

```

1  # All the codes and homeworks for Madison hospital database
2
3  /*Sagar Kalauni*/
4  /* Lab2-*/
5  /*1.Create a table named prod_table. This table should have two columns named prod_id
and prod_description. These columns should be defined to store the following type of
data, respectively: prod_id stores numeric data that is a maximum of 3 characters in
size; prod_description stores variable character data that is a maximum of 25 characters
in size.*/
6  CREATE TABLE prod_table (
7      Prod_id          NUMBER (3),
8      prod_description  VARCHAR2 (25)
9  );
10 /*Table has been created*/
11
12 /*2.Insert two rows into the test_table*/
13 INSERT INTO prod_table VALUES (1, 'Wheel');
14 INSERT INTO prod_table VALUES (2, 'Nuts and Bolts');
15 /*Two records have been inserted into the 'prod_table'.*/
16
17
18 /*3.Use the DESCRIBE command to describe the prod_table. */
19 DESC prod_table;
20 /*The resulting output is displaying the 'prod_table' along with its columns and their
respective data types.*/
21
22 /*4.Use the following SELECT command to display the rows in the prod_table. */
23 SELECT * FROM prod_table;
24 /*The output encompass all columns of the 'prod_table'.*/
25
26 /*5.Use the DROP command to drop the prod_table. */
27 DROP TABLE prod_table;
28 /*The table has been successfully dropped from our database.*/
29
30 /*6.Create the deptBusiness table described below: */
31 CREATE TABLE DeptBusiness (
32     DepartmentNumber          NUMBER (4)
33     CONSTRAINT PK_DeptBusiness PRIMARY KEY,
34     DepartmentName            VARCHAR2 (25)
35     CONSTRAINT NN_DepartmentName NOT NULL,          /* Creating unique constraint
name*/
36     ManagerID                 CHAR (5)
37 );
38 /*A new table named 'DeptBusiness' has been successfully created in our database.*/
39
40 /*7.Add the data shown below to the deptBusiness table. Do not define any foreign
keys. Leave the ManagerID column values as NULL. */
41 INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
42 VALUES (1106, 'CMIS');
43
44 INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
45 VALUES (1105, 'Accounting');
46
47 INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
48 VALUES (1100, 'Production');
49
50 INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
51 VALUES (1102, 'Economic Finance');
52 /*All the provided values have been successfully added to the 'DeptBusiness' table.*/
53
54 /*8.a. COMMIT your row insertions in the deptBusiness table. b. Try to INSERT the data
for department number 1106 again in the deptBusiness table. Did Oracle accept it? */
55 COMMIT;
56 /*Upon the completion of the commit operation, all changes have been successfully stored
and applied.*/
57
58 INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
59 VALUES (1106, 'CMIS');
60 /*value is not accepted saying error: Unique constraint violated*/

```

```

61
62  /*9.Use the following SELECT command to display the rows in the deptBusiness table...*/
63  SELECT *
64  FROM DeptBusiness;
65  /*The output displays the 'DeptBusiness' table with all the provided data entries
effectively populated within it.*/
66
67  /*10.Delete the row for department number 1 from the deptBusiness table.      */
68  DELETE FROM DeptBusiness
69         WHERE DepartmentNumber= 1106;
70  /*A single record has been removed from our 'DeptBusiness' table.*/
71
72
73  /*11.Repeat the SELECT statement in question #9 above to verify your record has been
deleted*/
74  SELECT *
75  FROM DeptBusiness;
76  /*Output clearly shows that in the DeptBusiness table now we have no longer the
observation having department number 1106, which have been deleted in the last SQL
command*/
77
78
79  /*12.Assume that the deletion of your row was an error.  Execute the ROLLBACK command
(SQL> ROLLBACK;) to undelete your row.  Use the SELECT * statement again to verify that
your row has been restored to the table.  */
80  ROLLBACK;
81  /*This SQL command effectively reverses the impact of our previous SQL command, which
involved deleting one row from the 'DeptBusiness' table.*/
82
83  SELECT *
84  FROM DeptBusiness;
85  /*Output is clearly showing that the deleted row has been sucessfully retrived back*/
86
87  /*13.The name for 'Production' department got changed to 'Operations'. Update the
DepartmentName column of this change accordingly. Also Repeat the SELECT statement in
question #9 above to verify your output.  */
88  UPDATE DeptBusiness
89         SET DepartmentName = 'Operations'
90         WHERE DepartmentName= 'Production';
91  /*The 'DepartmentName' has been updated from 'Production' to 'Operations'*/
92
93  SELECT *
94  FROM DeptBusiness;
95  /*The modified department name can be observed in the output.*/
96
97  /*14.Alter the deptBusiness table to add a column that will be used to store the
department phone. Name this column DepartmentPhone and use an appropriate NUMBER
datatype specification. You do not need to store any data to this column. Also Repeat
the SELECT statement in question #9 above to verify your output. */
98  ALTER TABLE DeptBusiness ADD (DepartmentPhone NUMBER (10));
99  /*A new column named 'DepartmentPhone' has been successfully added to the 'DeptBusiness'
table.*/
100
101  SELECT *
102  FROM DeptBusiness;
103  /*Output clearly shows newly added column 'DepartmentPhone' in the table DeptBusiness*/
104
105
106  /*15.Use the DROP command to drop the DeptBusiness table. Use the SELECT statement given
in question #9 above to display the deptBusiness table.  */
107  DROP TABLE DeptBusiness;
108  /*Table has been sucessfully dropped out from out database*/
109
110  SELECT *
111  FROM DeptBusiness;
112  /*yes,error message appears, Since Table was already dropped out form the database so the
output is: table does not exist*/
113  /*THE END*/
114

```

```

115 #####
116 #####
117
118
119 /*Sagar Kalauni*/
120 /*Lab4-Kalauni*/
121
122 /*Q NO. 1) Write a query that will select all columns from the Specialty table without
using the (*) in your query. You may wish to use the DESCRIBE command to examine the
structure of the Specialty table*/
123 /*First let's look at all the columns in the Specialty Table using DESC command */
124 DESC Specialty;
125
126 /*Manually selecting all columns of the Specialty table*/
127 SELECT SpecialtyID, Title, AwardedBy
128 FROM Specialty;
129 /*OBSERVATION;- We can see Specialty table with all its columns*/
130
131 /*Q NO.2)Your manager wonders what types of employee titles are tracked in the Employee
table. Produce a sample listing of the titles of employees at the hospital that does
not include any duplicate rows. */
132 /*Using distinct command to select the all non duplicated titles in the employee table*/
133 SELECT distinct title
134 FROM Employee;
135 /*OBSERVATION:- So There are 10 different type of employee titles in the employee table*/
136
137 /*Q NO. 3)Execute a query that will display all treatment dates for patient 100302.
Include the patientID, employeeID, and date treated.*/
138 /*To fully show column name and make our output infomative, formating done initially*/
139 COLUMN PatientID FORMAT A10;
140 COLUMN EmployeeID FORMAT A12;
141 COLUMN DateTreated FORMAT A12;
142 SELECT PatientID, EmployeeID, DateTreated
143 From Treatment
144 WHERE PatientID='100302';
145 /*OBSERVATION:- All required data for patient with patientid=100302*/
146
147 /*Q NO.4)Execute a query that lists all the male nurses - i.e., employees with a job
title that includes the degree 'R.N..' List each employee's last name, first name,
title, and gender. Format your columns so that lastName is 12 characters, firstName is
12 characters, title is 5 characters, and gender is 6 characters.*/
148 /*M- Male as a gender*/
149 COLUMN LastName FORMAT A12;
150 COLUMN FirstName FORMAT A12;
151 COLUMN Title FORMAT A5;
152 COLUMN Gender FORMAT A6;
153 SELECT LastName, FirstName, Title, Gender
154 FROM Employee
155 WHERE Title LIKE '%R.N..%' AND Gender='M';
156
157 /*Q No.5)Produce a listing that will only display patients from Alton or Collinsville.
The result table should display the first name and last name of these patients as well
as the city they are from. Sort the report by city. Format first name and last name so
they are 12 characters wide each.*/
158 /*While looking for city in where condition, we should give the city name as they are in
the table, because it is case sensetive*/
159 COLUMN LastName FORMAT A12;
160 COLUMN FirstName FORMAT A12;
161 SELECT FirstName, LastName, City
162 FROM Patient
163 WHERE City IN ('Alton', 'Collinsville')
164 ORDER BY City;
165 /*OBSERVATION:- There are 21 patients from city Alton and Collinsville combinedly */
166
167 /*Q NO.6)Execute a query that will display all equipment with an original cost below
$500, AND either have 10 or more items in stock (quantityAvailable) OR are used in
project 8. List the equipment description, original cost, quantity available, and

