DEPARTMENT OF
DATA SCIENCE
2023-2024

Syllabus for B.Tech. II year II Semester
Computer Science and Engineering-Data Science
DATABASE MANAGEMENT SYSTEMS LAB

L	T	P/D	C	
0	0	3	1.5	Code: 9FC63

Course objective:
Design the optimal queries using structured and unstructured query languages like SQL and PL/SQL by making use of control structures, cursors, triggers and functions/procedures.

Course Outcomes:
At the end of this course, the student will be able to:

1 Create tables for a database and apply Queries using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.

2 Write Queries using Aggregate functions such as [COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING], Conversion functions and use string functions for a given application.

3 Explain and write programs using PL/SQL programs using exceptions, COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

4 Develop programs using WHILE LOOPS, FOR LOOPS, nested loops using BUILT–IN Exceptions and write Procedures.

5 Write Programs for stored functions invoke functions in SQL Statement and write Programs for packages specification.

6 Describe and write programs using features of CURSORs and its variables.

7 Develop Programs implementing Triggers.

Exercises:

1. Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
2. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.

   Example: - Select the roll number and name of the student who secured fourth rank in the class.
3. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping ofViews.
4. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char,to_date)
5. i)Creation of simple PL/SQL program which includes declaration section, executable Section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
   ii) Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQLblock.
6. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
7. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.
8. Programs development using creation of procedures, passing parameters IN and OUT Of PROCEDURES.
9. Program development using creation of stored functions, invoke functions in SQL Statement and write complex functions.
10. Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
11. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
12. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers.
13. Queries using SQL-INJECTION: AND/OR Attack, Comments Attack, String Concatenation Attack, UNION Injection Attack

TEXT BOOKS:

1) ORACLE PL/SQL by example. Benjamin Rosenzweig, Elena Silvestrova, Pearson Education 3Edition
2) ORACLE DATA BASE LOG PL/SQL Programming SCOTT URMAN, Tata Mc-Graw Hill.
3) SQL and PL/SQL for Oracle 10g, Black Book, Dr. P. S.Deshpande.

List 1

1. Create Sailors table with the following fields and describe the table (*sid:* integer, *sname:* string, *rating:* integer)

2. Add field (age: integer) to the Sailors table and describe the table

3. Modify field age to real in the Sailors table and describe the table

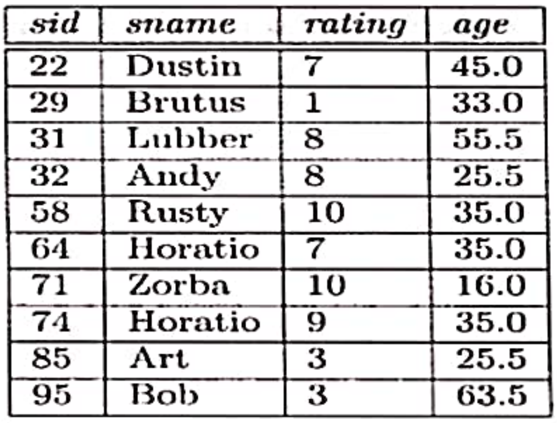
4. Delete age field from the Sailors table and describe the table

5. Add field (age: real) to the Sailors table and describe the table

6. Delete Sailors table

7. Create Sailors table with the following fields and describe the table (*sid:* integer, *sname:* string, *rating:* integer, *age:* real)

8. Insert the following records in to the Sailors table



sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

9. Retrieve all Records from the Sailors table

10. Retrieve all Records (unique) from the Sailors table

11. Retrieve the names and ages of all sailors from the Sailors table

12. Find all sailors with a rating above 7

13. Create Sailors1 table with the following fields and describe the table (sid: integer, sname: string, rating: integer, age: real)

14. Insert tuples of Sailors table in to the Sailors1 table

15. Retrieve all Records from the Sailors1 table

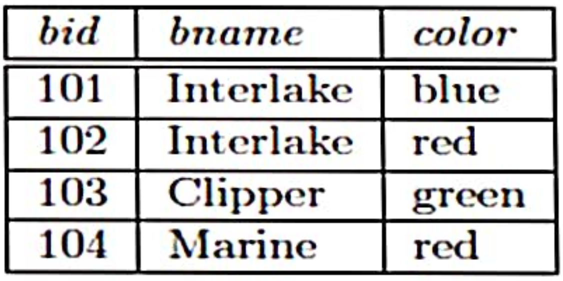
16. Delete sailor from the Sailors1 table whose sid is 58

17. Retrieve all Records from the Sailors1 table

18. Delete all tuples from the Sailors1 table

19. Create Boats table with the following fields and describe the table (*bid:* integer, *bname: string*, color*: string)*

