```
SQL> /* CLARA CHEROTICH CHELIBEI */
SQL> /* LAB1-CHELIBEI */
SQL>
SQL> /* Q.no.1 */
SQL> SELECT *    -- selecting everything
  2  FROM ROOM
  3  WHERE RoomNumber LIKE 'RA%';  -- restricting few things
  4
SQL> /* Q.no.2 */
SQL> SELECT TreatmentNumber, DateTreated, EmployeeID --selecting only
specified columns
  2  FROM Treatment
  3  WHERE EmployeeID IN ('88202', '23244') AND TreatmentNumber < 3;  --with
restrections
  4
SQL> SPOOL OFF;
```

```
SQL> /* CLARA CHEROTICH CHELIBEI */
SQL> /* LAB2-CHELIBEI */
SQL> /* Q.no.1 * 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 */
SQL> CREATE TABLE prod_table ( prod_id   NUMBER (3),
  2  prod_description  VARCHAR2 (25));

Table created.

SQL> /* Displays created table */
SQL>
SQL> /* Q.no.2 Insert 2 rows into the test_table */
SQL> INSERT INTO prod_table VALUES (1, 'Wheel');

1. row created.

SQL> INSERT INTO prod_table VALUES (2, 'Nuts and Bolts');

1. row created.

SQL> /* the 2 rows has been created into 'prod_table' */
SQL>
SQL>  /* Q.no.3 Use DESCRIBE to describe the prod_table */
SQL> DESC prod_table;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------
----
 PROD_ID                                            NUMBER(3)
 PROD_DESCRIPTION                                   VARCHAR2(25)

SQL>  /*The output displays the 'prod_table' with its columns their data
types.*/
SQL>
SQL> /* Q.no.4 Use the following SELECT command to display the rows in the
prod_table.*/
SQL> SELECT * FROM prod_table;

   PROD_ID PROD_DESCRIPTION
---------- -------------------------
         1. Wheel
         2. Nuts and Bolts

SQL> /*output shows columns of the 'prod_table.'*/
SQL>
SQL>  /* Q.no.5 Use the DROP command to drop the prod_table.*/
SQL> DROP TABLE prod_table;

Table dropped.

SQL> /* table successfully dropped */
SQL>
SQL> /* Q.no.6 Create the deptBusiness table described below:*/
SQL> CREATE TABLE DeptBusiness (
```

```
  3. DepartmentNumber      NUMBER(4)
  4. CONSTRAINT PK_DeptBusiness PRIMARY KEY,
  5. DepartmentName        VARCHAR2(25)
  6. CONSTRAINT NN_DepartmentNamee NOT NULL,
  7. ManagerID             CHAR(5)
  8. );
CREATE TABLE DeptBusiness (
             *
ERROR at line 1:
ORA-00955: name is already used by an existing object


SQL> CREATE TABLE DeptBusiness (
  9. DepartmentNumber      NUMBER(4)
  10.CONSTRAINT PK_DeptBusiness PRIMARY KEY,
  11.DepartmentName        VARCHAR2(25)
  12.CONSTRAINT NN_DepartmentNamee NOT NULL,
  13.ManagerID             CHAR(5)
  14.);
CREATE TABLE DeptBusiness (
             *
ERROR at line 1:
ORA-00955: name is already used by an existing object


SQL> DROP TABLE DeptBusiness;

Table dropped.

SQL>  CREATE TABLE DeptBusiness (
  15.DepartmentNumber      NUMBER(4)
  16.CONSTRAINT PK_DeptBusiness PRIMARY KEY,
  17.DepartmentName        VARCHAR2(25)
  18.CONSTRAINT NN_DepartmentNamee NOT NULL,
  19.ManagerID             CHAR(5)
  20.);

Table created.

SQL> /* Q.no.7 Add the data shown below to the deptBusiness table.  Do not
define any foreign keys.  Leave the ManagerID column values as NULL.*/
SQL> INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
  21.VALUES (1106, 'CMIS');

1. row created.

SQL> INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
  2. VALUES (1105, 'Accounting');

1. row created.

SQL>  INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
  2. VALUES (1100, 'Production');

1. row created.

SQL>  INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
```

```
  2. VALUES (1102, 'Economic Finance');

1. row created.

SQL> /* all values added to deptBusiness table*/
SQL>
SQL> /*Q.no.8 a. COMMIT your row insertions in the deptBusiness table. (SQL>
COMMIT;)
SQL>  b. Try to INSERT the data for department number 1106 again in the
deptBusiness table. Did Oracle accept i
SQL> COMMIT;
SQL> COMMIT;
SQL> COMMIT;
SQL>
SQL> COMMIT;
SQL> COMMIT TABLE DeptBusiness;
SQL> SELECT @@TRANCOUNT;
SQL>
SQL>
SQL> COMMIT
SQL> COMMIT;
SQL> INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
SQL> INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
SQL> VALUES (1106, 'CMI
SQL> INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
SQL> VALUES (1106, 'CMI
SQL> COMMIT;
SQL>
SQL> COMMIT;
SQL>
SQL>
SQL>
SQL>
SQL> COMMIT;

Commit complete.

SQL> /* all changes successfully applied*/
SQL> INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
  2. VALUES (1106, 'CMIS');
INSERT INTO DeptBusiness (DepartmentNumber, DepartmentName)
*
ERROR at line 1:
ORA-00001: unique constraint (USER2.PK_DEPTBUSINESS) violated


SQL> /*value is not accepted error: Unique constraint violated*/
SQL>
SQL> /* Q.no.9 Use the following SELECT command to display the rows in the
deptBusiness table.*/
SQL> SELECT * FROM DeptBusiness;

DEPARTMENTNUMBER DEPARTMENTNAME            MANAG
---------------- ------------------------- -----
            1106 CMIS
            1105 Accounting
            1100 Production
```

```
            1102 Economic Finance

SQL> /*displays the 'DeptBusiness' table with all  data entries within it.*/
SQL> /* Q.no.10 Delete the row for department number 1 from the deptBusiness
table.   (HINT:  READ the relevant part of Chapter 2 (SQL Example 2.23 and
SQL Example 2.24) to compose your DELETE query correctly. Include the
following WHERE clause in your DELETE statement: WHERE  DepartmentNumber =
1106).*/
SQL>  DELETE FROM DeptBusiness
  3. WHERE DepartmentNumber= 1106;

1. row deleted.

SQL> /*A single row has been removed from our 'DeptBusiness' table.*/
SQL>
SQL>  /*Q.no.11 Repeat the SELECT statement in question #9 above to verify
your record has been deleted.*/
SQL>  SELECT * FROM DeptBusiness;

DEPARTMENTNUMBER DEPARTMENTNAME            MANAG
---------------- ------------------------- -----
            1105 Accounting
            1100 Production
            1102 Economic Finance

SQL> /* Q.no.12 Assume that the deletion of your row was an error.  Execute
the ROLLBACK command (SQL> ROLLBACK;) to undelete your row (Note: Do not
simply reinsert the row to the table).  Use the SELECT * statement again to
verify that your row has been restored to the table.*/
SQL> ROLLBACK;

Rollback complete.

SQL> /*This SQL command undeletes the row that had been deleted previously
'DeptBusiness' table.*/
SQL> SELECT * FROM DeptBusiness;

DEPARTMENTNUMBER DEPARTMENTNAME            MANAG
---------------- ------------------------- -----
            1106 CMIS
            1105 Accounting
            1100 Production
            1102 Economic Finance

SQL> /*Output  shows that the deleted row has been sucessfully retrived back
after it had been deleted*/
SQL>  /* Q.no.13 The name for âÄòProductionâÄô department got changed to
âÄòOperationsâÄô. Update the DepartmentName column of this change
accordingly.  (HINT:  READ the relevant part of Chapter 2 (SQL Example 2.26
and SQL Example 2.27) to compose your UPDATE query correctly. Do not forget
to include the single quotes in the department name values as these are
character data). Repeat the SELECT statement in question #9 above to verify
your output.*/
SQL> UPDATE DeptBusiness
  2. SET DepartmentName = 'Operations'
  3. WHERE DepartmentName= 'Production';
```

```
1. row updated.

SQL> /*The 'DepartmentName' has been updated from 'Production' to
'Operations'*/
SQL>  SELECT * FROM DeptBusiness;

DEPARTMENTNUMBER DEPARTMENTNAME            MANAG
---------------- ------------------------- -----
            1106 CMIS
            1105 Accounting
            1100 Operations
            1102 Economic Finance

SQL>  /*The update department name can be seen in the output.*/
SQL> /* Q.no.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. (HINT:  READ the relevant part of Chapter 2 with
example to compose your ALTER command correctly). Repeat the SELECT
statement in question #9 above to verify your output.*/
SQL> ALTER TABLE DeptBusiness ADD (DepartmentPhone  NUMBER (10));

Table altered.

SQL>  SELECT * FROM DeptBusiness;

DEPARTMENTNUMBER DEPARTMENTNAME            MANAG DEPARTMENTPHONE
---------------- ------------------------- ----- ---------------
            1106 CMIS
            1105 Accounting
            1100 Operations
            1102 Economic Finance

SQL> /*Output shows newly added column 'DepartmentPhone' in the table
DeptBusiness*/
SQL>  /* Q.no.15 Use the DROP command to drop the DeptBusiness table. Use
the SELECT statement given in question #9 above to display the deptBusiness
table.  You should get an error message as the table no longer exists?*/
SQL> /*Table sucessfully droped*/
SQL>  SELECT * FROM DeptBusiness;

DEPARTMENTNUMBER DEPARTMENTNAME            MANAG DEPARTMENTPHONE
---------------- ------------------------- ----- ---------------
            1106 CMIS
            1105 Accounting
            1100 Operations
            1102 Economic Finance

SQL> /*error message appears, because the Table was already droped, the
output is: table does not exist*/
SQL> SPOOL OFF;
```

```
SQL> /*CLARA CHEROTICH CHELIBEI*/
SQL> /* LAB4-CHELIBEI*/
SQL> /* Q.no.1 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 */
SQL> /* I want to examine the specialty of my table */
SQL> DESC Specialty;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------
----
 SPECIALTYID                               NOT NULL CHAR(6)
 TITLE                                     NOT NULL VARCHAR2(50)
 AWARDEDBY                                          VARCHAR2(100)

SQL> /* it displays the columns of the specialty table */
SQL> SELECT SPECIALTYID,TITLE, AWARDEDBY FROM Specialty;

SPECIA TITLE
------ --------------------------------------------------
AWARDEDBY
--------------------------------------------------------------------------------
----
OPT    Optometrist
Complete certified program of instruction for Optometry.

ONC    Oncologist
Complete Medical Doctor of Oncology board certification.

RAD    Radiologist
Complete Medical Doctor of Radiology board certification.


SPECIA TITLE
------ --------------------------------------------------
AWARDEDBY
--------------------------------------------------------------------------------
----
CAR    Cardiologist
Complete Medical Doctor of Cardiology board certification.

GYN    Gynecologist
Complete Medical Doctor of Gynecology board certification.

GMD    General Practitioner
Complete Medical School.


SPECIA TITLE
------ --------------------------------------------------
AWARDEDBY
--------------------------------------------------------------------------------
----
SU1    Surgeon-Thoracic
Complete Thoracic Surgeon board certification.

SU2    Surgeon-General
Complete General Surgeon board certification.
```

```
SU3    Neurosurgeon
Complete Neurosurgery board certification.


SPECIA TITLE
------ --------------------------------------------------
AWARDEDBY
--------------------------------------------------------------------------------
----
PED    Pediatrician
Complete Medical Doctor of Pediatrics board certification.

SU4    Surgeon-Abdominal Cavity
Complete Abdominal Surgical Procedures board certification.

RN1    Registered Nurse
Complete Registered Nurse board certification.


SPECIA TITLE
------ --------------------------------------------------
AWARDEDBY
--------------------------------------------------------------------------------
----
LPN    Licensed Practicing Nurse
Complete Licensed Practicing Nurse board certification.

NPR    Nurse-Practitioner
Complete Nurse-Practitioner board certification.

RA2    Radiology Technologist
Complete program of instruction in radiology technology.


15 rows selected.

SQL> /* I can all the columns in the specialty table */
SQL> /* Q.no.2 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 */
SQL> SELECT distinct title FROM employee;

TITLE
--------------------------------------------------
Hospital Chief
M.D.
Pharmacist
M.D.-Chief of Surgery
Records Clerk
Rad. Tech.
R.N.
V.P. Admin
Building Custodian
L.P.N.

10 rows selected.
```

```
SQL> /* We can see 10 different types of the employee titles in the employee
table */
SQL> /* Q.no.3 3.Execute a query that will display all treatment dates for
patient 100302.  Include the patientID, employeeID, and date treated */
SQL> /* need to do formating first */
SQL> COLUMN PatientID FORMAT A10;
SQL> COLUMN EmployeeID FORMAT A12;
SQL> COLUMN DateTreated FORMAT A12;
SQL> SELECT PatientID,EmployeeID, DateTreated FROM Treatment WHERE PatientID
='100302';

PATIENTID  EMPLOYEEID   DATETREATED
---------- ------------ ------------
100302     66427        22-MAY-24
100302     67585        22-MAY-24
100302     67585        22-MAY-24
100302     66444        22-MAY-24
100302     67585        22-MAY-24

SQL> /* it has displayed all the employees with PatientID 100302 */
SQL>
SQL>
SQL> /* Q.no.4 4.Execute a query that lists all the male nurses âÄì i.e.,
employees with a job title that includes the degree âÄòR.N..âÄô List each
employeeâÄôs 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 */
SQL> /* M-Male Gender*/
SQL> COLUMN LastName FORMAT A12;
SQL> COLUMN FirstName FORMAT A12;
SQL> COLUMN Title FORMAT A5;
SQL> COLUMN Gender FORMAT A6;
SQL> SELECT LastName,FirstName,Title,Gender FROM Employee WHERE Title like
'%R.N..%' AND Gender= 'M';

no rows selected

SQL> /* there is no row selected*/
SQL>
SQL>
SQL> /*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 */
SQL> SELECT FirstName,LastName,City WHERE City IN ('Alton', 'Collinsville')
ORDE by City;
SELECT FirstName,LastName,City WHERE City IN ('Alton', 'Collinsville') ORDE
by City
                               *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected
```

```
SQL> SELECT FirstName, LastName, City FROM Patient WHERE City IN
('Alton','Collinsville') ORDER BY City;

FIRSTNAME    LASTNAME     CITY
------------ ------------ -------------------------
Andrew       Able         Alton
Ronald       Howard       Alton
Barbara      Benton       Alton
Gretchen     Greathouse   Alton
Gregory      Grant        Alton
Harold       Harnett      Alton
Ivy          Iona         Alton
Juliet       Juneau       Alton
Keith        Kraut        Alton
Linda        Lima         Alton
Hank         Henderson    Alton

FIRSTNAME    LASTNAME     CITY
------------ ------------ -------------------------
Ilama        Ilama        Alton
Norman       November     Alton
Rudolph      Pappa        Alton
Renny        Reinhardt    Alton
Zina         Zenna        Alton
Albert       Algebra      Alton
Bradley      Beaufort     Alton
Danny        Dunland      Alton
David        Davis        Collinsville
Rue          Chen         Collinsville

21 rows selected.

SQL> /* OBSERVATIONS: There are 21 rows combined from both city Alton and
Collinsville. 19 are from city Alton specific and 2 from city Collinsville
specific */
SQL>
SQL> /* 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 */
SQL> /*taking a look at the table and resizing */
SQL> COLUMN Description FORMAT A12;
SQL> COLUMN 'Original Cost' FORMAT 9999.99;
SQL> COLUMN 'Quantity Available' FORMAT 9999;
SQL> COLUMN 'Project Number' FORMAT 99;
SQL> COLUMN 'Orginal Cost' FORMAT 9999.99;
SQL> SELECT Description, OriginalCost "Orginal Cost", QuantityAvailable
"Quantity Available", ProjectNumber "Project Number" FROM Equipment
  2  WHERE OriginalCost<500 AND (QuantityAvailable >=10 OR ProjectNumber=8);

DESCRIPTION  Orginal Cost Quantity Available Project Number
------------ ------------ ------------------ --------------
Tanks, Nitro       355.55                 10              2
us Oxide
```

```
Desk, Child        285.40                  6              8
Chair, Child        65.40                 12              8

SQL> /* the output is distributed perfectly*/
SQL>
SQL> /*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 */
SQL> COLUMN Description A30;
SP2-0158: unknown COLUMN option "A30"
SQL> SELECT Description, StandardCharge, CategoryID, FROM Service
  2  ORDER  BY CategoryID,StandardCharge DESC;
SELECT Description, StandardCharge, CategoryID, FROM Service
                                                *
ERROR at line 1:
ORA-00936: missing expression


SQL> SELECT Description, StandardCharge, CategoryID
  2  FROM Service
  3  ORDER BY CategoryID,StandardCharge DESC;

DESCRIPTION  STANDARDCHARGE CAT
------------ -------------- ---
EKG/Interp               85 CAR
Hep B 20-adu            195 INJ
lt

Hep B 0-19 V            185 INJ
FC

Hep A vaccin            175 INJ
e

Antibiotic I            110 INJ

DESCRIPTION  STANDARDCHARGE CAT
------------ -------------- ---
nj

Depo Provera             95 INJ
 Hormone tx

Prevnar Pedi             92 INJ
Pneumovax ad             88 INJ
ult

Depo Provera             85 INJ
 Contracept

DESCRIPTION  STANDARDCHARGE CAT
------------ -------------- ---
```