```

```

project number. Format your columns so that the output fits on one row. Format your
columns so the output fits on one row.*/
168 /*First looking at the table and resizing the format*/
169 COLUMN Description FORMAT A12;
170 COLUMN "Original Cost" FORMAT 9999.99;
171 COLUMN "Quantity Available" FORMAT 9999;
172 COLUMN "Project Number" FORMAT 99;
173 SELECT Description, OriginalCost "Original Cost", QuantityAvailable "Quantity Available",
ProjectNumber "Projec Number"
174 FROM Equipment
175 WHERE OriginalCost<500 AND (QuantityAvailable>=10 OR ProjectNumber=8);
176 /*OBSERVATION:- Output fits perfectly*/
177
178 /*Q NO.7)The CFO needs a report to justify standard charges to the State Health
Department. The report should contain the description, standard charge, and category ID.
Sort the report by categoryID and then by standard charge. Use the appropriate command
to limit the output column width for the description to 30 characters. The report should
be sorted first by categoryID and then by standard charge with the highest standard
charge for each category appearing first.*/
179 COLUMN Description A30;
180 SELECT Description, StandardCharge, CategoryID
181 FROM Service
182 ORDER BY CategoryID, StandardCharge DESC;
183 /*OBSERVATION:-
184 Nicely shorted table, with 105 rows*/
185
186 #####
187 #####
188 #####
189 #####
189 /*Sagar Kalauni*/
190 /*Lab5-Kalauni*/
191
192 /*Q No.1)The charge nurse wants to see the medications that have instructions for child
dosages and have a quantity on hand of less than 1,000. Instructions indicating if the
medicine is for children can be found in the dosage field of the Medicine table. The
term child may appear anywhere in this field. Create a list that includes the common
name and scientific name of the medication. Restrict common name to 15 characters and
scientific names to 20 characters.*/
193 COLUMN Commonname FORMAT A15;
194 COLUMN Commonname FORMAT A20;
195 SELECT Commonname, ScientificName
196 FROM Medicine
197 WHERE Dosage LIKE '%child%' AND QuantityOnhand < 1000;
198 /*OBSERVATION:- So the required medication name is Atarax(commonname)*/
199
200 /*Q.NO2)The HR director of the hospital has learned that the average monthly salary of
employees is about $15,000. She would like a report of all employee names who earn more
than $10,000 a month and less than $20,000 monthly. Use the BETWEEN operator. Sort by
salary. Include the first name and last name of the employee (restrict both names to 12
characters) as well as the salary.*/
201 COLUMN "First Name" FORMAT A12;
202 COLUMN "Last Name" FORMAT A12;
203 COLUMN "moSalary" FORMAT $99,999;
204 SELECT FirstName "First Name", LastName "Last Name", Salary "moSalary"
205 FROM Employee
206 WHERE Salary BETWEEN 10000 AND 20000
207 ORDER BY Salary;
208 /*OBSERVATION:- So there are 8 employees whose salary is in between $10,000 to $20,000*/
209
210 /*Q NO.3)There has been a surge of parents requesting private rooms for their children
who have been admitted for surgery. Provide a list of beds available in the following
pediatric rooms: PED101 - PED105 inclusive. Use the IN command. The room number,
bedtype, and availability should be shown in the report. Use meaningful column titles.*/
211 /*To make output result more informative and clear, formating done*/
212 COLUMN "Bed Number" FORMAT A12;
213 COLUMN "Bed Type" FORMAT A10;
214 COLUMN "Bed Availability" FORMAT A18;

```

```

215 SELECT RoomNumber "Bed Number", BedType "Bed Type", Availability "Bed Availability"
216 FROM Bed
217 WHERE RoomNumber IN ('PED101', 'PED102', 'PED103', 'PED104', 'PED105');
218 /*OBSERVATION:- All required information of Pediatric Rooms for 101 to 105*/
219
220 /*Q NO.4)The director of HR would like to implement a process of sending birthday cards
to the children of employees. She is requesting a report that contains the first names
and birthdates of all sons and daughters of employees. You must use the IN command.*/
221 SELECT Name "First Name", BirthDate
222 FROM Dependent
223 WHERE RelationshipToEmployee IN ('SON', 'DAUGHTER');
224 /*OBSERVATION:- Table showing firstname and birthday of son's and daughter's of
employees*/
225
226 /*Q NO.5)Execute a query that will display all employees whose last name contains the
lower case letter 'o' except for the second character (i.e., the second character can be
anything but 'o'. List each employee's first and last name. Use meaningful column
titles.*/
227 SELECT FirstName "First Name", LastName "Last Name"
228 FROM Employee
229 WHERE LastName LIKE '%o%' AND LastName NOT LIKE '_o%';
230 /*OBSERVATION:- So there are 7 employees who have o in their lastname but not in the
second position*/
231
232 /*Q NO.6)Execute a query that lists all employee table rows that contain a null value in
the salary column. List each employee's last name and supervisor identifying number. Use
meaningful column titles. Limit column width so a line fits on a single row.*/
233 COLUMN "Last Name" FORMAT A10;
234 COLUMN "Supervisor Identifying Number" FORMAT A30;
235 SELECT LastName "Last Name", SupervisorID "Supervisor Identifying Number"
236 FROM Employee
237 WHERE Salary IS NULL;
238 /*OBSERVATION:- So there are 2 employees whose salary is null*/
239
240 /*Q NO.7)Execute a query that will display all patients whose first name begins with the
same letter as your first name. List each patient's first name and last name. Format the
output so the full name appears on one line. Provide meaningful column titles. If your
last name begins with the X, list patients whose first name begins with the same letter
as your middle name. Use meaningful column titles.*/
241 /*As my first name is Sagar*/
242 COLUMN "First Name" FORMAT A12;
243 COLUMN "Last Name" FORMAT A12;
244 SELECT FirstName "First Name", LastName "Last Name"
245 FROM Patient
246 WHERE FirstName LIKE 'S%';
247
248 /*As my lastname is Kalauni, and I do not have middle name, below I am searching for
employee whose first name begin with first character of my last Name*/
249 SELECT FirstName "First Name", LastName "Last Name"
250 FROM Patient
251 WHERE FirstName LIKE 'K%';
252
253 /*Q NO.8)Execute a query that will display each employee's last name, annual salary,
monthly salary, and weekly salary. The list should only include employees with a weekly
salary that is less than $1,000.00. Label the column names for annual salary, monthly
salary, and weekly salary as Annual, Monthly, and Weekly, respectively. Sort the output
by employee last name. Format the columns named Annual, Monthly, and Weekly as
$999,999.99. Be careful in how you compute the weekly salary! You may assume 4 weeks to
a month. You can assume the field Salary represents the monthly salaries of employees.
Use meaningful column titles*/
254 COLUMN "Annual" FORMAT $999,999.99;
255 COLUMN "Monthly" FORMAT $999,999.99;
256 COLUMN "Weekly" FORMAT $999,999.99;
257 SELECT LastName "Last Name", Salary*12 "Annual", Salary "Monthly", Salary/4 "Weekly"
258 FROM Employee
259 WHERE Salary/4 < 1000
260 ORDER By LastName;
261
262 /*The End*/