20. Insert the following records in to the Boats table

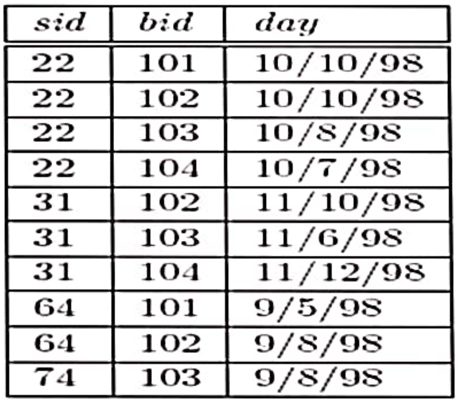


bid	*bname*	*color*
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

21. Retrieve all Records from the Boats table

22. Create Reserves table with the following fields and describe the table (*sid:* integer, *bid:* integer, *day:* date)

23. Insert the following records in to the Reserves table



sid	*bid*	*day*
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

24. Retrieve all Records from the Reserves table

List -1 Solutions

1. Create Sailors table with the following fields and describe the table (*sid:* integer, *sname:* string, *rating:* integer)

Query: Create table Sailors(sid number(2),sname varchar2(20),rating number(2))

Output: Table Created

Query: desc Sailors

Output:

Name	Null?	Type
SID		NUMBER(2)
SNAME		VARCHAR2(20)
RATING		NUMBER(2)

2. Add field (age: integer) to the Sailors table and describe the table

Query: ALTER TABLE Sailors
ADD (age NUMBER(2))

Output: Table Altered

Query: desc Sailors

Output:

Name	Null?	Type
SID		NUMBER(2)
SNAME		VARCHAR2(10)
RATING		NUMBER(2)
AGE		NUMBER(2)

3. Modify field age to real in the Sailors table and describe the table

Query: ALTER TABLE Sailors
MODIFY (age NUMBER(3,1))

Output: Table Altered

Query: desc Sailors

Output:

Name	Null?	Type
SID		NUMBER(2)
SNAME		VARCHAR2(10)
RATING		NUMBER(2)
AGE		NUMBER(3,1)

4. Delete age field from the Sailors table and describe the table

Query: ALTER TABLE Sailors
DROP COLUMN age

Output: Table Altered

Query: desc Sailors

Output:

Name	Null?	Type
SID		NUMBER(2)
SNAME		VARCHAR2(10)
RATING		NUMBER(2)

5. Add field (age: real) to the Sailors table and describe the table

Query:
```
ALTER TABLE Sailors
ADD (age NUMBER(3,1))
```
Output: Table Altered

Query: desc Sailors

Output:

Name	Null?	Type
SID		NUMBER(2)
SNAME		VARCHAR2(10)
RATING		NUMBER(2)
AGE		NUMBER(3,1)

6. Delete Sailors table

Query: DROP TABLE Sailors

Output: Table dropped.

7. Create Sailors table with the following fields and describe the table (sid: integer, sname: string, rating: integer, age: real)

Query: CREATE TABLE Sailors (sid NUMBER(2), sname VARCHAR2(20), rating NUMBER(2),age NUMBER(3,1))

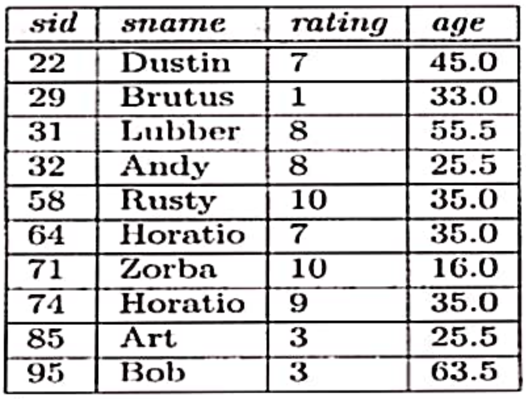
Output: Table Created

Query: desc Sailors

Output:

Name	Null?	Type
SID		NUMBER(2)
SNAME		VARCHAR2(20)
RATING		NUMBER(2)
AGE		NUMBER(3,1)

8. Insert the following records in to the Sailors table



sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Query:
```
insert into Sailors values(22,'Dustin',7,45.0);
insert into Sailors values(29,'Brutus',1,33.0);
insert into Sailors values(31,'Lubber',8,55.5);
insert into Sailors values(32,'Andy',8,25.5);
insert into Sailors values(58,'Rusty',10,35.0);
insert into Sailors values(64,'Horatio',7,35.0);
insert into Sailors values(71,'Zorba',10,16.0);
insert into Sailors values(74,'Horatio',9,35.0);
insert into Sailors values(85,'Art',3,25.5);
insert into Sailors values(95,'Bob',3,63.5);
```
Output:
```
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
```

9. Retrieve all Records from the Sailors table

Query: select * from Sailors

Output:

SID	SNAME	RATING	AGE
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25.5
95	Bob	3	63.5

10 records selected.

10. Retrieve all Records (unique) from the Sailors table

Query: select distinct sname,age from Sailors

Output:

SNAME	AGE
Andy	25.5
Art	25.5
Bob	63.5
Brutus	33
Dustin	45
Horatio	35
Lubber	55.5
Rusty	35
Zorba	16

9 records selected.

11. Retrieve the names and ages of all sailors from the Sailors table

Query: select sname,age from sailors

Output:

SNAME	AGE
Dustin	45
Brutus	33
Lubber	55.5
Andy	25.5
Rusty	35
Horatio	35
Zorba	16
Horatio	35
Art	25.5
Bob	63.5

10 records selected.