Therapeutic Inj	75	INJ
Vaccine Inj #2 + more	75	INJ
DPT-AC VFC	75	INJ
DPT-AC /HIB	75	INJ
dT Adult VFC	75	INJ
Varicella Va	65	INJ

DESCRIPTION	STANDARDCHARGE	CAT
c VFC		
HIB VFC	65	INJ
Fluvax	55	INJ
IPV VFC	55	INJ
MMR VFC	55	INJ
Allergy #2 + more	55	INJ
Vaccine Inj #1	45	INJ

DESCRIPTION	STANDARDCHARGE	CAT
Allergy #1	25	INJ
Complete Met abolic	115	LAB
Prenatal Pan el	110	LAB
Hgb A1C	95	LAB
Hepatic Func tion	95	LAB

DESCRIPTION	STANDARDCHARGE	CAT
TSH	90	LAB
PSA	85	LAB
Protime/INR	75	LAB
Arthritis Pa nel (RA, ANA , UA, ESR)	75	LAB
Pap Smear	75	LAB
General Pane l	55	LAB

DESCRIPTION	STANDARDCHARGE	CAT
Pathology-Ge	50	LAB

neral		
Lipid Panel	45	LAB
Throat Culture	45	LAB
Urine Culture	45	LAB

DESCRIPTION	STANDARDCHARGE	CAT
Basic Metabolic	35	LAB
SGOT	30	LAB
CBC	21	LAB
Emergency	155	OLA
After Hours	125	OLA
Comprehensive High	95	OLA
Detailed Pro	95	OLA

DESCRIPTION	STANDARDCHARGE	CAT
blem		
Comprehensive. Moderate	75	OLA
Expanded Problem	75	OLA
Problem Focused	55	OLA

DESCRIPTION	STANDARDCHARGE	CAT
Special Handling	35.75	OLA
Blood Draw	35.55	OLA
Hemoglobin	25	OLA
Blood Glucose	20.4	OLA
Hemocult	15.4	OLA
KOH	15	OLA
Wet Smear	15	OLA

DESCRIPTION	STANDARDCHARGE	CAT
Strep Screen	13.5	OLA
Urine/Micro	12.9	OLA
Prognosis, U	12	OLA

rine		
Urine/Dip	10.75	OLA
Spirometry	55	PRO
Audiometry	45	PRO
Tympanometry	40	PRO
Cerumen-oval	35	PRO
Burn Debride	35	PRO

DESCRIPTION	STANDARDCHARGE	CAT
ment		
Breathing TX	35	PRO
DRE	30	PRO
Pulse Oxygen	25	PRO
Anoscopy	21	PRO
Lumbar Spine (5 view)	675	RAD
Abdomen Obst Series	340	RAD

DESCRIPTION	STANDARDCHARGE	CAT
Foot (3 view)	325	RAD
Ankle (3 view)	325	RAD
Knee (3 view)	325	RAD
Wrist (3 vie	285	RAD

DESCRIPTION	STANDARDCHARGE	CAT
w min)		
Hand (3 view)	280	RAD
Calcaneus (2 view min)	275	RAD
Soft Tissue Neck	275	RAD

DESCRIPTION	STANDARDCHARGE	CAT
Hip (2 view min)	275	RAD
Clavicle (2	250	RAD

view)		
Shoulder (2 view min)	250	RAD
Abdomen (KUB)	240	RAD

DESCRIPTION	STANDARDCHARGE	CAT
CXR (2 view)	225	RAD
Elbow (2 vie w)	225	RAD
Toe (2 view min)	225	RAD
Finger (2 vi ew)	225	RAD

DESCRIPTION	STANDARDCHARGE	CAT
C-Spine (4 v iew min)	205	RAD
CXR (1 view)	170	RAD
Cranial	10000	SUR
Thoracic-Hea rt	9500	SUR
Abdominal-In testine	7800	SUR

DESCRIPTION	STANDARDCHARGE	CAT
Liver	7800	SUR
Kidneys	7500	SUR
Pancreatic	6500	SUR
Thoracic-Lun g	6500	SUR
Thoracic-Gen eral Explora tory	6200	SUR
Abdominal-Ge	6000	SUR