```

```

263
264
265
266 #####
267 #####
268
269 /*Sagar Kalauni*/
270 /* Lab6-Kalauni*/
271
272 /*Q.NO. 1)A manager from the human resources department needs you to write a query to
count the number of employees of the company that are nurses (either Title = 'R.N.' OR
Title = 'L.P.N.'. Label the output column Number of Nurses. */
273 /*OBSERVATION:- There are 2 Nurses*/
274 SELECT count(*) "Number of Nurses"
275 FROM Employee
276 WHERE Title in ('R.N', 'L.P.N.');
```

277

```

278 /*Q.NO.2)Accountants working on the company's annual budgeting process need to know the
average cost of the equipment being used on projects (originalCost) and the sum of all
equipment costs. The information is stored in the Equipment table. The result table
should have two columns based on a single query. Label the columns Average Equipment
Cost and Total Equipment Cost. Format the output as $99,999.99. */
279 /*OBSERVATION:- Average Cost is $2,019.85 and Total cost is $18,178.67*/
280 COLUMN "Average Equipment Cost" FORMAT $99,999.99;
281 COLUMN "Total Equipment Cost" FORMAT $99,999.99;
282 SELECT AVG(OriginalCost) "Average Equipment Cost", SUM(OriginalCost) "Total Equipment
Cost"
283 FROM Equipment;
```

284

```

285 /*Q>NO.3)The BirthDate column in the dependent table stores date of birth information
for dependents of employees of the company. Write a query to display the date of birth
of the oldest dependent listed in the table. No special output column label is
required.*/
286 /*Used Minimum to find the oldest dependent*/
287 SELECT MIN(BirthDate)
288 FROM Dependent;
```

289

```

290 /*Q.NO.4)Write a query to provide the Executive Director with the total hours worked per
project. Use the ProjectNumber and HoursWorked columns from the ProjectAssignment table
to obtain the project numbers and hours worked, respectively. Label the two columns
Project Number and Total Hours respectively. Sort by project number. Format the output
for the Total Hours column as 999.99.*/
291 /*OBSERVATION:- Nice table showing Project Number and Total hours as output*/
292 COLUMN "Total Hours" FORMAT 999.99;
293 SELECT ProjectNumber "Project Number" ,SUM(HoursWorked) "Total Hours"
294 FROM ProjectAssignment
295 GROUP BY ProjectNumber
296 ORDER BY ProjectNumber;
```

297

```

298 /*Q.NO.5)The government reporting regulation also requires a report of the count of all
employees who are NOT M.D.s. M.D. could appear anywhere in the title. Write a query that
will produce a result table with two columns labeled Title and Non M.D. Employees.
Format the Title column so it is 20 characters.*/
299 /*Where clause has eliminated any who has something title like M.D*/
300 /*OBSERVATION:- So there are total of 11 non M.D Employees*/
301 COLUMN "Title" FORMAT NAMEA20;
302 SELECT Title "Title", COUNT(Title) "NON M.D Employees"
303 FROM Employee
304 WHERE Title NOT LIKE '%M.D%'
305 GROUP BY Title;
```

306

```

307 /*Q.NO.6)The CEO would like a report with PatientIDs and the total amount of treatment
charges they have had. The resulting report should have two columns: Patient and Total
Charges. The report should be listed in Total Charges order with the patients with the
lowest Total Charges at the top. The CEO wants to see only those patients whose total
charges are less than $350. */
308 /*OBSERVATION:- Output is table with patient and their total charge*/
```



```

309 SELECT PatientID "Patient", SUM(CHARGEAmount) "Total Charges"
310 FROM Treatment
311 GROUP BY PatientID
312 HAVING SUM(CHARGEAmount)< 350
313 ORDER BY SUM(CHARGEAmount);
314
315 /*Q.NO.7)Modify the query written for question 6. Now the CEO would like to see average
charges for treatments each patient is responsible for. But, he does not want treatment
charges of under $500 considered in the average. After averaging, the report should
show only those patients whose average treatment charges are greater than $500. */
316 /*Where condition eliminates rows having chargeamount less then or equal 500*/
317 /*OBSERVATION:- Output is table with patient and their average charge*/
318 SELECT PatientID "Patient", AVG(CHARGEAmount) "average charges"
319 FROM Treatment
320 WHERE CHARGEAmount>=500
321 GROUP BY PatientID
322 HAVING AVG(CHARGEAmount)>500
323 ORDER BY SUM(CHARGEAmount);
324
325 /* THE END */
326
327
328
329 #####
330 #####
331 #####
332 #####
333
334 /*Sagar Kalauni*/
335 /* Lab7-Kalauni*/
336
337 /*Q.NO.1)The company's vice president for project management needs a listing of
employees who have received specialties. The result table should list the employee name
(last name first, then first name) and their associated specialty title and the date
they received the specialty. Format the columns so output lines are on a single row.
The column header for the Specialty title should be 'Specialty Name'. */
338 /*OBSERVATION:- So 21 employees has recived Specialities */
339 COLUMN "Employee Name" FORMAT A24;
340 COLUMN "Specialty Name" FORMAT A27;
341 COLUMN "Date Recived" FORMAT A12;
342 SELECT LastName||', '||FirstName "Employee Name" , s.Title "Specialty Name", DateReceived
343 "Date Recived"
344 FROM Employee e JOIN EmployeeSpecialty es ON (e.EmployeeID=es.EmployeeID) JOIN Specialty
345 s ON (es.SpecialtyID=s.SpecialtyID);
346
347 /*Q.NO.2)The hospital pharmacist would like a report listing patient first and last
names (concatenated) who have been prescribed Valium. The result table should have just
two columns, Patient Name and CommonName. Use relevant column headers and format
commonName so it is 10 or fewer characters.*/
348 /*OBSERVATION:- Patient Name who are Prescribed Valium*/
349 COLUMN "CommonName" FORMAT A10;
350 SELECT LastName||', '||FirstName "Patient Name", CommonName "CommonName"
351 FROM patient p JOIN Prescription pr ON (p.PatientID=pr.PatientID) JOIN Medicine m ON (pr.
352 MedicineCode=m.MedicineCode)
353 WHERE Commonname='Valium';
354
355 /*Q.NO.3)The company's vice president for project management needs a listing of
employees assigned to projects. The result table should list the LastName and FirstName
column values (concatenated into one column) and their ProjectTitle and HoursWorked
(from the projectAssignment table). The result table should only list employees
assigned to a project that have worked on the project more than 10 hours. Sort results
by employee last name and then by project number. Use alias names for the table names.
Give each column an appropriate column name. HoursWorked has a decimal.*/
356 /*OBSERVATION:- Nicely formatted table with employee name, their project title and hours
worked*/
357 COLUMN "Hours Worked" FORMAT 99.9;
358 SELECT LastName||', '||FirstName "Employee Name", ProjectTitle "Project Title",
359 HoursWorked "Hours Worked"
360 FROM Employee e JOIN ProjectAssignment pa ON (e.EmployeeID=pa.EmployeeID) JOIN Project p