12. Find all sailors with a rating above 7

Query: select * from sailors where rating>7

Output:

SID	SNAME	RATING	AGE
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
71	Zorba	10	16
74	Horatio	9	35

5 records selected

13. Create Sailors1 table with the following fields and describe the table (sid: integer, sname: string, rating: integer, age: real)

Query: CREATE TABLE Sailors1 (sid NUMBER(2), sname VARCHAR2(20), rating NUMBER(2),age NUMBER(3,1))

Output: Table created.

Query: desc Sailors1

Output:

Name	Null?	Type
SID		NUMBER(2)
SNAME		VARCHAR2(20)
RATING		NUMBER(2)
AGE		NUMBER(3,1)

14. Insert tuples of Sailors table in to the Sailors1 table

Query: INSERT INTO Sailors1 SELECT * FROM Sailors

Output: 10 rows created.

15. Retrieve all Records from the Sailors1 table

Query: select * from Sailors1

Output:

SID	SNAME	RATING	AGE
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25.5
95	Bob	3	63.5

10 records selected

16. Delete sailor from the Sailors1 table whose sid is 58

Query: DELETE FROM Sailors1
WHERE sid=58

Output: 1 row deleted.

17. Retrieve all Records from the Sailors1 table

Query: SELECT * FROM Sailors1

Output:

SID	SNAME	RATING	AGE
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25.5
95	Bob	3	63.5

9 records selected

18. Delete all tuples from the Sailors1 table

Query: DELETE FROM Sailors1

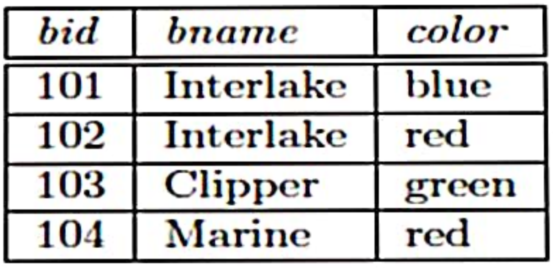
Output: 9 rows deleted.

19. Create Boats table with the following fields and describe the table
(*bid:* integer, *bname: string*, color: *string*)

Query: create table boats(bid number(3),bname varchar2(9),color varchar2(5))
Output: Table Created
Query: desc boats
Output:

Name	Null?	Type
BID		NUMBER(3)
BNAME		VARCHAR2(9)
COLOR		VARCHAR2(5)

20. Insert the following records in to the Boats table



bid	*bname*	*color*
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Query:

```
insert into boats values(101,'Interlake','blue');
insert into boats values(102,'Interlake','red');
insert into boats values(103,'Clipper','green');
insert into boats values(104,'Marine','red');
```

Output:

```
1 row created.
1 row created.
1 row created.
1 row created.
```

21. Retrieve all Records from the Boats table
Query: select * from boats
Output:

BID	BNAME	COLOR
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

4 records selected

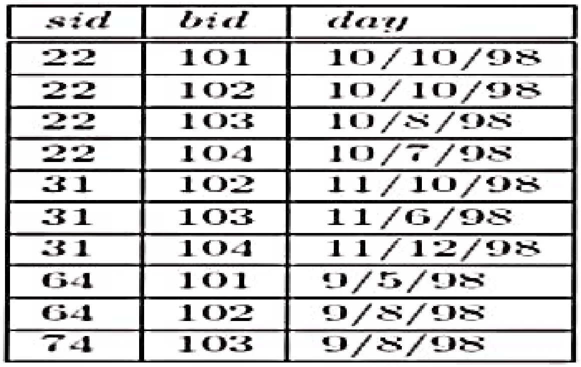
22. Create Reserves table with the following fields and describe the table
(sid: integer, bid: integer, day: date)

Query: create table reserves(sid number(2), bid number(3), day date)
Output: Table Created.
Query: desc reserves

Output:

Name	Null?	Type
SID		NUMBER(2)
BID		NUMBER(3)
DAY		DATE

23. Insert the following records in to the Reserves table



sid	*bid*	*day*
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Query:

```
insert into reserves values(22,101,'10-oct-98');
insert into reserves values(22,102,'10-oct-98');
insert into reserves values(22,103,'10-aug-98');
insert into reserves values(22,104,'10-jul-98');
insert into reserves values(31,102,'11-oct-98');
insert into reserves values(31,103,'11-jun-98');
insert into reserves values(31,104,'11-dec-98');
insert into reserves values(64,101,'9-may-98');
insert into reserves values(64,102,'9-aug-98');
insert into reserves values(74,103,'9-aug-98');
```

Output:

```
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
```

24. Retrieve all Records from the Reserves table

Query: `select * from reserves`

Output:

SID	BID	DAY
22	101	10-OCT-98
22	102	10-OCT-98
22	103	10-AUG-98
22	104	10-JUL-98
31	102	11-OCT-98
31	103	11-JUN-98
31	104	11-DEC-98
64	101	09-MAY-98
64	102	09-AUG-98
74	103	09-AUG-98