DESCRIPTION	STANDARDCHARGE	CAT
neral		
Spinal-Disc	3800	SUR
Spinal-Explo ratory	3500	SUR

```
Fracture-Com           2500 SUR
plex

Fracture-Sim           1500 SUR
ple

DESCRIPTION  STANDARDCHARGE CAT
------------ -------------- ---

Appendectomy            555 SUR
I and D Comp            320 SUR
. Multiple

I and D Simp            258 SUR
le

Cerumen-oval            230 SUR
 F.B.


DESCRIPTION  STANDARDCHARGE CAT
------------ -------------- ---
Skin Lesion             225 SUR
Paring

Nail Bed Des            185 SUR
truct

Nail Avulsio            175 SUR
n


105 rows selected.

SQL> /*It displays a table with 105 rows */
SQL>
SQL> SPOOL OFF;
```

```
SQL> /*CLARA CHEROTICH CHELIBEI*/
SQL> /* LAB5-CHELIBEI*/
SQL> /* 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*/
SQL> COLUMN Commonname FORMAT A20;
SQL> COLUMN ScientificName FORMAT A20;
SQL> COLUMN CommonName FORMAT A15;
SQL> COLUMN Commonname FORMAT A15;
SQL> SELECT Commonname, ScientificName
  2  FROM Medicine
  3  WHERE Disage LIKE '%child%' AND QuantityOnhand <1000;
WHERE Disage LIKE '%child%' AND QuantityOnhand <1000
      *
ERROR at line 3:
ORA-00904: "DISAGE": invalid identifier


SQL> SELECT Commonname, ScientificName
  2  FROM Medicine
  3  WHERE Dosage LIKE '%child%' AND QuantityOnhand <1000;

COMMONNAME      SCIENTIFICNAME
--------------- --------------------
Atarax          Hydroxyzine

SQL> /* the medication name required is Atarax (Commonname)*/
SQL>
SQL> /* Q.no.2.  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 */
SQL> COLUMN 'First Name' FORMAT A12;
SQL> COLUMN 'Last Name' FORMAT A12;
SQL> COLUMN 'moSalary' FORMAT $99,999;
SQL> SELECT FirstName 'First Name', LastName 'Last Name', Salary 'moSalary'
  2  FROM Employee
  3  WHERE Salary BETWEEN 10000 AND 20000
  4  ORDER BY Salary;
SELECT FirstName 'First Name', LastName 'Last Name', Salary 'moSalary'
                 *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> COLUMN "First Name" FORMAT A12;
SQL> COLUMN "Last Name" FORMAT A12;
SQL> COLUMN
COLUMN   moSalary ON
FORMAT   $99,999
```

```
COLUMN   Last Name ON
FORMAT   A12

COLUMN   First Name ON
FORMAT   A12

COLUMN   ScientificName ON
FORMAT   A20

COLUMN   Commonname ON
FORMAT   A15

COLUMN   Orginal Cost ON
FORMAT   9999.99

COLUMN   Project Number ON
FORMAT   99

COLUMN   Quantity Available ON
FORMAT   9999

COLUMN   Original Cost ON
FORMAT   9999.99

COLUMN   Description ON
FORMAT   A12

COLUMN   Gender ON
FORMAT   A6

COLUMN   Title ON
FORMAT   A5

COLUMN   FirstName ON
FORMAT   A12

COLUMN   LastName ON
FORMAT   A12

COLUMN   DateTreated ON
FORMAT   A12

COLUMN   EmployeeID ON
FORMAT   A12

COLUMN   PatientID ON
FORMAT   A10

COLUMN   NAME_COL_PDB_CLOUD_IDENTITY ON
HEADING  'CLOUD IDENTITY'

COLUMN   NAME_COL_PDB_APP_ROOT_CON_ID ON
HEADING  'APP|ROOT|CONID' headsep '|'
FORMAT   9999

COLUMN   NAME_COL_PDBD_RESTRICTED ON
HEADING  'REST'
```

```
FORMAT   A4

COLUMN   NAME_COL_PDB_APP_ROOT_CLONE ON
HEADING  'APP|ROOT|CLONE' headsep '|'
FORMAT   A5
word_wrap

COLUMN   NAME_COL_PDBD_CON_ID ON
HEADING  'CONID'
FORMAT   9999

COLUMN   NAME_COL_PDBD_NAME ON
HEADING  'CON_NAME'
FORMAT   A28
word_wrap

COLUMN   NAME_COL_PDB_PROXY_PDB ON
HEADING  'PXY|PDB' headsep '|'
FORMAT   A3
word_wrap

COLUMN   NAME_COL_PDB_APP_SEED ON
HEADING  'APP|SEED' headsep '|'
FORMAT   A4
word_wrap

COLUMN   NAME_COL_PDB_APP_PDB ON
HEADING  'APP|PDB' headsep '|'
FORMAT   A3
word_wrap

COLUMN   NAME_COL_PDB_APP_ROOT ON
HEADING  'APP|ROOT' headsep '|'
FORMAT   A4
word_wrap

COLUMN   NAME_COL_PLUS_PDB_RESTRICTED ON
HEADING  'RESTRICTED'
FORMAT   A10

COLUMN   NAME_COL_PLUS_PDB_OPEN_MODE ON
HEADING  'OPEN MODE'
FORMAT   A10

COLUMN   NAME_COL_PLUS_PDB_NAME ON
HEADING  'CON_NAME'
FORMAT   A30
word_wrap

COLUMN   NAME_COL_PLUS_PDB_CON_ID ON
HEADING  'CON_ID'

COLUMN   NAME_COL_PLUS_PDB_CONTAINERID ON
HEADING  'CON_ID'
FORMAT   a30
word_wrap
```

```
COLUMN   NAME_COL_PLUS_PDB_CONTAINER ON
HEADING  'CON_NAME'
FORMAT   a30
word_wrap

COLUMN   NAME_COL_PLUS_SHOW_EDITION ON
HEADING  'EDITION'
FORMAT   a30
word_wrap

COLUMN   result_plus_xquery ON
HEADING  'Result Sequence'

COLUMN   other_plus_exp ON
FORMAT   a44

COLUMN   other_tag_plus_exp ON
FORMAT   a29

COLUMN   object_node_plus_exp ON
FORMAT   a8

COLUMN   plan_plus_exp ON
FORMAT   a60

COLUMN   parent_id_plus_exp ON
HEADING  'p'
FORMAT   990

COLUMN   id_plus_exp ON
HEADING  'i'
FORMAT   990

COLUMN   droptime_plus_show_recyc ON
HEADING  'DROP TIME'
FORMAT   a19

COLUMN   objtype_plus_show_recyc ON
HEADING  'OBJECT TYPE'
FORMAT   a12

COLUMN   objectname_plus_show_recyc ON
HEADING  'RECYCLEBIN NAME'
FORMAT   a30

COLUMN   origname_plus_show_recyc ON
HEADING  'ORIGINAL NAME'
FORMAT   a16

COLUMN   SID_COL_PLUS_SHOW_SPPARAM ON
HEADING  'SID'
FORMAT   a8
word_wrap

COLUMN   VALUE_COL_PLUS_SHOW_SPPARAM ON
HEADING  'VALUE'
FORMAT   a28
```

```
word_wrap

COLUMN   NAME_COL_PLUS_SHOW_SPPARAM ON
HEADING  'NAME'
FORMAT   a29
word_wrap

COLUMN   value_col_plus_show_param ON
HEADING  'VALUE'
FORMAT   a30

COLUMN   name_col_plus_show_param ON
HEADING  'NAME'
FORMAT   a36

COLUMN   units_col_plus_show_sga ON
FORMAT   a15

COLUMN   name_col_plus_show_sga ON
FORMAT   a24

COLUMN   ERROR ON
FORMAT   A65
word_wrap

COLUMN   LINE/COL ON
FORMAT   A8

COLUMN   ROWLABEL ON
FORMAT   A15
SQL> COLUMN "First Name" FORMAT A12;
SQL> COLUMN "Last Name" FORMAT A12;
SQL> COLUMN "moSalary" FORMAT $99,999;
SQL> SELECT FirstName "First Name", LastName "Last Name", Salary "moSalary"
  2  FROM Employee
  3  WHERE Salary BETWEEN 10000 AND 20000
  4  ORDER BY Salary;

First Name   Last Name    moSalary
------------ ------------ --------
Maxwell      Eakin         $15,000
Robert       Klepper       $15,055
Douglas      Bock          $16,250
Elizabeth    Sumner        $16,500
Eugene       Webber        $17,500
Beverly      Boudreaux     $17,520
Robert       Schultheis    $17,525
Bijoy        Bordoloi      $17,850

8 rows selected.

SQL> /* it displays 8 employess whose salary is between 10000 and 20000*/
SQL>
SQL> /*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 âÄì PED105 inclusive.
```

```
Use the IN command. The room number, bedtype, and availability should be
shown in the report. Use meaningful column titles */
SQL> /* formating them first*/
SQL> COLUMN "Bed Number" FORMAT A12;
SQL> COLUMN "Bed Type" FORMAT A10;
SQL> COLUMN "Bed Availability" FORMAT A18;
SQL> SELECT BedNumber "Bed Number", BedType "Bed Type", BedAvailability "Bed
Availability"
  2  FROM Bed
  3   WHERE RoomNumber IN ('PED101','PED102','PED103','PED104','PED105');
SELECT BedNumber "Bed Number", BedType "Bed Type", BedAvailability "Bed
Availability"
                                                    *
ERROR at line 1:
ORA-00904: "BEDAVAILABILITY": invalid identifier


SQL> SELECT BedNumber "Bed Number", BedType " Bed Type", Availability "Bed
Availability"
  2  FROM Bed
  3  WHERE RoomNumber IN ('PED101','PED102','PED103','PED104','PED105');

Bed Number  B Bed Availability
---------- -- ------------------
########## P1 N
########## P1 Y
########## P1 N
########## P2 Y
########## P2 Y

SQL> SELECT BedNumber "Bed Number", BedType "Bed Type", Availability "Bed
Availability"
  2  FROM Bed
  3  WHERE RoomNumber IN ('PED101', 'PED102', 'PED103', 'PED104', 'PED105');

Bed Number Bed Type   Bed Availability
---------- ---------- ------------------
########## P1         N
########## P1         Y
########## P1         N
########## P2         Y
########## P2         Y

SQL> /* All information pediatric room 101-105 are displayed.*/
SQL>
SQL> /* 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 */
SQL> SELECT Name "First Name", BirthDate
  2  FROM Dependent
  3  WHERE RelationshipToEmployee IN ('SON', 'DAUGHTER');

First Name   BIRTHDATE
------------ ---------
Jo Ellen     05-APR-16
Andrew       25-OCT-18
```

```
Jeffery      01-JAN-08
Deanna       31-DEC-09
Rachael      04-OCT-15
Michelle     17-MAR-04
Anita        06-JUL-14
Monica       30-DEC-16
Rita         11-MAY-18

9 rows selected.

SQL> /* it displays 9 FirstNames AND Birthdays of sons and daughters of
employees*/
SQL>
SQL>
SQL> /* 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 âÄòoâÄô. List each
employee's first and last name. Use meaningful column titles */
SQL> SELECT FirstName "First Name" , LastName "Last Name
  2
SQL> SELECT FirstName "First Name", LastName "Last Name"
  2  FROM Employee
  3  WHERE LastName LIKE '%o%' AND LastName NOT LIKE '_o'%;
WHERE LastName LIKE '%o%' AND LastName NOT LIKE '_o'%
                                                    *
ERROR at line 3:
ORA-00911: invalid character


SQL> SELECT FirstName "First Name", LastName "Last Name"
  2  FROM Employee
  3  WHERE LastName LIKE '%o%' AND LastName NOT LIKE '_o%';

First Name   Last Name
------------ ------------
Lester       Simmons
Billy        Thornton
William      Clinton
William      Barlow
Toni         Quattromani
Mary Ellen   Brockwell
Leslie       Simmons

7 rows selected.

SQL> /* there are 7 employees who have o in their last name but not in the
2nd position*/
SQL>
SQL> /*Q.no.6.   Execute a query that lists all employee table rows that
contain a null value in the salary column. List each employeeâÄôs last name
and supervisor identifying number. Use meaningful column titles. Limit