```

```

355 ON (pa.ProjectNumber=p.ProjectNumber)
356 WHERE HoursWorked >10
357 ORDER BY LastName, p.ProjectNumber;
358
/*Q.NO4)Produce a query that will list all employee last names, employee gender,
dependent names and dependent gender where the employee's have dependents of the same
gender. Also list the dependent relationship. The columns needed in the result table
are LastName, employee.Gender, dependent.Name, dependent.Gender, and
RelationshipToEmployee. Use the employee and dependent tables. Use the FROM clause to
join the tables. Use the column names and formats shown below. Sort the result table
by LastName.
359 COLUMN "Employee" FORMAT A10;
360 COLUMN "Emp Gender" FORMAT A10;
361 COLUMN "Dependent" FORMAT A10;
362 COLUMN "Dep Gender" FORMAT A10;
363 COLUMN "Relationship" FORMAT A12; */
364 /*OBSERVATION:- Nicely formatted table with information of employee, dependent their
relation and gender */
365 COLUMN "Employee" FORMAT A10;
366 COLUMN "Emp Gender" FORMAT A10;
367 COLUMN "Dependent" FORMAT A10;
368 COLUMN "Dep Gender" FORMAT A10;
369 COLUMN "Relationship" FORMAT A12;
370 SELECT LastName "Employee", employee.Gender "Emp Gender", dependent.Name "Dependent",
dependent.Gender "Dep Gender", RelationshipToEmployee "Relationship"
371 FROM employee JOIN dependent ON (employee.EmployeeID=dependent.EmployeeID)
372 WHERE employee.Gender=dependent.Gender
373 ORDER BY LastName;
374
375 /*Q.NO.5)The director of Pediatric Nursing requires a report listing each patient name,
their bedNumber and their roomNumber for pediatric patients only. You will need to
figure out how to determine which rooms are for pediatric patients and there is more
than one way to do this. The result table should display the patient's first and last
names concatenated, their bednumber, and roomnumber. Assign appropriate column headers
and format so the columns fit on a single line.*/
376 /*OBSERVATION:- Name and bed number and room no. of the patient in pediatric*/
377 COLUMN "Patient Name" FORMAT A20;
378 SELECT p.FirstName||' '||p.LastName "Patient Name", bedNumber "Bed Number", roomNumber
"Room Number"
379 FROM Patient p JOIN Bed b
380 ON (p.bedNo = b.bedNumber)JOIN Prescription pr
381 ON (pr.patientID = p.patientID) JOIN Employee e
382 ON (pr.employeeID = e.employeeID) JOIN Department d
383 ON (e.departmentNumber = d.departmentNumber)
384 WHERE departmentName = 'Pediatrics-Gynecology';
385
386 /*Q.NO.6)Management is expecting to start several new projects in the near future. A
list of employees who are not currently assigned to a project is needed. This will
provide management with a list of employees who are potentially available to be assigned
to projects. The result table should list the LastName and FirstName column values
(concatenated into one column) from the employee table. Sort the output by LastName and
FirstName. Hint: You will need an OUTER join.*/
387 /*OBSERVATION:- So there are 13 employee who are not assigned to any project currently*/
388 SELECT LastName||', '||FirstName "Employee Name"
389 FROM Employee e LEFT OUTER JOIN ProjectAssignment pa ON (e.EmployeeID=pa.EmployeeID) LEFT
OUTER JOIN Project p ON (pa.ProjectNumber=p.ProjectNumber)
390 WHERE pa.Projectnumber IS NULL
391 ORDER BY e.LastName, e.FirstName;
392
393 /*Q.NO.7)The head of Facilities Management needs a list of all roomnumbers and the
patients currently in each room. The result table should include ALL room numbers in
the hospital. If a patient is currently in the room, their name should be shown in
firstName lastName order and should be concatenated in a single column. This will allow
the Facilities Management team to clean rooms that are currently unoccupied.*/
394 /*OBSERVATION:- Nicely formatted table showing bed number and the patient name if
occupied*/
395 SELECT RoomNumber "Room Number", FirstName||' '||LastName "Patient Name"
396 FROM Bed b LEFT OUTER JOIN Patient p ON (p.Bedno=b.BedNumber);
397

```



```

398  /*Q.NO.8)Produce a listing of number of employees who have earned each specialty. The
result table should list the Specialty Name and number of employees who have earned the
specialty. Give each column an appropriate column name and format Specialty Title to 30
characters.*/
399  /*OBSERVATION:- Nicely formatted table showing count of employee have particular type of
specialty */
400  COLUMN "Specialty Name" FORMAT A30;
401  SELECT s.Title "Specialty Name", Count(e.employeeID) "No. of Employees"
402  FROM Employee e JOIN EmployeeSpecialty es ON (e.EmployeeID=es.EmployeeID) JOIN Specialty
s ON (es.SpecialtyID=s.SpecialtyID)
403  GROUP BY s.Title;
404
405  /*Q.NO.9)Produce a listing with the number (count) of employees assigned to each project
within each department. The result table should list the DepartmentName, ProjectTitle
and number of employees. Give each column an appropriate column name. Use either the
WHERE or the FROM clause to join the tables. This query is a little tricky - because you
have two attributes on the SELECT line along with the aggregate, your GROUP BY statement
will also need attributes.*/
406  /*OBSERVATION:- Nicely formatted table showing no. of employee assigned to each project
within each department*/
407  SELECT DepartmentName "Department Name", ProjectTitle "Project Title", count(e.employeeID
) "No. of Employees"
408  FROM Employee e JOIN Department d ON (e.DepartmentNumber=d.DepartmentNumber) JOIN
Project p ON (d.DepartmentNumber=p.DepartmentNumber)
409  GROUP BY p.ProjectTitle, d.DepartmentName;
410
411  /*Q.NO.10)Produce a listing of the number of patients treated and the total service
charges ChargeAmount) received by each doctor (Title includes 'M.D.'). The result table
should have three columns: the LastName and FirstName concatenated into one column,
number of patients, and the total service charges. Give each column an appropriate
column name. Use either the WHERE or the FROM clause to join the tables. You must
determine on your own which tables and columns are required to produce the result
table.*/
412  /*OBSERVATION:- Nicely formatted table showing doctor's name, number of patient he
treated and service charge*/
413  COLUMN "Doctor Name" FORMAT A20;
414  COLUMN "Total Service Charge" FORMAT $99,999.99;
415  SELECT LastName||', '||FirstName "Doctor Name", Count(t.PatientID) "Number of Patient",
SUM(t.ChargeAmount) "Total Service Charge"
416  FROM Employee e JOIN Treatment t on (e.EmployeeID=t.EmployeeID)
417  WHERE Title LIKE '%M.D.%'
418  GROUP BY e.LastName, e.FirstName;
419
420  /*Q.NO.11)The payroll department needs to regularly access information about employee
salary information. The DBA of the company has directed you to create a view based on
the employee table named vwSalary. This view should include the employee identifying
number, employee last and first names (LastName and FirstName), and the salary for each
employee. Name the columns of the view as follows: EmpID, EmpLastName, EmpFirstName,
and EmpSalary. Write the SQL code needed to create this view. Write a SELECT statement
to display rows from the view for employees with salaries at or above $20,000. Format
all output appropriately.*/
421  /*Creating a view named vwSalary*/
422  COLUMN "EmpID" FORMAT A8;
423  COLUMN "EmpLastName" FORMAT A15;
424  COLUMN "EmpFirstName" FORMAT A15;
425  COLUMN "EmpSalary" FORMAT $999,999,999.99;
426  CREATE VIEW vwSalary
427  (EmpID, EmpLastName, EmpFirstName, EmpSalary) AS
428  SELECT EmployeeID, LastName, FirstName, Salary
429  FROM Employee;
430
431  /*accessing vwSalary*/
432  SELECT *
433  FROM vwSalary
434  WHERE EmpSalary>=20000;
435
436  /*Q.No.12)The Company's senior project manager needs to access information about
departments that manage projects for a specific set of projects, namely those located in
either Maryville or Edwardsville. Create a view named vwDepartmentProjects that includes