10 records selected

List2

1. Find the names of sailors who have reserved boat number 103 (using AND operator)
2. Find all sailors whose age is in between 45.0 and 63.5 (using BETWEEN operator)
3. Find all sailors whose age is in the list of values(15.0,33.2,45.7,63.5) (using IN operator)
4. Find all sailors whose second letter in the names contain 'u'
5. Find all sailors whose first letter and third letter in the names are 'A' and 'd'
6. Find all sailors from sailors and sailors1 table by using UNION operator
7. Find all sailors from sailors and sailors1 table by using INTERSECT operator
8. Find all sailors from sailors and sailors1 table by using MINUS operator
9. Find the names of sailors who have reserved boat 103 (using nested query)
10. Find the names of sailors who have reserved a blue boat (using nested query)
11. Find the names of sailors who have not reserved a red boat (using nested query)
12. Find the names of sailors who have reserved boat 103 (using Correlated Nested Query)
13. Find sailors whose rating is better than some sailor called Horatio (using ANY operator)
14. Find sailors whose rating is better than some sailor called Horatio (using ALL operator)
15. Find the sailors with the highest rating
16. Find the average age of all sailors
17. Find the minimum age of the sailor
18. Find the maximum age of the sailor
19. Count the number of sailors
20. Count the number of different sailor names

List -2 Solutions

1. Find the names of sailors who have reserved boat number 103 (using AND operator)

Query: select sname from Sailors S,reserves R
where S.sid=R.sid and R.bid=103;

Output:

SNAME
Dustin
Lubber
Horatio

2. Find all sailors whose age is in between 45.0 and 63.5 (using BETWEEN operator)

Query: select * from Sailors
where age between 45.0 and 63.5

Output:

SID	SNAME	RATING	AGE
31	Lubber	8	55.5
95	Bob	3	63.5
22	Dustin	7	45

3. Find all sailors whose age is in the list of values (15.0,33.2,45.7,63.5) (using IN operator)

Query: select * from Sailors
where age in(15.0,33.2,45.7,63.5)

Output:

SID	SNAME	RATING	AGE
95	Bob	3	63.5

4. Find all sailors whose second letter in the names contain 'u'

Query: select * from Sailors
where sname like '_u%'

Output:

SID	SNAME	RATING	AGE
31	Lubber	8	55.5
58	Rusty	10	35
22	Dustin	7	45

5. Find all sailors whose first letter and third letter in the names are 'A' and 'd'

Query: select * from Sailors
where sname like 'A_d%'

Output:

SID	SNAME	RATING	AGE
32	Andy	8	25.5

6. Find all sailors from sailors and sailors1 table by using (UNION operator)

Query:
```
select * from Sailors
 union select * from Sailors1
```
Output:

SID	SNAME	RATING	AGE
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25.5
95	Bob	3	63.5

10 rows selected.

7. Find all sailors from sailors and sailors1 table by using INTERSECT operator

Query:
```
select * from Sailors
 intersect select * from Sailors1
```
Output:

SID	SNAME	RATING	AGE
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25.5
95	Bob	3	63.5

8. Find all sailors from sailors and sailors1 table by using MINUS operator

Query:
```
select * from Sailors
 minus   select * from Sailors1
```
Output:

SID	SNAME	RATING	AGE
58	Rusty	10	35

9. Find the names of sailors who have reserved boat 103 (using nested query)

Query:
```
select sname from Sailors
 where sid in(select sid from reserves where bid =103);
```
Output:

SNAME
Dustin
Lubber
Horatio

10. Find the names of sailors who have reserved a blue boat (using nested query)

Query:
```
select sname from Sailors
 where sid in(select sid from reserves where bid in(select bid from boats
  where color='blue'));
```
Output:

SNAME
Dustin
Horatio

11. Find the names of sailors who have not reserved a red boat (using nested query)

Query:
```
select sname from Sailors
        where sid in(select sid from reserves where bid in(select bid from boats
         where color!='red'));
```

Output:

SNAME
Dustin
Lubber
Horatio
Horatio

12. Find the names of sailors who have reserved boat 103 (using Correlated Nested Query)

Query:
```
select s.sname from Sailors s
        where exists(select * from reserves r where
        r.bid=103 and r.sid=s.sid);
```

Output:

SNAME
Lubber
Horatio
Dustin

13. Find sailors whose rating is better than some sailor called Horatio (using ANY operator)

Query:
```
select * from Sailors
         where rating> any(select rating from Sailors where sname='Horatio');
```

Output:

SID	SNAME	RATING	AGE
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
71	Zorba	10	16
74	Horatio	9	35

14. Find sailors whose rating is better than some sailor called Horatio (using ALL operator)

Query:
```
select * from Sailors
         where rating> all(select rating from Sailors where sname='Horatio');
```

Output:

SID	SNAME	RATING	AGE
58	Rusty	10	35
71	Zorba	10	16

15. Find the sailors with the highest rating

Query:
```
select * from Sailors
         where rating=(select max(rating) from Sailors)
```

Output:

SID	SNAME	RATING	AGE
58	Rusty	10	35
71	Zorba	10	16

16. Find the average age of all sailors

Query:
```
select avg(age) from Sailors
```

Output:

AVG(AGE)
36.9

17. Find the minimum age of the sailor

Query:
```
select min(age) from Sailors
```

Output:

MIN(AGE)
16

18. Find the maximum age of the sailor

Query:
```
select max(age) from Sailors
```

Output:

MAX(AGE)
63.5

19. Count the number of sailors
Query: select count(*) from Sailors
Output:

```
COUNT(*)
   10
```

20. Count the number of different sailor names
Query: select count(distinct sname) from Sailors
Output:

```
COUNT(DISTINCT SNAME)
          9
```

List – 3

1. Find the sum of ratings of all sailors
2. Find the second maximum age of the sailor
3. Find the name of the sailor whose age is second maximum
4. Find the name and age of the oldest sailor
5. Find the number of sailors belongs to each rating level
6. Find the age of the youngest sailor for each rating level
7. Find the age of the youngest sailor for each rating level, which is greater than 7
8. Find the age of the youngest sailor who is eligible to vote (i.e., is at least 18 years old) for each rating level with at least two such sailors
9. Find the average age of sailors for each rating level that has at least two sailors
10. Drop sailors table
11. Create Sailors table with the following fields and describe the table (sid: integer, sname: string, rating: integer, age: real)

Note: Underlined attribute represents the primary key

12. Insert records in to the Sailors table
13. Drop Reserves table
14. Create Reserves table with the following fields and describe the table (*sid:* integer, *bid:* integer, *day:* date)
15. Insert records in to the Reserves table
16. Drop Boats table
17. Create Boats table with the following fields and describe the table (*bid:* integer, *bname: string*, color: *string*).
18. Insert records in to the Boats table
19. For each red boat, find the number of reservations for this boat.
20. Find those ratings for which the average age of sailors is the minimum over all ratings
21. Count the number of sailors who have reserved boats between '9/5/98' and '10/10/98'
22. Find the names of sailors who have reserved boats between '9/4/98' and '1/8/98'
23. Find the number of reservations made for each boat
24. Find the number of reservations (with at least three) made for each boat
25. Find the number of reservations made for each boat during '9/5/98' and '10/10/98'
26. Find the days on which blue color boat is reserved
27. Find the days on which 'Horatio' reserved the boats
28. Find the number of boat reservations made by each sailor
29. Find the number of boats in each color
30. Find the colors of boats reserved by 'Lubber'

LIST – 3 Solutions

1. Find the sum of ratings of all sailors
Query: select sum(rating) from Sailors
Output:

```
SUM(RATING)
    66
```

2. Find the second maximum age of the sailor
Query: select max(age) from Sailors
where age!=(select max(age) from Sailors)
Output:

```
MAX(AGE)
  55.5
```

3. Find the name of the sailor whose age is second maximum

Query:
```
select sname from Sailors
    where age=(select max(age) from Sailors
    where age!=(select max(age) from Sailors));
```

Output:

SNAME
Lubber

4. Find the name and age of the oldest sailor

Query:
```
select sname,age from Sailors
    where age=(select max(age) from Sailors)
```

Output:

SNAME	AGE
Bob	63.5

5. Find the number of sailors belongs to each rating level

Query:
```
select rating,count(rating) from Sailors
    group by rating
```

Output:

RATING	COUNT(RATING)
1	1
3	2
7	2
8	2
9	1
10	2

6. Find the age of the youngest sailor for each rating level

Query:
```
select rating,min(age) from Sailors
    group by rating
```

Output:

RATING	MIN(AGE)
1	33
3	25.5
7	35
8	25.5
9	35
10	16

7. Find the age of the youngest sailor for each rating level, which is greater than 7

Query:
```
select rating,min(age) from Sailors
    group by rating
    having rating>7
```

RATING	MIN(AGE)
8	25.5
9	35
10	16

8. Find the age of the youngest sailor who is eligible to vote (i.e., is at least 18 years old) for each rating level with at least two such sailors

Query:
```
select rating,min(age) from Sailors
where age>=18
group by rating
having count (rating)>=2
```

Output:

RATING	MIN(AGE)
3	25.5
7	35
8	25.5

9. Find the average age of sailors for each rating level that has at least two sailors

Query:
```
select rating,avg(age) from Sailors
group by rating
having count (rating)>=2
```

Output:

RATING	AVG(AGE)
3	44.5
7	40
8	40.5
10	25.5

10. Drop sailors table

Query: DROP TABLE Sailors

Output: Table dropped.

11. Create Sailors table with the following fields and describe the table (sid: integer, sname: string, rating: integer, age: real)

Note: Underlined attribute represents the primary key

Query: CREATE TABLE Sailors (sid NUMBER(2) primary key, sname VARCHAR2(20),rating NUMBER(2),age NUMBER(3,1))

Output:

Table created.

Query: desc Sailors

Output:

Name	Null?	Type
SID	NOT NULL	NUMBER(2)
SNAME		VARCHAR2(20)
RATING		NUMBER(2)
AGE		NUMBER(3,1)

12. Insert records in to the Sailors table

Query:
```
insert into Sailors values(22,'Dustin',7,45.0);
insert into Sailors values(29,'Brutus',1,33.0);
insert into Sailors values(31,'Lubber',8,55.5);
insert into Sailors values(32,'Andy',8,25.5);
insert into Sailors values(58,'Rusty',10,35.0);
insert into Sailors values(64,'Horatio',7,35.0);
insert into Sailors values(71,'Zorba',10,16.0);
insert into Sailors values(74,'Horatio',9,35.0);
insert into Sailors values(85,'Art',3,25.5);
insert into Sailors values(95,'Bob',3,63.5);
```

Output:
```
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
```

13. Drop Reserves table

Query: DROP TABLE reserves

Output: Table dropped.