column width so a line fits on a single row */
SQL> SELECT LastName "Last Name", SupervisorID "Supervisor Identifying
Number"
  2  FROM Employee
  3  WHERE Salary IS NULL;
```

```
Last Name    Super
------------ -----
Thornton     33355
Clinton      33355

SQL> /* It displays 2 employees whose salaries are null*
SQL> /
SQL> /* 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 patientâÄôs
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 */
SQL> /* my FirstName is Clara*/
SQL> COLUMN "First Name
string beginning ""First Nam..." missing terminating quote (").
SQL> COLUMN "First Name" FORMAT A12;
SQL> COLUMN "Last Name" FORMAT A12;
SQL> SELECT FirstName "First Name", LastName "Last Name"
  2  FROM Patient
  3  WHERE FirstName LIKE 'C%';

First Name   Last Name
------------ ------------
Charlie      Chang
Clyde        Crawford

SQL> /* i want to search for employees whose first name begin with the first
letter of my last name which is Chelibei*/
SQL> SELECT FirstName "First Name", LastName "Last Name"
  2  FROM Patient
  3  WHERE FirstName LIKE 'C%';

First Name   Last Name
------------ ------------
Charlie      Chang
Clyde        Crawford

SQL>
SQL> /*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 */
SQL> COLUMN "Annual" FORMAT $999,999.99;
SQL> COLUMN "Monthly" FORMAT $999,999.99;
SQL> COLUMN "Weekly" FORMAT $999,999.99;
SQL> SELECT LastName "Last Name", Salary*12 "Annual", Salary "Monthly",
Salary/4 "Weekly"
  2  FROM Employee
  3  WHERE Salary/4 < 1000
```

```
  4  ORDER BY LastName;

Last Name          Annual      Monthly       Weekly
------------ ------------ ------------ ------------
Simmons        $26,400.00    $2,200.00      $550.00
Young          $26,400.00    $2,200.00      $550.00

SQL>
SQL>
SQL> SPOOL OFF;
```

```
SQL> /*CLARA CHEROTICH CHELIBEI*/
SQL> /*LAB6-CHELIBEI*/
SQL> /*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 */
SQL> SELECT count(*) "Number of Nurses"
  2  FROM Employee
  3   WHERE Title in ('R.N', 'L.P.N.');

Number of Nurses
----------------
               2

SQL> /* There are 2 Nurses */
SQL>
SQL> /* 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 */
SQL> COLUMN "Average Equipment Cost" FORMAT $99,999.99;
SQL> COLUMN "Total Equipment Cost" FORMAT $99,999.99;
SQL> SELECT AVG(OriginalCost) "Average Equipment Cost", SUM(OriginalCost)
  2  "Total Equipment Co
  3  FROM Equipment;
ERROR:
ORA-01740: missing double quote in identifier


SQL> COLUMN "Average Equipment Cost" FORMAT $99,999.99;
SQL> COLUMN "Total Equipment Cost" FORMAT $99,999.99;
SQL> SELECT AVG(OriginalCost) "Average Equipment Cost", SUM(OriginalCost)
  2  "Total Equipment Cost"
  3  FROM Equipment;

Average Equipment Cost Total Equipment Cost
---------------------- --------------------
             $2,019.85           $18,178.67

SQL> /* The Average Equipment cost is $2,019.85 and      Total Equipment
Cost is $ 18,178.67 */
SQL>
SQL> \/* 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 */
SP2-0734: unknown command beginning "\/* Q.no.3..." - rest of line ignored.
SQL>
SQL> /* 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 */
SQL> SELECT MIN(BirthDate)
  2  FROM Dependent;
```

```
MIN(BIRTH
---------
05-MAY-76

SQL> /* This displays the oldest depenedent born on 5-MAY-76 */
SQL>
SQL> /* 4.  Write a query to provide the Executive Director with the total
hours worked per project. Use the ProjectNumber and HoursWorked columns from
the ProjectAssiginment 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. */
SQL> COLUMN "Total Hours" FORMAT 999.99;
SQL> SELECT ProjectNumber "Project Number" ,SUM(HoursWorked) "Total Hours"
  2  FROM ProjectAssignment
  3  GROUP BY ProjectNumber
  4  ORDER BY ProjectNumber;

Project Number Total Hours
-------------- -----------
             1. 14.20
             2. 10.60
             3. 52.80
             4. 69.10
             5. 86.10
             6
             7       27.00
             8       47.10

8 rows selected.

SQL> /* I displays a table that shows Project Number and Total Hours */
SQL>
SQL> /* 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 */
SQL> SELECT Title "Title", COUNT(Title) "NON M.D Employees"
  6. FROM Employee
  7. WHERE Title NOT LIKE '%M.D%'
  8. GROUP BY Title;

Title                                                NON M.D Employees
---------------------------------------------------- -----------------
Hospital Chief                                                       1
Pharmacist                                                           1
Records Clerk                                                        1
Rad. Tech.                                                           1
R.N.                                                                 2
V.P. Admin                                                           1
Building Custodian                                                   2
L.P.N.                                                               2

8 rows selected.

SQL> /* The where clause eliminated anything withy Title M.D */
```

```
SQL> /* The whyere clause eliminated any emloyee with    Title M.D*/
SQL> /*There are 11 Non M.D employees */
SQL>
SQL> /* 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 */
SQL> SELECT PatientID "Patient", SUM(ChargeAmount) "Total Charges"
  9. FROM Treatment
  10.GROUP BY PatientID
  11.HAVING SUM(ChargeAmount)< 350
  12.ORDER BY SUM(ChargeAmount);

Patien Total Charges
------ -------------
100001          15.4
555005            30
100024         55.95
421224         60.55
100002            65
100051            75
100028            75
222002            75
333115            95
333110           110
421227           115

Patien Total Charges
------ -------------
100029           125
100026           130
421223           145
100025           150
100423           155
333113           165
100506           170
666120           185
333111        185.55
333114        190.55
100305           195

Patien Total Charges
------ -------------
421226        210.55
421225           215
100030        215.55
666118           225
100301           230
100503           240
100502        245.55
100425           250
222006           258
100505           300
333116        310.55
```

```
Patien Total Charges
------ -------------
421228           325
421222           325
100302        325.55
100501        331.55

37 rows selected.

SQL> /* It displays a table of patients and their total charge*/
SQL>
SQL> /*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 */
SQL> SELECT PatientID "Patient", AVG(ChargeAmount) "average charges"
  13.FROM Treatment
  14.WHERE ChargeAmount>=500
  15.GROUP BY PatientID
  16.HAVING AVG(ChargeAmount)>500
  17.ORDER BY SUM(ChargeAmount);

Patien average charges
------ ---------------
100003             555
100031             600
421221            1480
100306            6200
100500            6500
666121            6500
222001            7800
100424            8500

8 rows selected.

SQL> /* It displays a table patients with their Average Charge */
SQL>
SQL>
SQL> SPOOL OFF;
```

```
SQL> /*CLARA CHEROTICH CHELIBEI */
SQL> /*LAB7-CHELIBEI*/
SQL>
SQL> /*Q.n.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'. */
SQL> COLUMN
COLUMN   NAME_COL_PDB_CLOUD_IDENTITY ON
HEADING  'CLOUD IDENTITY'

COLUMN   NAME_COL_PDB_APP_ROOT_CON_ID ON
HEADING  'APP|ROOT|CONID' headsep '|'
FORMAT   9999

COLUMN   NAME_COL_PDBD_RESTRICTED ON
HEADING  'REST'
FORMAT   A4

COLUMN   NAME_COL_PDB_APP_ROOT_CLONE ON
HEADING  'APP|ROOT|CLONE' headsep '|'
FORMAT   A5
word_wrap

COLUMN   NAME_COL_PDBD_CON_ID ON
HEADING  'CONID'
FORMAT   9999

COLUMN   NAME_COL_PDBD_NAME ON
HEADING  'CON_NAME'
FORMAT   A28
word_wrap

COLUMN   NAME_COL_PDB_PROXY_PDB ON
HEADING  'PXY|PDB' headsep '|'
FORMAT   A3
word_wrap

COLUMN   NAME_COL_PDB_APP_SEED ON
HEADING  'APP|SEED' headsep '|'
FORMAT   A4
word_wrap

COLUMN   NAME_COL_PDB_APP_PDB ON
HEADING  'APP|PDB' headsep '|'
FORMAT   A3
word_wrap

COLUMN   NAME_COL_PDB_APP_ROOT ON
HEADING  'APP|ROOT' headsep '|'
FORMAT   A4
word_wrap

COLUMN   NAME_COL_PLUS_PDB_RESTRICTED ON
HEADING  'RESTRICTED'
```

```
FORMAT   A10

COLUMN   NAME_COL_PLUS_PDB_OPEN_MODE ON
HEADING  'OPEN MODE'
FORMAT   A10

COLUMN   NAME_COL_PLUS_PDB_NAME ON
HEADING  'CON_NAME'
FORMAT   A30
word_wrap

COLUMN   NAME_COL_PLUS_PDB_CON_ID ON
HEADING  'CON_ID'

COLUMN   NAME_COL_PLUS_PDB_CONTAINERID ON
HEADING  'CON_ID'
FORMAT   a30
word_wrap

COLUMN   NAME_COL_PLUS_PDB_CONTAINER ON
HEADING  'CON_NAME'
FORMAT   a30
word_wrap

COLUMN   NAME_COL_PLUS_SHOW_EDITION ON
HEADING  'EDITION'
FORMAT   a30
word_wrap

COLUMN   result_plus_xquery ON
HEADING  'Result Sequence'

COLUMN   other_plus_exp ON
FORMAT   a44

COLUMN   other_tag_plus_exp ON
FORMAT   a29

COLUMN   object_node_plus_exp ON
FORMAT   a8

COLUMN   plan_plus_exp ON
FORMAT   a60

COLUMN   parent_id_plus_exp ON
HEADING  'p'
FORMAT   990

COLUMN   id_plus_exp ON
HEADING  'i'
FORMAT   990

COLUMN   droptime_plus_show_recyc ON
HEADING  'DROP TIME'
FORMAT   a19

COLUMN   objtype_plus_show_recyc ON
```

```
HEADING  'OBJECT TYPE'
FORMAT   a12

COLUMN   objectname_plus_show_recyc ON
HEADING  'RECYCLEBIN NAME'
FORMAT   a30

COLUMN   origname_plus_show_recyc ON
HEADING  'ORIGINAL NAME'
FORMAT   a16

COLUMN   SID_COL_PLUS_SHOW_SPPARAM ON