```

```

the DepartmentNumber and DepartmentName columns from the department table and the
ProjectTitle and Location columns from the project table. The view should only
reference rows for projects that are located in either Maryville or Edwardsville. The
columns in the view should be named DeptNo, Department, Project, and Location,
respectively. Write a SELECT statement to display all of the rows that are accessible
through the view. Format the output columns of the SELECT statement as A25 for Project
and Department, and A15 for Location.*/
437 /*Creating a view named vwDepartmentProjects*/
438 COLUMN "Project" FORMAT A25;
439 COLUMN "Department" FORMAT A25;
440 COLUMN "Location" FORMAT A15;
441 CREATE VIEW vwDepartmentProjects
442 (DeptNo, Department, Project, Location) AS
443 SELECT d.DepartmentNumber, d.DepartmentName, p.ProjectTitle, p.Location
444 FROM Department d JOIN Project p ON (d.DepartmentNumber=p.DepartmentNumber)
445 WHERE Location IN ('Maryville', 'Edwardsville');
446
447 /*accessing vwDepartmentProjects*/
448 SELECT *
449 FROM vwDepartmentProjects;
450
451 /*Q.No.13)Create a view named vwProjectHours that will be used by the senior project
manager to access information about work hours that have been reported for different
projects. The view should join the project and projectAssignment tables. The view
should have two columns; project title (not project number) and the average hours worked
on each project. Name the columns Project and AverageHours in the view. (Hint: The
rows in the view should be grouped by the project name). Write a SELECT statement
against this view to display projects where the average hours is equal to or greater
than 15. HINT: When creating the View you have renamed the average hours to
AverageHours. This is the attribute you will reference in the query against the view.*/
452 /*Creating a view named vwProjectHours*/
453 CREATE VIEW vwProjectHours
454 ("Project", "AverageHours")AS
455 SELECT ProjectTitle, Avg(Hoursworked)
456 FROM ProjectAssignment pa JOIN Project p ON (pa.ProjectNumber=p.ProjectNumber)
457 GROUP BY ProjectTitle
458 HAVING Avg(Hoursworked)>=15;
459
460 /*accessing vwProjectHours*/
461 SELECT *
462 FROM vwProjectHours;
463
464
465 /*last line of question 13 is some confusing, write the select statement against this
view to display projects, by this if question mean just need the project names, this is
code of APPROACH TO THIS QUESTION*/
466 /*Befor that let's drop old one*/
467 DROP VIEW vwProjectHours;
468 CREATE VIEW vwProjectHours
469 (Project, AverageHours)AS
470 SELECT ProjectTitle, Avg(Hoursworked)
471 FROM ProjectAssignment pa JOIN Project p ON (pa.ProjectNumber=p.ProjectNumber)
472 GROUP BY ProjectTitle;
473
474 /*accessing vwProjectHours*/
475 SELECT Project
476 FROM vwProjectHours
477 WHERE AverageHours >=15;
478
479
480 /* THE */
END */
481
482 #####
483 #####
484 #####
485 #####

```

```

486  /*Sagar Kalauni*/
487  /* Lab8-Kalauni*/
488
489  /*Q.NO.1)The ProjectAssignment table stores data about the hours that employees are
working on specific projects. A senior project manager needs a listing of employee
names (first and last concatenated) who have not worked on projects 1, 2, 7 or 8. Use a
subquery approach and sort the rows of the result table by employee last name.*/
490  COLUMN "Employee" FORMAT A20;
491  SELECT LastName||', '||FirstName "Employee"
492  FROM Employee
493  WHERE EmployeeID NOT IN ( SELECT EmployeeID
494                           FROM ProjectAssignment
495                           WHERE ProjectNumber IN (1, 2, 7, 8))
496  ORDER BY LastName;
497
498
499  /*Q.NO.2)Management would like a report on all employees with a salary GREATER than the
MINIMUM salary of the employees in Department 3. Use a subquery approach and include
the employee first name, last name (concatenated), department number, and salary. Sort
the rows by departmentnumber. */
500  COLUMN "Employee" FORMAT A20;
501  COLUMN Salary FORMAT $99,999.99;
502  SELECT LastName||', '||FirstName "Employee", DepartmentNumber, Salary
503  FROM Employee
504  WHERE Salary > (SELECT MIN(Salary)
505                  FROM Employee
506                  WHERE DepartmentNumber=3)
507  ORDER BY DepartmentNumber;
508
509  /*Q.NO.3)Management is concerned that some employees are not putting in sufficient work
hours on assigned projects 1, 2, and 3. List the names of employees (last and first
concatenated) for those employees who worked on one of these three projects, but worked
fewer hours than the average number of hours worked on these three projects combined.
This is a nested subquery. Order the report by last name.*/
510  COLUMN "Employee Name" FORMAT A20;
511  SELECT LastName||', '||FirstName "Employee Name"
512  FROM Employee
513  WHERE EmployeeID IN ( SELECT EmployeeID
514                        FROM ProjectAssignment
515                        WHERE ProjectNumber IN (1,2,3) AND HoursWorked < (SELECT SUM(AVG(
516                                                                FROM
517                                                                ProjectAssignment
518                                                                WHERE
519                                                                ProjectNumber IN
520                                                                (1,2,3)
521                                                                GROUP BY
522                                                                ProjectNumber))
523  ORDER BY LastName;
524
525  /*Q.NO.4)The previous report has piqued the project manager's curiosity. He would now
like a report that lists all employees who have worked fewer hours than the average for
all projects combined. Computer the average hours worked on a project in a subquery.
In the report, list the employee's first and last name. */
526  COLUMN "Employee Name" FORMAT A20;
527  SELECT LastName||', '||FirstName "Employee Name"
528  FROM Employee
529  WHERE EmployeeID IN ( SELECT EmployeeID
530                        FROM ProjectAssignment
531                        WHERE HoursWorked < (SELECT SUM(AVG(HoursWorked))
532                                                                FROM ProjectAssignment
533                                                                GROUP BY ProjectNumber));
534
535  /*Q.NO.5)The Chief Nurse would like a list of all room numbers that have a bed with a
description that has Surgical anywhere in the field and are available. The result field
should list the room */
536  COLUMN "Room Number" FORMAT A12;
537  SELECT RoomNumber "Room Number"
538  FROM Bed

```

```

535 WHERE Availability='Y' AND BedType IN (SELECT BedType
536                                         FROM BedClassification
537                                         WHERE Description LIKE '%Surgical%');
538
539 /*Q.NO.6)The project manager needs to know all available employees who can work on a new
project. He wants a report that lists the employee last name, first name, and department
name. But, he only wants the employees who are in the departments that are currently
working on projects 3 or 5. This query will require both a JOIN (in the outer query) and
a subquery. */
540 COLUMN "Last Name" FORMAT A12;
541 COLUMN "First Name" FORMAT A15;
542 SELECT LastName "Last Name", FirstName "First Name", d.DepartmentName "Department Name"
543 FROM Employee e LEFT OUTER JOIN Department d ON (e.DepartmentNumber=d.departmentnumber)
544 WHERE d.DepartmentNumber IN (SELECT DepartmentNumber
545                             FROM Project
546                             WHERE ProjectNumber IN (3,5));

547
548 /*Q.NO.7)The head of Pediatrics would like a list of all current patients in Pediatrics
beds. Pediatric beds will always have PED somewhere in the roomnumber field. The
hospital may add more pediatric beds in the future. Create the list by using a
subquery. Format patient names with one heading.*/
549 COLUMN "Patient Name" FORMAT A20;
550 SELECT LastName||', '||FirstName "Patient Name"
551 FROM Patient
552 WHERE Bedno IN (SELECT BedNumber
553                FROM Bed
554                WHERE RoomNumber LIKE '%PED%');
555
556 /*Q.NO.8)Provide the treatment number, patientId, and employeeID of all treatments with
a service Category description of 'Surgery' and have a standardCharge greater than
$5,000. This is a nested subquery. Order the report by treatment number. */
557 COLUMN "Patient ID" FORMAT A10;
558 COLUMN "Employee ID" FORMAT A12;
559 SELECT treatmentNumber "Treatment Number", PatientID "Patient ID", EmployeeID "Employee
ID"
560 FROM Treatment
561 WHERE ServiceID IN (SELECT ServiceID
562                    FROM Service
563                    Where StandardCharge>5000 AND CategoryID IN (SELECT CategoryID
564                                                                FROM ServiceCategory
565                                                                WHERE Description LIKE
566                                                                '%Surgery%'))
567
568 ORDER BY treatmentNumber;
569
570 /*Q.NO.9)The Hospital Chief has requested a list of employees whose salary is less than
all employee salaries in Department 2. The result should not include any employee who
has a wage rate rather than a salary (i.e. salary should not be null). Use a subquery
approach and the ALL function.*/
571 COLUMN "Employee" FORMAT A20;
572 COLUMN Salary FORMAT $999,999.99;
573 SELECT LastName||', '||FirstName "Employee", Salary, DepartmentNumber
574 FROM employee
575 WHERE salary < ALL (SELECT Salary
576                    FROM employee
577                    WHERE DepartmentNumber = 2 AND salary IS NOT NULL);
578
579 /*
580 THE
581 */
582 #####
583 #####
584 #####
585 #####
586
587 /*Sagar Kalauni*/