14. Create Reserves table with the following fields and describe the table (*sid:* integer, *bid:* integer, *day:* date)

Query: create table reserves(sid number(2), bid number(3), day date,primary key(sid,bid))

Output: Table Created.

Query: desc reserves

Output:

Name	Null?	Type
SID	NOT NULL	NUMBER(2)
BID	NOT NULL	NUMBER(3)
DAY		DATE

15. Insert records in to the Reserves table

Query:

```
insert into reserves values(22,101,'10-oct-98');
insert into reserves values(22,102,'10-oct-98');
insert into reserves values(22,103,'10-aug-98');
insert into reserves values(22,104,'10-jul-98');
insert into reserves values(31,102,'11-oct-98');
insert into reserves values(31,103,'11-jun-98');
insert into reserves values(31,104,'11-dec-98');
insert into reserves values(64,101,'9-may-98');
insert into reserves values(64,102,'9-aug-98');
insert into reserves values(74,103,'9-aug-98');
```

Output:

```
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
```

16. Drop Boats table

Query: drop table boats

Output: table dropped.

17. Create Boats table with the following fields and describe the table (*bid:* integer, *bname: string*, color*: string*).

Query: create table boats(bid number(3) primary key,bname varchar2(9),color varchar2(5))

Output: Table Created

Query: desc boats

Output:

Name	Null?	Type
BID	NOT NULL	NUMBER(3)
BNAME		VARCHAR2(9)
COLOR		VARCHAR2(5)

18. Insert records in to the Boats table

Query:

```
  insert into boats values(101,'Interlake','blue');
  insert into boats values(102,'Interlake','red');
  insert into boats values(103,'Clipper','green');
  insert into boats values(104,'Marine','red');
```

Output:

```
1 row created.
1 row created.
1 row created.
1 row created.
```

19. For each red boat, find the number of reservations for this boat.

Query:

```
select bid,count(bid) from reserves
        where bid in(select bid from boats where color='red')
        group by bid
```

Output:

BID	COUNT(BID)
102	3
104	2

20. Find those ratings for which the average age of sailors is the minimum over all ratings

Query:
```
select temp1.rating,temp1.ave  age from (select rating,avg(age) ave from sailors group by rating) temp1
      where temp1.ave=(select min(temp2.ave) from (select rating,avg(age) ave from sailors group by rating) temp2)
```

Output:

RATING	AGE
10	25.5

21. Count the number of sailors who have reserved boats between '9/5/98' and '10/10/98'

Query:
```
select count(sid) from reserves
      where day between '9-may-98' and '10-oct-98'
```

Output:

COUNT(SID)
8

22. Find the names of sailors who have reserved boats between '9/4/98' and '1/8/98'

Query:
```
select sname from sailors
      where sid in(select sid from reserves  where day between '9-apr-98' and '1-aug-            98')
```

Output:

SNAME
Dustin
Lubber
Horatio

23. Find the number of reservations made for each boat

Query:
```
select bid, count(bid) from reserves
      group by bid
```

Output:

BID	COUNT(BID)
101	2
102	3
103	3
104	2

24. Find the number of reservations (with at least three) made for each boat

Query:
```
select bid,count(bid) from reserves
      group by bid
      having count(bid)>=3
```

Output:

BID	COUNT(BID)
102	3
103	3

25. Find the number of reservations made for each boat during '9/5/98' and '10/10/98'

Query:
```
select bid,count(bid) from reserves
      where day between '9-may-98' and '10-oct-98'
      group by bid
```

Output:

BID	COUNT(BID)
101	2
102	2
103	3
104	1

26. Find the days on which blue color boat is reserved

Query:
```
select day from reserves where bid in( select bid from boats where color='blue')
```

Output:

DAY
10-OCT-98
09-MAY-98

27. Find the days on which 'Horatio' reserved the boats

Query: select day from reserves where sid in(select sid from sailors where sname='Horatio')

Output:

DAY
09-MAY-98
09-AUG-98
09-AUG-98

28. Find the number of boat reservations made by each sailor

Query: select s.sname,count(*)from sailors s, reserves r
    where s.sid=r.sid
    group by s.sname;

Output:

SNAME	COUNT(*)
Dustin	4
Horatio	3
Lubber	3

29. Find the number of boats in each color

Query: select count(bid),color from boats
    group by color

Output:

COUNT(BID)	COLOR
1	blue
1	green
2	red

30. Find the colors of boats reserved by 'Lubber'

Query: select distinct color from boats where bid in
    (select bid from reserves where sid in
    (select sid from sailors where sname='Lubber'))

Output:

COLOR
green
red

List 4

1.Create **Persons** table with the following fields (declare **FirstName** as **DEFAULT 'Ola'**) and describe the table

Pid	FirstName	LastName	Address	City
1	Ola	Hansen	Timoteivn 10	Sandnes
2	Tove	Svendson	Borgvn 23	Sandnes
3	Kari	Pettersen	Storgt 20	Stavanger

2. Specify **UNIQUE** constraint on field **LastName** in the Persons table
3. Specify **PRIMARY KEY** constraint on field **Pid** in the Persons table
4. Create **Orders** table with the following fields (declare **Oid** as **Primary Key** and **Pid** as **Foreign Key**) and describe the table