HEADING  'SID'
FORMAT   a8
word_wrap

COLUMN   VALUE_COL_PLUS_SHOW_SPPARAM ON
HEADING  'VALUE'
FORMAT   a28
word_wrap

COLUMN   NAME_COL_PLUS_SHOW_SPPARAM ON
HEADING  'NAME'
FORMAT   a29
word_wrap

COLUMN   value_col_plus_show_param ON
HEADING  'VALUE'
FORMAT   a30

COLUMN   name_col_plus_show_param ON
HEADING  'NAME'
FORMAT   a36

COLUMN   units_col_plus_show_sga ON
FORMAT   a15

COLUMN   name_col_plus_show_sga ON
FORMAT   a24

COLUMN   ERROR ON
FORMAT   A65
word_wrap

COLUMN   LINE/COL ON
FORMAT   A8

COLUMN   ROWLABEL ON
FORMAT   A15
SQL> COLUMN "Employee Name" FORMAT A24;
SQL> COLUMN "Specialty Name" FORMAT A27;
SQL> COLUMN "Date Recived" FORMAT A12;
SQL> SELECT LastName||', '||FirstName "Employee Name" , s.Title
  2
SQL> SELECT LastName||', '||FirstName "Employee Name" , s.Title "Specialty
Name", DateReceived "Date Recived"
```

```
  2  FROM Employee e JOIN EmployeeSpecialty es ON
(e.EmployeeID=es.EmployeeID) JOIN Specialty s ON
(es.SpecialtyID=s.SpecialtyID);

Employee Name            Specialty Name              Date Recived
------------------------ --------------------------- ------------
Simmons, Lester          Registered Nurse            04-FEB-12
Eakin, Maxwell           General Practitioner        04-DEC-16
Eakin, Maxwell           Surgeon-General             04-DEC-19
Bock, Douglas            General Practitioner        12-FEB-07
Webber, Eugene           Radiologist                 04-MAY-12
Bordoloi, Bijoy          Radiologist                 11-AUG-07
Smith, Alyssa            Radiology Technologist      04-DEC-08
Sumner, Elizabeth        General Practitioner        05-DEC-14
Sumner, Elizabeth        Surgeon-General             15-DEC-14
Becker, Robert           Surgeon-Thoracic            02-NOV-00
Jones, Quincey           Surgeon-General             15-DEC-09

Employee Name            Specialty Name              Date Recived
------------------------ --------------------------- ------------
Barlow, William          Neurosurgeon                12-MAY-15
Smith, Susan             Surgeon-General             22-AUG-16
Klepper, Robert          Oncologist                  04-JAN-03
Zumwalt, Mary            Registered Nurse            08-MAR-08
Quattromani, Toni        Cardiologist                25-MAY-12
Becker, Roberta          Surgeon-Abdominal Cavity    04-DEC-02
Brockwell, Mary Ellen    Nurse-Practitioner          06-DEC-15
Simmons, Leslie          Licensed Practicing Nurse   22-MAR-18
Young, Yvonne            Licensed Practicing Nurse   15-DEC-15
Schultheis, Robert       General Practitioner        10-DEC-99

21 rows selected.

SQL> /* 21 Employees has received Specialties */
SQL>
SQL> /*Q.n.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.*/
SQL> COLUMN "CommonName" FORMAT A10;
SQL> SELECT LastName||', '||FirstName "Patient Name", CommonName
"CommonName"
  2  FROM patient p JOIN Prescription pr ON (p.PatientID=pr.PatientID) JOIN
Medicine m ON (pr.MedicineCode=m.MedicineCode)
  3  WHERE Commonname='Valium';

Patient Name                                  CommonName
--------------------------------------------- ----------
Youngman, Yvonne                              Valium
Zebulon, Zeb                                  Valium
Ridgeway, Ricardo                             Valium

SQL> /* Patient Name who are Prescribed Valium */
SQL>
SQL> /*Q.n.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 */
SQL> COLUMN "Hours Worked" FORMAT 99.9;
SQL> SELECT LastName||', '||FirstName "Employee Name", ProjectTitle "Project
Title", HoursWorked "Hours Worked"
  2  FROM Employee e JOIN ProjectAssignment pa ON
(e.EmployeeID=pa.EmployeeID) JOIN Project p ON
(pa.ProjectNumber=p.ProjectNumber)
  3  WHERE HoursWorked >10
  4  ORDER BY LastName, p.ProjectNumber;

Employee Name            Project Title             Hours Worked
------------------------ ------------------------- ------------
Adams, Adam              Child Care Center                 23.0
Bock, Douglas            New MRI Installation              10.2
Bordoloi, Bijoy          Remodel ER Suite                  10.3
Brockwell, Mary Ellen    New Pediatric Monitors            14.8
Eakin, Maxwell           New Surgical Suite                14.2
Eakin, Maxwell           Remodel Surgical Suite            10.6
Klepper, Robert          New MRI Installation              11.8
Klepper, Robert          Remodel ER Suite                  19.2
Simmons, Lester          Personnel Records Update          35.4
Simmons, Lester          New Pediatric Monitors            12.2
Simmons, Lester          Child Care Center                 24.1

Employee Name            Project Title             Hours Worked
------------------------ ------------------------- ------------
Smith, Alyssa            New MRI Installation              30.8
Smith, Susan             Remodel ER Suite                  34.5
Thornton, Billy          Personnel Records Update          41.2

14 rows selected.

SQL> /*Displays  table with employee name, their project title and hours
worked*/
SQL>
SQL> /* Q.n.4.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
*/
SQL> COLUMN "Employee" FORMAT A10;
SQL> COLUMN "Emp Gender" FORMAT A10;
SQL> COLUMN "Dependent" FORMAT A10;
SQL> COLUMN "Dep Gender" FORMAT A10;
SQL> COLUMN "Relationship" FORMAT A12;
SQL> SELECT LastName "Employee", employee.Gender "Emp Gender", ependent.Name
"Dependent", dependent.Gender "Dep Gender",
  2
```

```
SQL> SELECT LastName "Employee", employee.Gender "Emp Gender",
dependent.Name "Dependent", dependent.Gender "Dep
Gender",RelationshipToEmployee "Relationship"
  2  FROM employee JOIN dependent ON
(employee.EmployeeID=dependent.EmployeeID)
  3   WHERE employee.Gender=dependent.Gender
  4   ORDER BY LastName;

Employee   Emp Gender Dependent  Dep Gender Relationship
---------- ---------- ---------- ---------- ------------
Bock       M          Jeffery    M          SON
Simmons    M          Andrew     M          SON

SQL> /* Displays  table with infromation of employee, Dependent their
relation and Gender*/
SQL>
SQL> Q.n.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
tale 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 */
SP2-0734: unknown command beginning "Q.n.5.The ..." - rest of line ignored.
SQL> /*Q.n.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
tale 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 */
SQL>  COLUMN "Patient Name" FORMAT A20;
SQL> SELECT p.FirstName||' '||p.LastName "Patient Name", bedNumber "Bed
Number",  roomNumber "Room Number"
  2  FROM Patient p JOIN Bed b
  3  ON (p.bedNo = b.bedNumber)JOIN Prescription pr
  4  ON (pr.patientID = p.patientID) JOIN Employee e
  5  ON (pr.employeeID = e.employeeID) JOIN Department d
  6  ON (e.departmentNumber = d.departmentNumber)
  7  WHERE departmentName = 'Pediatrics-Gynecology';

Patient Name         Bed Number Room N
-------------------- ---------- ------
Freddy Fender                70 RE0001
Zeb Zebulon                  52 ER0001
Arthur Ashcroft              53 ER0001

SQL> /*  Name and bed number and room no. of the patient in Pediatric */
SQL>
SQL> /*Q.n.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 */
SQL> SELECT LastName||', '||FirstName "Employee Name"
```

```
  2  FROM Employee e LEFT OUTER JOIN ProjectAssignment pa ON
(e.EmployeeID=pa.EmployeeID) LEFT OUTER JOIN Project p ON
(pa.ProjectNumber=p.ProjectNumber)
  3  WHERE pa.Projectnumber IS NULL
  4  ORDER BY e.LastName, e.FirstName;

Employee Name
------------------------
Barlow, William
Becker, Robert
Becker, Roberta
Boudreaux, Betty
Boudreaux, Beverly
Clinton, William
Jones, Quincey
Schultheis, Robert
Simmons, Leslie
Sumner, Elizabeth
Webber, Eugene

Employee Name
------------------------
Young, Yvonne
Zumwalt, Mary

13 rows selected.

SQL> /* There are 13 employees  who are not assigned to any project
currently */
SQL>
SQL> /*Q.n.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.
SQL>
SQL> /;Q.n.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.
SQL>
SQL> SELECT RoomNumber "Room Number", FirstName||' '||LastName "Patient
Name"
SQL>  SELECT RoomNumber "Room Number", FirstName||' '||LastName "Patient
Name"
SQL> FROM Bed b LEFT OUTER JOIN Patient p ON (p.Bedno=b.BedNumber);
SQL> /* Q.n.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.*/
SQL>  SELECT RoomNumber "Room Number", FirstName||' '||LastName "Patient
Name"
```

```
  2  FROM Bed b LEFT OUTER JOIN Patient p ON (p.Bedno=b.BedNumber);

Room N Patient Name
------ --------------------
MSS001 Barbara Benton
MSS002
MSS003
MSS004 Rue Chen
MSS005
MSS006 David Davis
MSS010 Earnest Earnhardt
MSS010
MSS011
MSS011
MSS012 Frank Franken

Room N Patient Name
------ --------------------
MSS012
MSS013
MSS013 Gregory Grant
MSS014 Harold Harnett
MSS014 Ivy Iona
MSS015
MSS015 Juliet Juneau
MSN201
MSN202 Keith Kraut
MSN203 Mandy Monday
MSN204

Room N Patient Name
------ --------------------
MSN205 Linda Lima
MSN210
MSN210 Nancy Nunn
MSN211
MSN211 Opal Ophelia
MSN212
MSN212 Paul Pauley
MSN213 Quincy Quentin
MSN213
MSN214
MSN214 Ricardo Ridgeway

Room N Patient Name
------ --------------------
MSN215
MSN215 Samuel Santiago
ER0001 Yancey Young
ER0001
ER0001 Zeb Zebulon
ER0001 Arthur Ashcroft
ER0002
ER0002 Charlie Chang
ER0002 Darlene Davidson
ER0002
ER0002 Earlene Earnhardt
```