```

```

585  /* Lab10-Kalauni*/
586
587  /*Q.No.1)Management requires a listing of employees by last name, first name, and middle
initial for department number 8. The last name should be displayed in all capital
letters. The entire name should be concatenated together so as to display in a single
field with a column heading of "Employee Name." The rows should be sorted by employee
last name, then employee first name. */
588  COLUMN "Employee Name" FORMAT A22;
589  SELECT FirstName ||' '|| SUBSTR(MiddleName, 1,1) ||' '|| UPPER(LastName) "Employee
Name"
590  FROM Employee
591  WHERE DepartmentNumber=8
592  ORDER BY LastName, FirstName;
593
594  /*Q.No.2)Write a query that displays the department name and the length in number of
characters of each department's name. Use the department table. Label the column
headings appropriately*/
595  COLUMN "Dept. Name" FORMAT A26;
596  COLUMN "Length" FORMAT 999999;
597  SELECT DepartmentName "Dept. Name", LENGTH(DepartmentName) "Length"
598  FROM Department;
599
600  /*Q.No.3)Management wants a listing of department numbers and names (use the department
table)–display the output as a single column with the heading "Department Information" –
convert the DepartmentNumber column to character data as part of the query. */
601  COLUMN "Department Information" FORMAT A32;
602  SELECT To_CHAR(DepartmentNumber, '999')||' '|| DepartmentName "Department Information"
603  FROM Department;
604
605  /*Q.No.4)Write a query that displays the first four characters of each employee's last
name and the last four digits of each employee's SSN for department 8. Label the column
headings "Name" and "SSN." Order the result table rows by employee last name. */
606  COLUMN "Name" FORMAT A5;
607  COLUMN "SSN" FORMAT A5;
608  SELECT SUBSTR(LastName, 1,4) "Name", SUBSTR(SSN,6) "SSN"
609  FROM Employee
610  WHERE DepartmentNumber=8
611  ORDER BY LastName;
612
613  /*Q.No.5)Write a query that displays all employee names as well as their work phone.
Format their workphone to 999-999-9999.*/
614  COLUMN "Employee Name" FORMAT A25;
615  COLUMN "Phone No." FORMAT A18;
616  SELECT LastName ||', '|| FirstName "Employee Name", SUBSTR(WorkPhone,1,3) ||'- '||SUBSTR(
WorkPhone,4,3) ||'- '||SUBSTR(WorkPhone,7,4) "Phone No."
617  FROM Employee;
618
619  /*Q.No.6)Write a query to display a listing of employee last names and the EmployeeID of
each employee's supervisor for the employees working in department 8. If the employee
has no supervisor, display the message "Top Supervisor." Provide appropriate headings.
Sort the result table by employee last name. */
620  COLUMN "Emp Last Name" FORMAT A14;
621  COLUMN "EmployeeID" FORMAT A18;
622  SELECT LastName "Emp Last Name", DECODE(SupervisorID, NULL, 'Top Supervisor' , EmployeeID
) "EmployeeID"
623  FROM Employee
624  WHERE DepartmentNumber=8
625  ORDER BY LastName;
626
627  /*If there was typo in the question for supervisorid*/
628  COLUMN "Emp Last Name" FORMAT A14;
629  COLUMN "SupervisorID" FORMAT A18;
630  SELECT LastName "Emp Last Name", NVL(SupervisorID, 'Top Supervisor') "SupervisorID"
631  FROM Employee
632  WHERE DepartmentNumber=8
633  ORDER BY LastName;
634
635  /*Q.No.7)Develop a listing for the company's senior project manager that lists employees
that reported working between 15 and 25 hours (inclusive) on assigned projects. List

```

```

the employee last name, project number, and hours worked. Use the ABS function. Do NOT
use the BETWEEN operator or any logical operator. Join the tables by use of the FROM
clause. Use meaningful column headings. Sort the rows of the result table by employee
last name. HINT: 20 is the midpoint between 15 and 25 hours, the absolute value of the
difference would be 5 hours. */
636 COLUMN "Emp Last Name" FORMAT A14;
637 COLUMN "Project No." FORMAT 99;
638 COLUMN "Hours worked" FORMAT 99.9;
639 SELECT LastName "Emp Last Name", ProjectNumber "Project No.", HoursWorked "Hours worked"
640 FROM Employee e JOIN ProjectAssignment pa ON (e.employeeId=pa.employeeId)
641 WHERE ABS(HoursWorked -20) <=5
642 ORDER By LastName;
643
644 /*Q.No.8)The senior project manager needs a listing by employee last name, project
number, and hours worked (HoursWorked column) rounded to the nearest integer value for
projects 3 and 8. Join the tables by use of the FROM clause. Sort the result table by
employee last name within project number. Use meaningful column names. */
645 COLUMN "Emp Last Name" FORMAT A14;
646 COLUMN "Project No." FORMAT 99;
647 SELECT LastName "Emp Last Name", ProjectNumber "Project No.", ROUND(HoursWorked,0)
648 FROM Employee e JOIN ProjectAssignment pa ON (e.employeeId=pa.employeeId)
649 WHERE ProjectNumber IN (3,8)
650 ORDER BY ProjectNumber, LastName;
651
652 /*Q.No.9)Write a query to display information about female dependents for the human
resources manager. Display each dependent's name, gender, and date of birth. The date
of birth should be displayed as: Month Name (spelled out), two-digit day, and
four-digit year (e.g., December 05, 1970). Use the COLUMN commands shown here to format
the first two columns of output. */
653 COLUMN "Gender" Format A6;
654 COLUMN "Dep Name" FORMAT A15;
655 SELECT Name "Dep Name", Gender "Gender", TO_CHAR(BirthDate, 'Month DD, YYYY') "Date of
Birth"
656 FROM Dependent
657 WHERE Gender= 'F';
658
659 /*Q.No.10)Write a query to display each dependent's name, date of birth, and date on
which the dependent turned or will turn 65 years of age, but only for dependents born
after January 1, 1980. Use meaningful column names. Display each date using the
DD-MON-YYYY format. Use the ADD_MONTHS, TO_CHAR, and TO_DATE functions. Hint: 65 years
equals 780 months. */
660 SELECT Name "Dep Name", TO_CHAR(BirthDate, 'DD-MON-YYYY') "Date of Birth", TO_CHAR(
ADD_MONTHS(BirthDate, 780), 'DD-MON-YYYY') "65th Birthday"
661 FROM Dependent
662 WHERE BirthDate > TO_DATE('01-JAN-1980', 'DD-MON-YYYY');
663
664 /*Q.No.11)Write a short query to display the current day of the week spelled out, for
example Monday or Wednesday. The value should be obtained from the operating system
internal date.*/
665 SELECT TO_CHAR(SYSDATE, 'DAY') "Current Day"
666 FROM Dual;
667
668 /*Q.No.12)The human resources manager needs a listing of dependents including their name
and gender, but only for dependents that are spouses. Instead of displaying the coded
values for gender, the result table must display the terms "Male" and "Female," as
appropriate. Use meaningful column headings. Sort the result table by dependent name.
*/
669 SELECT Name "Dep Name", DECODE(Gender, 'M', 'Male', 'Female') "Gender"
670 FROM Dependent
671 WHERE RelationshipToEmployee= 'SPOUSE'
672 ORDER BY Name;
673
674 /*Q.No.13)Write a query to display a listing of employee last names, title, and salary
for employees with a title of either 'Building Custodian' or 'L.P.N.'. If the employee
is paid a wage, the salary will be NULL. In this situation, display the value $0.00.
Provide appropriate headings. Sort the result table by staff member last name. */
675 COLUMN "Salary" FORMAT $999,990.99; --by giving tailing term as zero
676 COLUMN "Emp Last Name" FORMAT A15;
677 SELECT LastName "Emp Last Name", Title, NVL(Salary, '0.00') "Salary"