Oid	OrderNo	Pid
1	77895	3
2	44678	3
3	22456	2
4	24562	1

31. Insert the Persons records in to the Persons table
32. Insert the Orders records in to the Orders table
33. Insert (4,'Andy',**'Hansen'**,'Storgt 20','Stavanger') record in to the Persons table and observe the result
34. Insert (**3**,'Andy','Tahsen','Storgt 20','Stavanger') record in to the Persons table and observe the result
35. Insert (5,51212,4) record in to the Orders table and observe the result
36. Delete record from the Persons table whose Pid is 3 and observe the result
37. Give example which includes all Arithmetic functions
38. Give example which includes all String functions
39. Give example which includes all Date and Time functions
40. Retrieve tuples from the reserves table (display 'day' as dd-mon-yyyy)
41. Retrieve tuples from the boats table, in which trim 'In' from left side on 'bname'
42. Retrieve tuples from the sailors table, in which trim 'io' from right side on 'sname'
43. Convert '2010 10 feb' in to date
44. Give example which includes all General functions

List-4 Solutions

1. Create Persons table with the following fields (declare FirstName as DEFAULT 'Ola') and describe the table

Pid	FirstName	LastName	Address	City
1	Ola	Hansen	Timoteivn 10	Sandnes
2	Tove	Svendson	Borgvn 23	Sandnes
3	Kari	Pettersen	Storgt 20	Stavanger

Query:

create table Persons(Pid number(5),FirstName varchar2(10) default 'Ola',LastName varchar2(10), Address varchar2(15),City varchar2(10));

Output:

Table created.

2. Specify UNIQUE constraint on field LastName in the Persons table

Query:

alter table Persons add unique(LastName);

Output:

Table altered.

3. Specify PRIMARY KEY constraint on field Pid in the Persons table

Query:

alter table Persons add primary key(Pid);

Output:

Table altered.

4. Create Orders table with the following fields (declare Oid as Primary Key and Pid as Foreign Key) and describe the table

Oid	OrderNo	Pid
1	77895	3
2	44678	3
3	22456	2
4	24562	1

Query:

create table Orders (Oid number(2) PRIMARY KEY,OrderNo number(5),Pid number(2),FOREIGN KEY(Pid) REFERENCES Persons(Pid));

Output:

Table created.

5. **Insert Persons records in to Persons table**

Query:

```
insert into persons(pid,lastname,address,city) values(1,'Hansen','Timotevin 10','Sandnes');
```

Output:

```
1 row created.
```

Query:

```
Select * from persons;
```

Output:

PID	FIRSTNAME	LASTNAME	ADDRESS	CITY
1	Ola	Hansen	Timotevin 10	Sandnes

Query:

```
insert into persons values(2,'Tove','Svendson','Borgvn 23','Sandnes');
insert into persons values(3,'Kari','Pettersen','Storgt 20','Stavanger');
```

Query:

```
Select * from persons;
```

Output:

PID	FIRSTNAME	LASTNAME	ADDRESS	CITY
1	Ola	Hansen	Timotevin 10	Sandnes
2	Tove	Svendson	Borgvn 23	Sandnes
3	Kari	Pettersen	Storgt 20	Stavanger

6. **Insert Orders records in to Orders table**

Query:

```
insert into orders values(1,77895,3);
insert into orders values(2,44678,3);
insert into orders values(3,22456,2);
insert into orders values(4,24562,1);
```

Output:

OID	ORDERNO	PID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

7. **Insert (4,'Andy','Hansen','Storgt 20','Stavanger') record in to Persons table and observe the result**

Query:

```
insert into persons values(4,'Andy','Hansen','Storgt 20','Stavanger');
```

Output:

```
ERROR: ORA-00001: unique constraint (SCOTT.SYS_C005118) violated
```

8. **Insert (3,'Andy','Tahsen','Storgt 20','Stavanger') record in to Persons table and observe the result**

Query:

```
insert into persons values(3,'Andy','Tahsen','Storgt 20','Stavanger');
```

Output:

```
ERROR: ORA-00001: unique constraint (SCOTT.SYS_C005119) violated
```

9. **Insert (5,51212,4) record in to Orders table and observe the result**

Query:

```
insert into orders values(5,51212,4);
```

Output:

```
ERROR: ORA-02291: integrity constraint (SCOTT.SYS_C005121) violated - parent key not found
```

10. **Delete record from the Persons table whose Pid is 3 and observe the result**

Query:

```
delete from Persons where Pid=3;
```

Output:

```
ERROR: ORA-02292: integrity constraint (SCOTT.SYS_C005121) violated - child record found
```

11. Give example which includes all Arithmetic functions

Query:

```
select abs(4.55),ceil(4.55),floor(4.55), mod(5,4),power(2,3),sign(-10), sign(10), sqrt(9),round(4.333,2),trunc(4.333,2) from dual
```

Output:

ABS(4.55)	CEIL(4.55)	FLOOR(4.55)	MOD(5,4)	POWER(2,3)	SIGN(-10)	SIGN(10)	SQRT(9)	ROUND(4.333,2)	TRUNC(4.333,2)
4.55	5	4	1	8	-1	1	3	4.33	4.33