Room N	Patient Name
ER0001	Billy Boudreaux
RE0001	Freddy Fender
RE0023	
RE0023	Gina Gentry
RE0024	
RE0031	Hank Henderson
RE0032	Ilama Ilama
RE0032	James Jupiter
RE0032	Krakatoa Khan
RA0075	Teresa Tempest
RA0075	

Room N	Patient Name
RA0076	Ulysses Unicorn
RA0077	Victor Victory
RA0077	
RA0077	William Williams
RA0078	
RA0078	
SUR001	
SUR002	
SUR003	
SUR004	
SW3001	Mickey Mousseau

Room N	Patient Name
SW3002	
SW3003	Lillian Lakeside
SW3004	
SW3005	Oliver Overstreet
SW3005	
SW3006	Norman November
SW3006	Rudolph Pappa
SW3007	
SW3007	Renny Reinhardt
SW3008	Quentin Queen
SW3008	Sally Surrey

Room N	Patient Name
CC1001	Thomas Teal
CC1011	Vanna Vanquish
CC1021	
CC1031	Uley Uniform
CC1031	Yvonne Youngman
CC1051	Zina Zenna
CC1061	
PED101	Andrew Able
PED102	Albert Algebra
PED103	
PED104	Bradley Beaufort

```
Room N Patient Name
------ --------------------
PED105 Clyde Crawford
PED111
PED111
PED112 Danny Dunland
PED112 Ertha Ezzra
PED113
PED113 Filbert Funk
PED114
PED114 Gretchen Greathouse

97 rows selected.

SQL> /* formated table showing bed number and the patient name if occupied*/
SQL>
SQL> /* Q.n.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 */
SQL> COLUMN "Specialty Name" FORMAT A30;
SQL> SELECT s.Title "Specialty Name", Count(e.employeeID) "No. of Employees"
  2  FROM Employee e JOIN EmployeeSpecialty es ON
(e.EmployeeID=es.EmployeeID) JOIN Specialty s ON
(es.SpecialtyID=s.SpecialtyID)
  3  GROUP BY s.Title;

Specialty Name                 No. of Employees
------------------------------ ----------------
Cardiologist                                  1
General Practitioner                          4
Nurse-Practitioner                            1
Radiologist                                   2
Surgeon-Thoracic                              1
Surgeon-Abdominal Cavity                      1
Licensed Practicing Nurse                     2
Radiology Technologist                        1
Registered Nurse                              2
Neurosurgeon                                  1
Oncologist                                    1

Specialty Name                 No. of Employees
------------------------------ ----------------
Surgeon-General                               4

12 rows selected.

SQL> /*Displays table showing count of employee who have particular type of
specialty */
SQL>
SQL> /*Q.n.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 */
```

```
SQL> SELECT DepartmentName "Department Name", ProjectTitle "Project Title",
count(e.employeeID) "No. of Employees"
  2   FROM Employee e JOIN  Department d ON
(e.DepartmentNumber=d.DepartmentNumber) JOIN Project p ON
(d.DepartmentNumber=p.DepartmentNumber)
  3  GROUP BY p.ProjectTitle, d.DepartmentName;

Department Name           Project Title             No. of Employees
------------------------- ------------------------- ----------------
Admin/Labs                Personnel Records Update                 5
Emergency-Surgical        Remodel ER Suite                         5
Pediatrics-Gynecology     New Pediatric Monitors                   4
Pediatrics-Gynecology     Child Care Center                        4
Medical Surgical Ward 1   New Surgical Suite                       2
Medical Surgical Ward 1   Remodel Surgical Suite                   2
Radiology                 New MRI Installation                     3
Emergency-Surgical        Add Crash Cart Equipment                 5

8 rows selected.

SQL> /*Displays table showing no. of employee assigned to each project
within each department*/
SQL>
SQL> /*Q.n.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 */
SQL> COLUMN "Doctor Name" FORMAT A20;
SQL>  COLUMN "Total Service Charge" FORMAT $99,999.99;
SQL>  SELECT LastName||', '||FirstName "Doctor Name",
Count(t.PatientID)"Number of Patient", SUM(t.ChargeAmount) "Total Service
Charge"
  2  FROM Employee e JOIN Treatment t on (e.EmployeeID=t.EmployeeID)
  3   WHERE Title LIKE '%M.D.%'
  4  GROUP BY e.LastName, e.FirstName;

Doctor Name          Number of Patient Total Service Charge
-------------------- ----------------- --------------------
Becker, Robert                       3            $7,075.00
Becker, Roberta                      4           $15,455.00
Sumner, Elizabeth                    8            $7,495.40
Schultheis, Robert                  12            $1,295.55
Bock, Douglas                       14            $1,575.40
Eakin, Maxwell                      13            $1,668.00
Barlow, William                      1            $8,500.00
Webber, Eugene                       2              $650.00
Quattromani, Toni                    2              $195.00
Bordoloi, Bijoy                      8            $3,165.00
Klepper, Robert                     11            $1,110.00

11 rows selected.

SQL> /*Displays  table showing doctor's name, number of patient he treated
and service charege*/
```

```
SQL>
SQL> /* Q.n.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.
SQL>
SQL>
SQL> /*Q.n.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.*/
SQL> /*Create view  vwSalary*/
SQL> COLUMN "EmpID" FORMAT A8;
SQL> COLUMN "EmpLastName" FORMAT A15;
SQL> COLUMN "EmpFirstName" FORMAT A15;
SQL> COLUMN "EmpSalary" FORMAT $999,999,999.99;
SQL> CREATE VIEW vwSalary
  2  (EmpID, EmpLastName, EmpFirstName, EmpSalary) AS
  3  SELECT EmployeeID, LastName, FirstName, Salary
  4  FROM Employee;

View created.

SQL> /*access  vwSalary*/
SQL> SELECT *
  2  FROM vwSalary
  3  WHERE EmpSalary>=20000;

EMPID    EMPLASTNAME     EMPFIRSTNAME           EMPSALARY
-------- --------------- --------------- ----------------
67555    Simmons         Lester                $22,000.00
88101    Becker          Robert                $23,545.00
88303    Jones           Quincey               $30,550.00
88404    Barlow          William               $27,500.00
88505    Smith           Susan                 $32,500.00
66425    Quattromani     Toni                  $22,325.00
88202    Becker          Roberta               $23,000.00

7 rows selected.

SQL>
SQL> /*Q.n.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.*/
SQL>  /*Creating a view named vwDepartmentProjects*/
SQL> COLUMN "Project" FORMAT A25;
SQL> COLUMN "Department" FORMAT A25;
SQL> COLUMN "Location" FORMAT A15;
SQL> CREATE VIEW vwDepartmentProjects
  2  (DeptNo, Department, Project, Location) AS
  3   SELECT d.DepartmentNumber, d.DepartmentName, p.ProjectTitle,
p.Location
  4   FROM Department d JOIN Project p ON
(d.DepartmentNumber=p.DepartmentNumber)
  5  WHERE Location IN ('Maryville', 'Edwardsville');

View created.

SQL> /*accessing vwDepartmentProjects*/
SQL> SELECT *
  2  FROM  vwDepartmentProjects;

    DEPTNO DEPARTMENT                PROJECT                   LOCATION
---------- ------------------------- ------------------------- -------------
--
         2 Radiology                 New MRI Installation      Maryville
         3 Emergency-Surgical        Add Crash Cart Equipment  Edwardsville
         3 Emergency-Surgical        Remodel ER Suite          Maryville
         8 Admin/Labs                Personnel Records Update  Maryville

SQL>
SQL> /*Q.n.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.*/
SQL>  /*Creating a view named vwProjectHours*/
SQL>  CREATE VIEW vwProjectHours
  2   ("Project", "AverageHours")AS
  3   SELECT ProjectTitle, Avg(Hoursworked)
  4   FROM ProjectAssignment pa JOIN Project p ON
(pa.ProjectNumber=p.ProjectNumber)
  5  GROUP BY ProjectTitle
  6   HAVING Avg(Hoursworked)>=15;

View created.

SQL>  /*accessing vwProjectHours*/
SQL> SELECT *
  2  FROM vwProjectHours;
```

```
Project                   AverageHours
------------------------- ------------
Personnel Records Update          28.7
Remodel ER Suite                17.275
New MRI Installation              17.6
Child Care Center                 23.55

SQL>  CREATE VIEW vwProjectHours
  2  2  (Project, AverageHours)AS
  3  3  SELECT ProjectTitle, Avg(Hoursworked)
  4  4  FROM ProjectAssignment pa JOIN Project p ON
  5  (pa.ProjectNumber=p.ProjectNumber)
  6  5
  7
SQL> /* LET ME DROP THE EXISTING PROJECT */
SQL> CREATE VIEW vwProjectHours
  2  (Project, AverageHours)AS
  3  SELECT ProjectTitle, Avg(Hoursworked)
  4   FROM ProjectAssignment pa JOIN Project p ON
(pa.ProjectNumber=p.ProjectNumber)
  5   GROUP BY ProjectTitle;
CREATE VIEW vwProjectHours
            *
ERROR at line 1:
ORA-00955: name is already used by an existing object


SQL> DROP VIEW vwProjectHours;

View dropped.

SQL> CREATE VIEW vwProjectHours
  2  (Project, AverageHours)AS
  3  SELECT ProjectTitle, Avg(Hoursworked)
  4  FROM ProjectAssignment pa JOIN Project p ON
(pa.ProjectNumber=p.ProjectNumber)
  5  GROUP BY ProjectTitle;

View created.

SQL>  /*accessing vwProjectHours*/
SQL> SELECT Project
  2  FROM vwProjectHours
  3  WHERE AverageHours >=15;

PROJECT
-------------------------
Personnel Records Update
Remodel ER Suite
New MRI Installation
Child Care Center

SQL>
SQL> SPOOL OFF;
```

```
SQL> /*CLARA CHEROTICH CHELIBEI*/
SQL> /*LAB9-CHELIBEI*/
SQL> /*Q.n.Write a SQL*Plus program to produce a report that lists
dependents of employees.  Your report should look like the one shown in
Figure 1.  Your report needs to have the following characteristics: (3
points)
SQL> •     Display the values shown in both top title and bottom title lines
including date and page number.
SQL> •     Assign meaningful column names as shown.
SQL> •     Display detail line data from the dependent table of the company
database.
SQL> •     Order the detail lines by the EmployeeID column.
SQL>
SQL>
SQL> /Q.n.1.Write a SQL*Plus program to produce a report that lists
dependents of employees.  Your report should look like the one shown in
Figure 1.  Your report needs to have the following characteristics: •
     Display the values shown in both top title and bottom title lines
including date and page number.  •Assign meaningful column names as shown.•
     Display detail line data from the dependent table of the company
database.• Order the detail lines by the EmployeeID column.*/
SQL> --Program: Q.n.-1.sql
SQL> --Programmer:Clara Chelibei
SQL> --Description:Information about employee's Dependent
SQL>
SQL> TTITLE 'Dependent Information'
SQL> BTITLE SKIP 2 CENTER 'Not for external dissemination.'
SQL> SET LINESIZE 55
SQL> SET PAGESIZE 24
SQL> SET NWEPAGE 1
SP2-0158: unknown SET option "NWEPAGE"
SQL>
SQL> COLUMN "Emp ID" FORMAT A6;
SQL> COLUMN "Dependent" FORMAT A15;
SQL> COLUMN "Gender" FORMAT A6;
SQL> COLUMN "Date Birth" FORMAT A10;
SQL> COLUMN "Relationship" FORMAT A12;
SQL> SELECT EmployeeID "Emp ID", Name "Dependent", gender "Gender",
BirthDate "Date Birth", RelationshipToEmployee "Relationship"
  2  FROM Dependent
  3  ORDER BY EmployeeID;

Sat Jun 08                                    page    1
                Dependent Information

Emp ID Dependent       Gender Date Birth Relationship
------ --------------- ------ ---------- ------------
01885  Deanna          F      31-DEC-09  DAUGHTER
01885  Jeffery         M      01-JAN-08  SON
01885  Mary Ellen      F      05-MAY-76  SPOUSE
01885  Michelle        F      17-MAR-04  DAUGHTER
01885  Rachael         F      04-OCT-15  DAUGHTER
23100  Anita           F      06-JUL-14  DAUGHTER
23100  Mita            F      04-JUN-76  SPOUSE
23100  Monica          F      30-DEC-16  DAUGHTER
23100  Rita            F      11-MAY-18  DAUGHTER
33355  Allen           M      29-FEB-88  SPOUSE
```

```
67555  Andrew          M      25-OCT-18  SON
67555  Jo Ellen        F      05-APR-16  DAUGHTER
67555  Susan           F      03-MAY-95  SPOUSE



            Not for external dissemination.

13 rows selected.

SQL>
SQL> /*Q.n.2 Modify the program for question 1 to produce a control break
report like the one shown in Figure 2.  Your report needs to have the
following additional characteristics •  Display each EmployeeID column value
only once for each group of dependents belonging to each employee.• Use the
COUNT aggregate function to count the number of dependents and display this
count at the end of the report. */
SQL> --Program:Q.n.-2.sql
SQL> --Programmer:Clara Chelibei
SQL> --Description: Information about employee's Dependent
SQL>
SQL> TTITLE 'Dependent Information'
SQL> BTITLE SKIP 1 CENTER 'Not for external dissemination.'
SQL> SET LINESIZE 55
SQL> SET PAGESIZE 24
SQL> SET NWEPAGE 1
SP2-0158: unknown SET option "NWEPAGE"
SQL>
SQL> COLUMN "Emp ID" FORMAT A6;
SQL> COLUMN "Dependent" FORMAT A15;
SQL> COLUMN "Gender" FORMAT A6;
SQL> COLUMN "Date Birth" FORMAT A10;
SQL> COLUMN "Relationship" FORMAT A12;
SQL>
SQL> CLEAR BREAK
breaks cleared
SQL> BREAK ON "Emp ID" SKIP 2 ON REPORT
SQL> COMPUTE COUNT OF "Emp ID" ON REPORT
SQL>
SQL> SELECT EmployeeID "Emp ID", Name "Dependent", gender "Gender",
BirthDate "Date Birth", RelationshipToEmployee "Relationship"
  2  FROM Dependent
  3  ORDER BY EmployeeID;

Sat Jun 08                                    page    1
                Dependent Information

Emp ID Dependent       Gender Date Birth Relationship
------ --------------- ------ ---------- ------------
01885  Deanna          F      31-DEC-09  DAUGHTER
       Jeffery         M      01-JAN-08  SON
       Mary Ellen      F      05-MAY-76  SPOUSE
       Michelle        F      17-MAR-04  DAUGHTER
       Rachael         F      04-OCT-15  DAUGHTER
```

23100	Anita	F	06-JUL-14	DAUGHTER
	Mita	F	04-JUN-76	SPOUSE
	Monica	F	30-DEC-16	DAUGHTER
	Rita	F	11-MAY-18	DAUGHTER
33355	Allen	M	29-FEB-88	SPOUSE

Not for external dissemination.

Sat Jun 08 page 2

Dependent Information

Emp ID	Dependent	Gender	Date Birth	Relationship
67555	Andrew	M	25-OCT-18	SON
	Jo Ellen	F	05-APR-16	DAUGHTER
	Susan	F	03-MAY-95	SPOUSE

13

Not for external dissemination.