```



```

678 FROM Employee
679 WHERE Title LIKE '%Building Custodian%' OR Title LIKE '%L.P.N%'
680 ORDER BY LastName;
681
682
683 /*This Can be done alternatively by this way */
684 COLUMN "Emp Last Name" FORMAT A15;
685 COLUMN "Salary" FORMAT $99,999.99;
686 COLUMN "Title" FORMAT A20;
687 SELECT LastName "Emp Last Name", Title "Title", DECODE(TO_CHAR(Salary, '$9,999'), NULL,
' $0.00', TO_CHAR(Salary, '$9,999.99')) "Salary"
688 FROM Employee
689 WHERE Title LIKE '%Building Custodian%' OR Title LIKE '%L.P.N%'
690 ORDER BY LastName;
691
692 /*Q.No.14)Write a query to compare treatment charges to standard service charges.
Display the ServiceID and StandardCharge columns from the service table, and the
ChargeAmount column from the treatment table. Only display the value if the difference
between the service and actual charge is more than $50.00 in difference (either high or
low). Also display a computed column that is the difference between the service and
actual charge (as a positive number). Use appropriate column sizes and headings. Join
the tables by use of the FROM clause. */
693 COLUMN "Service ID"FORMAT A12;
694 COLUMN "Service Charge" FORMAT $999,999.99;
695 COLUMN "Treatment Charge" FORMAT $99,999.99;
696 COLUMN "Difference" FORMAT $999,999.99;
697 SELECT s.ServiceID "Service ID", s.StandardCharge "Service Charge", t.ChargeAmount
"Treatment Charge", ABS(StandardCharge - ChargeAmount) "Difference"
698 FROM Service s JOIN Treatment t ON (s.serviceID=t.serviceID)
699 WHERE ABS(s.StandardCharge - t.ChargeAmount)> 50;
700
701 /*Q.No.15)Modify the query for question 14 to display any rows where there is any
difference between the service and actual charge, but only where the actual charge is
less than the service charge. Order the output by differences from largest to
smallest.*/
702 COLUMN "Service ID"FORMAT A12;
703 COLUMN "Service Charge" FORMAT $999,999.99;
704 COLUMN "Actual Charge" FORMAT $999,999.99;
705 COLUMN "Difference" FORMAT $999,999.99;
706 SELECT s.ServiceID "Service ID", s.StandardCharge "Service Charge", t.ChargeAmount
"Actual Charge", ABS(StandardCharge - ChargeAmount) "Difference"
707 FROM Service s JOIN Treatment t ON (s.serviceID=t.serviceID)
708 WHERE ChargeAmount<StandardCharge
709 ORDER BY ABS(StandardCharge - ChargeAmount) DESC;
710
711 /* THE */
712 END */
713
714 /*Solution for Q.Qn.3*/
715 -- Program: Q.No.-3.sql
716 -- Programmer: Sagar Kalauni
717 -- Description: Information about employee's Dependent
718
719 TTITLE CENTER 'Employee Name:' EmployeeNameVar -
720 RIGHT 'Page: ' FORMAT 99 sql.pno SKIP 2
721 BTITLE SKIP 1 CENTER 'Not for external dissemination.'
722 SET LINESIZE 55
723 SET PAGESIZE 24
724 SET NEWPAGE 1
725
726 -- Define employee Variable
727 --COLUMN e.Lastname||', '||e.Firstname NEW_VALUE EmployeeNameVar NOPRINT
728
729 COLUMN "Emp ID" FORMAT A6;
730 COLUMN "Dependent Name" FORMAT A15;
731 COLUMN "Gender" FORMAT A6;
732 COLUMN "Date Birth" FORMAT A10;
733 COLUMN "Relationship" FORMAT A12;

```

```

734
735
736 CREATE OR REPLACE VIEW vwEmpDep (Employee, EmployeeID, Name, gender, BirthDate,
RelationshipToEmployee) AS
737 SELECT e.Lastname||', '||e.Firstname, d.EmployeeID "Emp ID", d.Name "Dependent Name", d.
gender "Gender", d.BirthDate "Date Birth", d.RelationshipToEmployee "Relationship"
738 FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID);
739
740
741 COLUMN Employee NEW_VALUE EmployeeNameVar NOPRINT
742 -- Set column sizes based on alias column names
743 COLUMN "Emp ID" FORMAT A6;
744 COLUMN "Dependent Name" FORMAT A15;
745 COLUMN "Gender" FORMAT A6;
746 COLUMN "Date Birth" FORMAT A10;
747 COLUMN "Relationship" FORMAT A12;
748
749 --CLEAR BREAK
750 BREAK ON Name ON PAGE
751 COMPUTE COUNT OF "Dependent" ON REPORT
752
753 SELECT Employee, EmployeeID, Name, gender, RelationshipToEmployee
754 FROM vwEmpDep;
755 /*SELECT d.EmployeeID "Emp ID", d.Name "Dependent", d.gender "Gender", d.BirthDate "Date
Birth", d.RelationshipToEmployee "Relationship"
756 FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
757 ORDER BY d.EmployeeID;*/
758
759 #####
#####
760 #####
#####
761
762 /*Sagar Kalauni*/
763 /* Lab9-Kalauni*/
764
765 /*Solution for Q.Qn.1*/
766 -- Program: Q.No.-1.sql
767 -- Programmer: Sagar Kalauni
768 -- Description: Information about employee's Dependent
769
770 TTITLE 'Dependent Information'
771 BTITLE SKIP 2 CENTER 'Not for external dissemination.'
772 SET LINESIZE 55
773 SET PAGESIZE 24
774 SET NWEPPAGE 1
775
776 COLUMN "Emp ID" FORMAT A6;
777 COLUMN "Dependent" FORMAT A15;
778 COLUMN "Gender" FORMAT A6;
779 COLUMN "Date Birth" FORMAT A10;
780 COLUMN "Relationship" FORMAT A12;
781 SELECT EmployeeID "Emp ID", Name "Dependent", gender "Gender", BirthDate "Date Birth",
RelationshipToEmployee "Relationship"
782 FROM Dependent
783 ORDER BY EmployeeID;
784
785 /*Solution for Q.Qn.2*/
786 -- Program: Q.No.-2.sql
787 -- Programmer: Sagar Kalauni
788 -- Description: Information about employee's Dependent
789
790 TTITLE 'Dependent Information'
791 BTITLE SKIP 1 CENTER 'Not for external dissemination.'
792 SET LINESIZE 55
793 SET PAGESIZE 24
794 SET NWEPPAGE 1
795
796 COLUMN "Emp ID" FORMAT A6;

