETWEEN '2011-07-01' AND '2011-09-30';