13 rows selected.

```
SQL>
SQL> /*Q.n.Write an SQL*Plus program that will use a view named vwEmpDep.
This view should include columns from the employee and dependent tables as
specified in Table 1.  The program must produce a master-detail report like
the one shown in Figure 3.  Figure 3 shows pages 1 and 4 from the report.
Pages 2 and 3 are intentionally deleted from the figure to reduce the size
of the figure.  Your report should display all pages of the report.  Your
report needs to have the following characteristics: •    List each
employee's name at the top of a new report page with a page number.• Break on
the EmployeeID value.• Display the number of dependents per employee as a
subtotal with the subtotal label shown in the figure.*/
SQL> --Program:Q.n.-3.sql
SQL> --Programmer:Clara Chelibei
SQL> --Description: Information about employee's Dependent
SQL> /*Work for first table*/
SQL>  TTITLE CENTER 'Employee Name:' EmployeeNameVar -  RIGHT 'Page: '
FORMAT 99 sql.pno SKIP 2
SQL> BTITLE SKIP 1 CENTER 'Not for external dissemination.'
SQL> SET LINESIZE 65
```

```
SQL> SET PAGESIZE 15
SQL> SET NEWPAGE 1
SQL>
SQL> --create view
SQL> CREATE OR REPLACE VIEW vwEmpDep ( Employee, "Emp ID",
DependentName,gender,Relationship) AS
  2  SELECT e.Lastname||', '||e.Firstname, d.EmployeeID, d.Name,d.gender,
d.RelationshipToEmployee
  3   FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
  4  ORDER BY -DependentName;
ORDER BY -DependentName
         *
ERROR at line 4:
ORA-00904: "DEPENDENTNAME": invalid identifier


SQL> COLUMN Employee NEW_VALUE EmployeeNameVar NOPRINT
SQL> COLUMN "Emp ID" FORMAT A20
SQL> COLUMN DependentName FORMAT A15
SQL> COLUMN gender FORMAT A6
SQL> COLUMN Relationship FORMAT A12
SQL>
SQL> BREAK ON Employee ON "Emp ID" PAGE;
SQL> COMPUTE COUNT LABEL "Number of Dependents" OF Relationship ON "Emp ID"
SQL>
SQL> SELECT Employee, "Emp ID", DependentName, gender, Relationship
  2  FROM vwEmpDep
  3  ORDER BY "Emp ID" , relationship desc;
FROM vwEmpDep
     *
ERROR at line 2:
ORA-00942: table or view does not exist


SQL> --Program:Q.n.-3.sql
SQL> --Programmer:Clara Chelibei
SQL> --Description:Information about employee's Dependent
SQL>
SQL> /*it works for the firt table*/
SQL> TTITLE CENTER 'Employee Name:' EmployeeNameVar-RIGHT 'Page:' 'FORMAT 99
sql.pno SKIP 2
string beginning "'FORMAT 99..." missing terminating quote (').
SQL> TTITLE CENTER 'Employee Name:' EmployeeNameVar-RIGHT 'Page:' FORMAT 99
sql.pno SKIP 2
SQL> BTITLE SKIP 1 CENTER 'Not for external dissemination.'
SQL> SET LINESIZE 65
SQL> SET PAGESIZE 15
SQL> SET NEWPAGE 1
SQL>
SQL> --create view
SQL> CREATE OR REPLACE VIEW vwEmpDep(Employee,"Emp
ID",DependentName,gender,Relationship) AS
  2  SELECT e.LastName||','||e.
      FirstName,d.EmployeeID,d.Name,d.gender,d.RelationshipToEmloyee
  3  FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
  4   ORDER BY DependentName;
      ORDER BY DependentName
```

```
                          *
ERROR at line 4:
ORA-00904: "DEPENDENTNAME": invalid identifier


SQL> CREATE OR REPLACE VIEW vwEmpDep(Employee,"Emp
ID",DependentName,gender,Relationship) AS
  2  SELECT
e.LastName||','||e.FirstName,d.EmployeeID,d.Name,d.gender,d.RelationshipToEm
ployee
  3  FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
  4  ORDER BY Dependent.Name;
ORDER BY Dependent.Name
         *
ERROR at line 4:
ORA-00904: "DEPENDENT"."NAME": invalid identifier


SQL> CREATE OR REPLACE VIEW vwEmpDep(Employee,"Emp ID",
DependentName,gender,Relationship) AS
  2  SELECT
e.LastName||','||e.FirstName,d.EmployeeID,d.Name,d.gender,d.RelationshipToEm
ployee
  3  FROM Dependent d JOIN Employee e ON (e.employeeID=d.employeeID)
  4  ORDER BY d.Name;

View created.

SQL> COLUMN Employee NEW_VALUE EmployeeNameVar NOPRINT
SQL> COLUMN "Emp ID" FORMAT A20
SQL> COLUMN DependentName FORMAT A15
SQL>  COLUMN gender FORMAT A6
SQL> COLUMN Relationship FORMAT A12
SQL>
SQL> BREAK ON Employee ON "Emp ID" PAGE;
SQL>  COMPUTE COUNT LABEL "Number of Dependents" OF Relationship ON "Emp ID"
SQL>
SQL> SELECT Employee, "Emp ID", DependentName, gender, Relationship
  2  FROM vwEmpDep
  3  ORDER BY "Emp ID" , relationship desc;

             Employee Name:Bock,Douglas-RIGHTPage:  1

Emp ID               DEPENDENTNAME   GENDER RELATIONSHIP
-------------------- --------------- ------ ------------
01885                Mary Ellen      F      SPOUSE
                     Jeffery         M      SON
                     Rachael         F      DAUGHTER
                     Michelle        F      DAUGHTER
                     Deanna          F      DAUGHTER
********************                        ------------
Number of Dependents                                   5


                  Not for external dissemination.

             Employee Name:Bordoloi,Bijoy-RIGHTPage:  2
```

```
Emp ID               DEPENDENTNAME   GENDER RELATIONSHIP
-------------------- --------------- ------ ------------
23100                Mita            F      SPOUSE
                     Rita            F      DAUGHTER
                     Monica          F      DAUGHTER
                     Anita           F      DAUGHTER
********************                        ------------
Number of Dependents                                   4
```

Not for external dissemination.

Employee Name:Boudreaux,Beverly-RIGHTPage: 3

```
Emp ID               DEPENDENTNAME   GENDER RELATIONSHIP
-------------------- --------------- ------ ------------
33355                Allen           M      SPOUSE
********************                        ------------
Number of Dependents                                   1
```

Not for external dissemination.

Employee Name:Simmons,Lester-RIGHTPage: 4

```
Emp ID               DEPENDENTNAME   GENDER RELATIONSHIP
-------------------- --------------- ------ ------------
67555                Susan           F      SPOUSE
                     Andrew          M      SON
                     Jo Ellen        F      DAUGHTER
********************                        ------------
Number of Dependents                                   3
```

Not for external dissemination.

```
13 rows selected.

SQL>  SELECT Employee, "Emp ID", DependentName, gender, Relationship
  2  FROM vwEmpDep
  3  ORDER BY "Emp ID" , relationship;
```

Employee Name:Bock,Douglas-RIGHTPage: 1

```
Emp ID               DEPENDENTNAME   GENDER RELATIONSHIP
-------------------- --------------- ------ ------------
01885                Rachael         F      DAUGHTER
                     Michelle        F      DAUGHTER
                     Deanna          F      DAUGHTER
```

	Jeffery	M	SON
	Mary Ellen	F	SPOUSE
**********			----------
Number of Dependents			5

Not for external dissemination.

Employee Name:Bordoloi,Bijoy-RIGHTPage: 2

Emp ID	DEPENDENTNAME	GENDER	RELATIONSHIP
23100	Rita	F	DAUGHTER
	Monica	F	DAUGHTER
	Anita	F	DAUGHTER
	Mita	F	SPOUSE
**********			----------
Number of Dependents			4

Not for external dissemination.

Employee Name:Boudreaux,Beverly-RIGHTPage: 3

Emp ID	DEPENDENTNAME	GENDER	RELATIONSHIP
33355	Allen	M	SPOUSE
**********			----------
Number of Dependents			1

Not for external dissemination.

Employee Name:Simmons,Lester-RIGHTPage: 4

Emp ID	DEPENDENTNAME	GENDER	RELATIONSHIP
67555	Jo Ellen	F	DAUGHTER
	Andrew	M	SON
	Susan	F	SPOUSE
**********			----------
Number of Dependents			3

Not for external dissemination.

13 rows selected.

```
SQL>  SELECT Employee, "Emp ID", DependentName, gender, Relationship
  2  FROM vwEmpDep
```

```
  3  ORDER BY "Emp ID" , relationship ASC;
```

Employee Name:Bock,Douglas-RIGHTPage: 1

Emp ID	DEPENDENTNAME	GENDER	RELATIONSHIP
01885	Rachael	F	DAUGHTER
	Michelle	F	DAUGHTER
	Deanna	F	DAUGHTER
	Jeffery	M	SON
	Mary Ellen	F	SPOUSE
**********			----------
Number of Dependents			5

Not for external dissemination.

Employee Name:Bordoloi,Bijoy-RIGHTPage: 2

Emp ID	DEPENDENTNAME	GENDER	RELATIONSHIP
23100	Rita	F	DAUGHTER
	Monica	F	DAUGHTER
	Anita	F	DAUGHTER
	Mita	F	SPOUSE
**********			----------
Number of Dependents			4

Not for external dissemination.

Employee Name:Boudreaux,Beverly-RIGHTPage: 3

Emp ID	DEPENDENTNAME	GENDER	RELATIONSHIP
33355	Allen	M	SPOUSE
**********			----------
Number of Dependents			1

Not for external dissemination.