```

```

797 COLUMN "Dependent" FORMAT A15;
798 COLUMN "Gender" FORMAT A6;
799 COLUMN "Date Birth" FORMAT A10;
800 COLUMN "Relationship" FORMAT A12;
801
802 CLEAR BREAK
803 BREAK ON "Emp ID" SKIP 2 ON REPORT
804 COMPUTE COUNT OF "Emp ID" ON REPORT
805
806 SELECT EmployeeID "Emp ID", Name "Dependent", gender "Gender", BirthDate "Date Birth",
RelationshipToEmployee "Relationship"
807 FROM Dependent
808 ORDER BY EmployeeID;
809
810 /*Solution for Q.Qn.3*/
811 -- Program: Q.No.-3.sql
812 -- Programmer: Sagar Kalauni
813 -- Description: Information about employee's Dependent
814
815 /*If done this way perfectly works for first table*/
816
817 TTITLE CENTER 'Employee Name:' EmployeeNameVar -
818 RIGHT 'Page: ' FORMAT 99 sql.pno SKIP 2
819 BTITLE SKIP 1 CENTER 'Not for external dissemination.'
820 SET LINESIZE 65
821 SET PAGESIZE 15
822 SET NEWPAGE 1
823
824 -- Create a view to use in the Select command
825 CREATE OR REPLACE VIEW vwEmpDep ( Employee, "Emp ID", DependentName, gender, Relationship
) AS
826 SELECT e.Lastname||', '||e.Firstname, d.EmployeeID, d.Name, d.gender, d.
RelationshipToEmployee
827 FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
828 ORDER BY -DependentName;
829
830 COLUMN Employee NEW_VALUE EmployeeNameVar NOPRINT
831 COLUMN "Emp ID" FORMAT A20
832 COLUMN DependentName FORMAT A15
833 COLUMN gender FORMAT A6
834 COLUMN Relationship FORMAT A12
835
836 BREAK ON Employee ON "Emp ID" PAGE;
837 COMPUTE COUNT LABEL "Number of Dependents" OF Relationship ON "Emp ID"
838
839 SELECT Employee, "Emp ID", DependentName, gender, Relationship
840 FROM vwEmpDep
841 ORDER BY "Emp ID" , relationship desc;
842
843 /*If done this way perfectly works for last table*/
844
845 TTITLE CENTER 'Employee Name:' EmployeeNameVar -
846 RIGHT 'Page: ' FORMAT 99 sql.pno SKIP 2
847 BTITLE SKIP 1 CENTER 'Not for external dissemination.'
848 SET LINESIZE 65
849 SET PAGESIZE 15
850 SET NEWPAGE 1
851
852 -- Create a view to use in the Select command
853 CREATE OR REPLACE VIEW vwEmpDep ( Employee, "Emp ID", DependentName, gender, Relationship
) AS
854 SELECT e.Lastname||', '||e.Firstname, d.EmployeeID, d.Name, d.gender, d.
RelationshipToEmployee
855 FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
856 ORDER BY -DependentName;
857
858 COLUMN Employee NEW_VALUE EmployeeNameVar NOPRINT
859 COLUMN "Emp ID" FORMAT A20
860 COLUMN DependentName FORMAT A15

```

```

861 COLUMN gender FORMAT A6
862 COLUMN Relationship FORMAT A12
863
864 BREAK ON Employee ON "Emp ID" PAGE;
865 COMPUTE COUNT LABEL "Number of Dependents" OF Relationship ON "Emp ID"
866
867 SELECT Employee, "Emp ID", DependentName, gender, Relationship
868 FROM vwEmpDep
869 ORDER BY "Emp ID" , relationship;
870
871 /*I spent a lot of time for this question, I don't find a way to exactly shows ordering,
But I think ordering does not matter here, because we do not have complete report photo
in question*/
872
873
874
875 #####
876 #####
877 #####
878
879 /*If done this way perfectly works for last table*/
880 TTITLE CENTER 'Employee Name:' EmployeeNameVar -
881 RIGHT 'Page: ' FORMAT 99 sql.pno SKIP 2
882 BTITLE SKIP 1 CENTER 'Not for external dissemination.'
883 SET LINESIZE 65
884 SET PAGESIZE 15
885 SET NEWPAGE 1
886
887 -- Create a view to use in the Select command
888 CREATE OR REPLACE VIEW vwEmpDep ( Employee, "Emp ID", DependentName, gender, Relationship
) AS
889 SELECT e.Lastname||', '||e.Firstname, d.EmployeeID, d.Name, d.gender, d.
RelationshipToEmployee
890 FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
891 ORDER BY -DependentName;
892
893 COLUMN Employee NEW_VALUE EmployeeNameVar NOPRINT
894 COLUMN "Emp ID" FORMAT A20
895 COLUMN DependentName FORMAT A15
896 COLUMN gender FORMAT A6
897 COLUMN Relationship FORMAT A12
898
899 BREAK ON Employee ON "Emp ID" PAGE;
900 COMPUTE COUNT LABEL "Number of Dependents" OF Relationship ON "Emp ID"
901
902 SELECT Employee, "Emp ID", DependentName, gender, Relationship
903 FROM vwEmpDep
904 ORDER BY "Emp ID" , relationship;
905
906 /*If done this way perfectly works for first table*/
907
908 TTITLE CENTER 'Employee Name:' EmployeeNameVar -
909 RIGHT 'Page: ' FORMAT 99 sql.pno SKIP 2
910 BTITLE SKIP 1 CENTER 'Not for external dissemination.'
911 SET LINESIZE 65
912 SET PAGESIZE 15
913 SET NEWPAGE 1
914
915 -- Create a view to use in the Select command
916 CREATE OR REPLACE VIEW vwEmpDep ( Employee, "Emp ID", DependentName, gender, Relationship
) AS
917 SELECT e.Lastname||', '||e.Firstname, d.EmployeeID, d.Name, d.gender, d.
RelationshipToEmployee
918 FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
919 ORDER BY -DependentName;
920
921 COLUMN Employee NEW_VALUE EmployeeNameVar NOPRINT

```

```
922 COLUMN "Emp ID" FORMAT A20
923 COLUMN DependentName FORMAT A15
924 COLUMN gender FORMAT A6
925 COLUMN Relationship FORMAT A12
926
927 BREAK ON Employee ON "Emp ID" PAGE;
928 COMPUTE COUNT LABEL "Number of Dependents" OF Relationship ON "Emp ID"
929
930 SELECT Employee, "Emp ID", DependentName, gender, Relationship
931 FROM vwEmpDep
932 ORDER BY "Emp ID" , relationship desc;
933
934
935 /*I spent a lot of time for this question, I don't find a way to exactly shows ordering,
But I think ordering does not matter here, because we do not have complete report photo
in question*/
936
937
938 #####
939 #####
940 #####
941 #####
942 #####
943 #####
944 #####
945 #####
946 #####
947 #####
948 #####
949 #####
950 #####
951 #####
952 #####
953 #####
954 #####
955 #####
956 #####
957 #####
958 #####
959 #####
960 #####
961 #####
962 #####
963 #####
964 #####
965 #####
966 #####
967 #####
968 #####
969 #####
970 #####
971 #####
972 #####
973 #####
974 #####
975 #####
976 #####
977 #####
978 #####
979 #####
980 #####
981 #####
982 #####
983 #####
984 #####
985 #####
986 #####
```