Employee Name:Simmons,Lester-RIGHTPage: 4

Emp ID	DEPENDENTNAME	GENDER	RELATIONSHIP
67555	Jo Ellen	F	DAUGHTER
	Andrew	M	SON
	Susan	F	SPOUSE
**********			----------
Number of Dependents			3

```
13 rows selected.

SQL> /*Well professor i have tried all possible ways of arrangement for the
table to look exactly like yours but its not displaying exact ordereing.
Hopefully the ordering doesnt matter.*/
SQL>
SQL> SPOOL OFF;
```

```
SQL> /*CLARA CHEROTICH CHELIBEI*/
SQL> /*LAB10-CHELIBEI*/
SQL> /*Q.n.1.1.  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. */
SQL> COLUMN "Employee Name" FORMAT A22;
SQL> SELECT FirstName ||' '||  SUBSTR(MiddleName, 1,1) ||' '||
UPPER(LastName)  "Employee Name"
  2  FROM Employee
  3  WHERE DepartmentNumber=8
  4  ORDER BY LastName,FirstName;

Employee Name
----------------------
Adam A ADAMS
Beverly B BOUDREAUX
William W CLINTON
Lester L SIMMONS
Billy B THORNTON

SQL>
SQL> /*Q.n.2.2.  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. */
SQL> COLUMN "Dept. Name" FORMAT A26;
SQL> COLUMN "Length" FORMAT 999999;
SQL> SELECT DepartmentName "Dept. Name", LENGTH(DepartmentName) "Length"
  2  FROM Department;

Dept. Name                  Length
-------------------------- -------
Medical Surgical Ward 1         23
Radiology                        9
Emergency-Surgical              18
Oncology Ward                   13
Critical Care-Cardiology        24
Pediatrics-Gynecology           21
Pharmacy Department             19
Admin/Labs                      10
OutPatient Clinic               17

9 rows selected.

SQL> /* 9 rows of Department names are displayed.*/
SQL>
SQL> /*Q.n.3.3.  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. */
SQL> COLUMN "Department Information" FORMAT A32;
SQL> SELECT To_CHAR(DepartmentNumber, '999')||'  '|| DepartmentName
"Department Information"
  2  FROM Department;
```

```
Department Information
------------------------------
   1. Medical Surgical Ward 1
   2. Radiology
   3. Emergency-Surgical
   4. Oncology Ward
   5. Critical Care-Cardiology
   6. Pediatrics-Gynecology
   7. Pharmacy Department
   8. Admin/Labs
   9. OutPatient Clinic

10.rows selected.

SQL> /* 9 rows of department information are displayed.*/
SQL>
SQL> /*Q.n.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. */
SQL> COLUMN "Name" FORMAT A5;
SQL> COLUMN "SSN" FORMAT A5;
SQL> SELECT SUBSTR(LastName, 1,4) "Name", SUBSTR(SSN,6) "SSN"
  11.FROM Employee
  12.WHERE DepartmentNumber=8
  13.ORDER BY LastName;

Name  SSN
----- -----
Adam  3287
Boud  6222
Clin  0233
Simm  9642
Thor  6129

SQL> \
SP2-0042: unknown command "\" - rest of line ignored.
SQL>
SQL> /*Q.n.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. */
SQL> COLUMN "Name" FORMAT A5;
SQL> COLUMN "SSN" FORMAT A5;
SQL> SELECT SUBSTR(LastName, 1,4) "Name", SUBSTR(SSN,6) "SSN"
  14.FROM Employee
  15.WHERE DepartmentNumber=8
  16.ORDER BY LastName;

Name  SSN
----- -----
Adam  3287
Boud  6222
Clin  0233
Simm  9642
Thor  6129
```

```
SQL>
SQL> /*Q.n.
SQL> 5.    Write a query that displays all employee names as well as their
work phone.  Format their workphone to 999-999-9999.
SQL>
SQL>
SQL> /*Q.n.5.    Write a query that displays all employee names as well as
their work phone.  Format their workphone to 999-999-9999.*/
SQL> COLUMN "Employee Name" FORMAT A25;
SQL> COLUMN "Phone No." FORMAT A18;
SQL> SELECT LastName ||','|| FirstName "Employee Name",
SUBSTR(WorkPhone,1,3)||'-'||SUBSTR(WorkPhone,4,3)||'-
'||SUBSTR(WorkPhone,7,4) "Phone No."
  17.FROM Employee;

Employee Name             Phone No.
------------------------- ------------------
Simmons,Lester            100-555-9401
Boudreaux,Beverly         100-555-8287
Adams,Adam                100-555-8287
Thornton,Billy            100-555-8287
Clinton,William           100-555-8287
Eakin,Maxwell             100-555-9268
Bock,Douglas              100-555-9268
Webber,Eugene             100-555-9270
Bordoloi,Bijoy            100-555-9270
Smith,Alyssa              100-555-9267
Sumner,Elizabeth          100-555-9271

Employee Name             Phone No.
------------------------- ------------------
Becker,Robert             100-555-9284
Jones,Quincey             100-555-9284
Barlow,William            100-555-9284
Smith,Susan               100-555-9284
Klepper,Robert            100-555-9268
Zumwalt,Mary              100-555-9401
Quattromani,Toni          100-555-9280
Becker,Roberta            100-555-9284
Brockwell,Mary Ellen      100-555-9401
Simmons,Leslie            100-555-9401
Young,Yvonne              100-555-9401

Employee Name             Phone No.
------------------------- ------------------
Boudreaux,Betty           100-555-8287
Schultheis,Robert         100-555-9284

24 rows selected.

SQL> /*All the employees names are displayed with their work phone number*/
SQL>
SQL> /*Q.n.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. */
```

```
SQL> COLUMN "Emp Last Name" FORMAT A14;
SQL> COLUMN "EmployeeID" FORMAT A18;
SQL> SELECT LastName "Emp Last Name", DECODE(SupervisorID, NULL, 'Top
Supervisor' , EmployeeID) "EmployeeID"
  18.FROM Employee
  19.WHERE DepartmentNumber=8
  20.ORDER BY lastName;

Emp Last Name  EmployeeID
-------------- ------------------
Adams          33344
Boudreaux      33355
Clinton        33359
Simmons        Top Supervisor
Thornton       33358

SQL> /*If supervisorID is misspelled*/
SQL> COLUMN "Emp Last Name" FORMAT A14;
SQL> COLUMN "SupervisorID" FORMAT A18;
SQL> SELECT LastName "Emp Last Name", NVL(SupervisorID, 'Top Supervisor')
"SupervisorID"
  21.FROM Employee
  22.WHERE DepartmentNumber=8
  23.ORDER BY LastName;

Emp Last Name  SupervisorID
-------------- ------------------
Adams          33355
Boudreaux      67555
Clinton        33355
Simmons        Top Supervisor
Thornton       33355

SQL>
SQL> /*Q.n.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. */
SQL> COLUMN "Emp Last Name" FORMAT A14;
SQL> COLUMN "Project No." FORMAT 99;
SQL> COLUMN "Hours worked" FORMAT 99.9;
SQL> SELECT LastName "Emp Last Name", ProjectNumber "Project No.",
HoursWorked "Hours worked"
  24.FROM Employee e JOIN ProjectAssignment pa ON
     (e.employeeid=pa.employeeid)
  25.WHERE ABS(HoursWorked -20) <=5
  26.ORDER BY LastName;

Emp Last Name  Project No. Hours worked
-------------- ----------- ------------
Adams                    8         23.0
Klepper                  4         19.2
Simmons                  8         24.1
```

```
SQL>
SQL> /*Q.n.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. */
SQL> COLUMN "Emp Last Name" FORMAT A14;
SQL> COLUMN "Project No." FORMAT 99;
SQL> SELECT LastName "Emp Last Name", ProjectNumber "Project No.",
ROUND(HoursWorked,0)
  27.FROM Employee e JOIN ProjectAssignment pa ON
     (e.employeeid=pa.employeeid)
  28.WHERE ProjectNumber IN (3,8)
  29.ORDER BY ProjectNumber,LastName;

Emp Last Name  Project No. ROUND(HOURSWORKED,0)
-------------- ----------- --------------------
Bock                     3                   10
Klepper                  3                   12
Smith                    3                   31
Adams                    8                   23
Eakin                    8
Simmons                  8                   24

30.rows selected.

SQL> /*6 rows are selected*/
SQL>
SQL> /*Q.n.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. */
SQL> COLUMN "Gender" Format A6;
SQL> COLUMN "Dep Name" FORMAT A15;
SQL> SELECT Name "Dep Name", Gender "Gender", TO_CHAR(BirthDate, 'Month
DD,YYYY') "Date of Birth"
  31.FROM Dependent
  32.WHERE Gender='F';

Dep Name        Gender Date of Birth
--------------- ------ -------------------------------------------
Jo Ellen        F      April     05,2016
Susan           F      May       03,1995
Deanna          F      December  31,2009
Rachael         F      October   04,2015
Michelle        F      March     17,2004
Mary Ellen      F      May       05,1976
Mita            F      June      04,1976
Anita           F      July      06,2014
Monica          F      December  30,2016
Rita            F      May       11,2018

33. rows selected.
```

```
SQL> /*10 rows of females are displayed*/
SQL>
SQL> /*Q.n.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. */
SQL> SELECT Name "Dep Name", TO_CHAR(BirthDate, 'DD-MON-YYYY') "Date of
SELECT Name "Dep Name", TO_CHAR(BirthDate, 'DD-MON-YYYY') "Date of Birthday"
  34.FROM Dependent
  35.WHERE BirthDate >  TO_DATE('01-JAN-1980', 'DD-MON-YYYY');
ERROR:
ORA-01740: missing double quote in identifier


SQL> SELECT Name "De Name", TO_CHAR(BirthDate,'DD-MON-YYYY') "Date of Birth"
TO_CHAR(ADD_MONTHS (BirthDate,780), 'DD-MON-YYYY' "65th BirthDay"
  36.FROM Dependent
  37.WHERE BirthDate>TO_DATE('01-JAN-1980','DD-MON-YYYY');
SELECT Name "De Name", TO_CHAR(BirthDate,'DD-MON-YYYY') "Date of Birth"
TO_CHAR(ADD_MONTHS (BirthDate,780), 'DD-MON-YYYY' "65th BirthDay"
                                                                        *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> SELECT Name "Dep Name", TO_CHAR(BirthDate,'DD-MON-YYYY') "Date of
Birth" TO_CHAR(ADD_MONTHS(BirthDate,780),'DD-MON-YYYY' "65th BirthDay"
  38.FROM Dependent
  39.WHERE BirthDate>TO_DATE('01-JAN-1980','DD-MON-YYYY');
SELECT Name "Dep Name", TO_CHAR(BirthDate,'DD-MON-YYYY') "Date of Birth"
TO_CHAR(ADD_MONTHS(BirthDate,780),'DD-MON-YYYY' "65th BirthDay"
                                                                       *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> SELECT Name "Dep Name", TO_CHAR(BirthDate, 'DD-MON-YYYY') "Date of
Birth", TO_CHAR(ADD_MONTHS(BirthDate, 780), 'DD-MON-YYYY') "65th BirthDay"
  40.FROM Dependent
  41.WHERE BirthDate > TO_DATE('01-JAN-1980', 'DD-MON-YYYY');

Dep Name        Date of Birth        65th BirthDay
--------------- -------------------- --------------------
Jo Ellen        05-APR-2016          05-APR-2081
Andrew          25-OCT-2018          25-OCT-2083
Susan           03-MAY-1995          03-MAY-2060
Allen           29-FEB-1988          28-FEB-2053
Jeffery         01-JAN-2008          01-JAN-2073
Deanna          31-DEC-2009          31-DEC-2074
Rachael         04-OCT-2015          04-OCT-2080
Michelle        17-MAR-2004          17-MAR-2069
Anita           06-JUL-2014          06-JUL-2079
Monica          30-DEC-2016          30-DEC-2081
Rita            11-MAY-2018          11-MAY-2083

42. rows selected.
```

```
SQL> /* 11 rows of dependent names are selected.*/
SQL>
SQL> /*Q.n.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.*/
SQL> SELECT TO_CHAR(SYSDATE, 'DAY') "Current Day"
  43.FROM Dual;

Current Day
-----------------------------------
SATURDAY

SQL>
SQL> /*Q.n.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. */
SQL> SELECT Name "Dep Name", DECODE(Gender, 'M', 'Male', 'Female') "Gender"
  44.FROM Dependent
  45.WHERE RelationshipToEmployee='SPOUSE'
  46.ORDER BY Name;

Dep Name        Gender
--------------- ------
Allen           Male
Mary Ellen      Female
Mita            Female
Susan           Female

SQL>
SQL> /*Q.n.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. */
SQL> COLUMN "Salary" FORMAT $999,990.99; --tailing term is equal to zero
SP2-0246: Illegal FORMAT string "$999,990.99;"
SQL> COLUMN "Emp Last Name" FORMAT A15;
SQL> SELECT LastName "Emp Last Name", Title, NVL(Salary, '0.00') "Salary"
  47.FROM Employee
  48.WHERE Title LIKE '%Building Custodian%' OR Title LIKE '%L.P.N%'
  49.ORDER BY LastName;

Emp Last Name   TITLE
Salary
--------------- -------------------------------------------------- ---------
-
Clinton         Building Custodian
0
Simmons         L.P.N.
2200
Thornton        Building Custodian
0
Young           L.P.N.
2200
```

```
SQL> /*Alternaqtively as */
SQL> COLUMN "Emp Last Name" FORMAT A15;
SQL> COLUMN "Salary" FORMAT $99,999.99;
SQL> COLUMN "Title" FORMAT A20;
SQL> SELECT LastName "Emp Last Name", Title "Title", DECODE(TO_CHAR(Salary,
'$9,999'), NULL, ' $0.00', TO_CHAR(Salary,'$9,999.99')) "Salary"
  50.FROM Employee
  51.WHERE Title LIKE '%Building Custodian%' OR Title LIKE '%L.P.N%'
  52.ORDER BY LastName;

Emp Last Name   Title                Salary
--------------- -------------------- ----------
Clinton         Building Custodian    $0.00
Simmons         L.P.N.                $2,200.00
Thornton        Building Custodian    $0.00
Young           L.P.N.                $2,200.00

SQL>
SQL> /*Q.n.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.  */
SQL> COLUMN "Service ID"FORMAT A12;
SQL> COLUMN "Service Charge" FORMAT $999,999.99;
SQL> COLUMN "Treatment Charge" FORMAT $99,999.99;
SQL> COLUMN "Difference" FORMAT $999,999.99;
SQL>  SELECT s.ServiceID "Service ID", s.StandardCharge "Service Charge",
t.ChargeAmount "Treatment Charge", ABS(StandardCharge - ChargeAmount)
"Difference"
  53.FROM Service s JOIN Treatment t ON (s.serviceID=t.serviceID)
  54.WHERE ABS(s.StandardCharge - t.ChargeAmount)> 50;

Service ID   Service Charge Treatment Charge   Difference
------------ -------------- ---------------- ------------
12001             $6,200.00          $450.00    $5,750.00
12007            $10,000.00        $8,500.00    $1,500.00
12010             $3,500.00        $1,480.00    $2,020.00
99058               $155.00          $435.00      $280.00

SQL>
SQL> /*Q.n.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.*/
SQL> COLUMN "Service ID"FORMAT A12;
SQL> COLUMN "Service Charge" FORMAT $999,999.99;
SQL> COLUMN "Actual Charge" FORMAT $999,999.99;
SQL> COLUMN "Difference" FORMAT $999,999.99;
SQL> SELECT s.ServiceID "Service ID", s.StandardCharge "Service Charge",
t.ChargeAmount "Actual Charge", ABS(StandardCharge - ChargeAmount)
"Difference"
  55.FROM Service s JOIN Treatment t ON (s.serviceID=t.serviceID)
```

```
  56.WHERE ChargeAmount<StandardCharge
  57.ORDER BY ABS(StandardCharge - ChargeAmount) DESC;

Service ID   Service Charge Actual Charge   Difference
------------ -------------- ------------- ------------
12001             $6,200.00       $450.00    $5,750.00
12010             $3,500.00     $1,480.00    $2,020.00
12007            $10,000.00     $8,500.00    $1,500.00
99203                $95.00        $75.00       $20.00
99203                $95.00        $75.00       $20.00
99058               $155.00       $150.00        $5.00
90782                $75.00        $70.00        $5.00

58.rows selected.

SQL>
SQL> SPOOL OFF;
```