12. Give example which includes all String functions

Query:

```
select ascii('A'),chr(65),concat('abc','def'),instr('abaabcab','ab',1,2),substr('abcd',2) from dual
```

Output:

ASCII('A')	C	CONCAT	INSTR('ABAABCAB','AB',1,2)	SUB
65	A	abcdef	4	bcd

```
select length('abcd'),lengthb('abcde'),lower('ABCD'),lpad('abcd',7,'e'),
ltrim('abcd','a') from dual
```

Output:

LENGTH('ABCD')	LENGTHB('ABCDE')	LOWE	LPAD('A	LTR
4	5	abcd	eeeabcd	bcd

```
select replace('abcd','a','ef'),rtrim('abcd','d'),rpad('abcd',7,'e'),to_char(123) from dual
```

Output:

REPLA	RTR	RPAD('A	TO_
efbcd	abc	abcdeee	123

```
select substr('abcdef',2,3),translate('abcdef','bc','ef'),upper('abcd'),initcap('abcd') from dual
```

Output:

SUB	TRANSL	UPPE	INIT
bcd	aefdef	ABCD	Abcd

13. Give example which includes all Date and Time functions

Query:

```
select add_months(current_date,5),current_date,current_timestamp(2),to_date('10-10-2011','dd-mm-yy'),months_between('10-oct-2011','10-oct-2010'),
EXTRACT(year from current_date),to_char(current_date,'mon dd yyyy')
from dual
```

Output:

ADD_MONTH	CURRENT_D	CURRENT_TIMESTAMP(2)	TO_DATE('	MONTHS_BETWEEN('10-OCT-2011','10-OCT-2010')	EXTRACT(YEARFROMCUR
10-APR-13	10-NOV-12	10-NOV-12 02.25.41.00 PM +05:30	10-OCT-11	12	2012

14. **Retrieve tuples from the reserves table (display 'day' as dd-mon-yyyy)**

Query:

```
select sid,bid,to_char(day,'dd-mon-yyyy') day from reserves
```

Output:

SID	BID	DAY
22	101	10-OCT-1998
22	102	10-OCT-1998
22	103	10-AUG-1998
22	104	10-JUL-1998
31	102	11-OCT-1998
31	103	11-JUN-1998
31	104	11-DEC-1998
64	101	09-MAY-1998
64	102	09-AUG-1998
74	103	09-AUG-1998

15. **Retrieve tuples from the boats table, in which trim 'In' from left side on 'bname'**

Query:

```
select ltrim(bname,'In') from boats
```

Output:

LTRIM(BNAME,'IN)
terlake
terlake
Clipper
Marine

16. **Retrieve tuples from the sailors table, in which trim 'io' from right side on 'sname'**

Query:

```
select rtrim(sname,'io') from sailors
```

Output:

RTRIM(SNAM
Dustin
Brutus
Lubber
Andy
Rusty
Horat
Zorba
Horat
Art
Bob

17. **Convert '2010 10 feb' in to date**

Query:

```
select to_date('2010 10 feb','yyyy dd mon') todate from dual
```

Output:

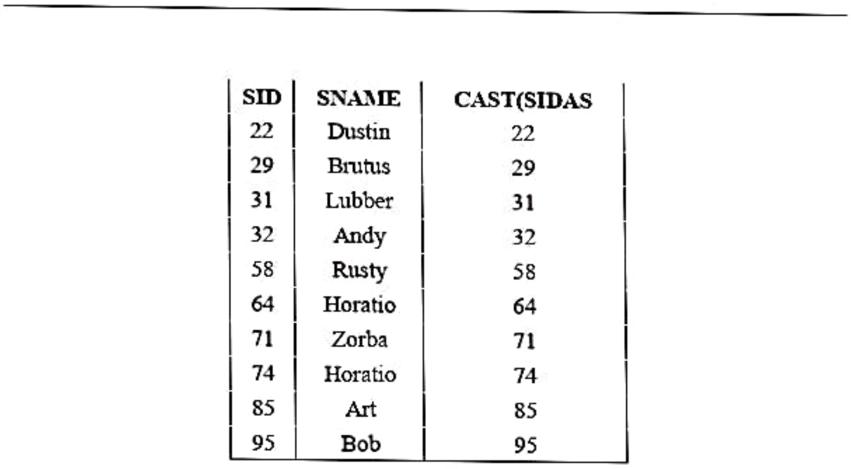
TODATE
10-FEB-10

18. Give example which includes all General functions

Query:

select sid,sname,cast(sid as varchar2(10)) from sailors

Output:



SID	SNAME	CAST(SIDAS
22	Dustin	22
29	Brutus	29
31	Lubber	31
32	Andy	32
58	Rusty	58
64	Horatio	64
71	Zorba	71
74	Horatio	74
85	Art	85
95	Bob	95

Query:

create table cal(id number);

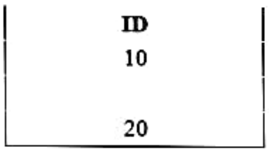
Output:

Table created.

Query:

insert into cal values(10);
insert into cal(id) values(null);
insert into cal values(20);

Query:



ID
10
20

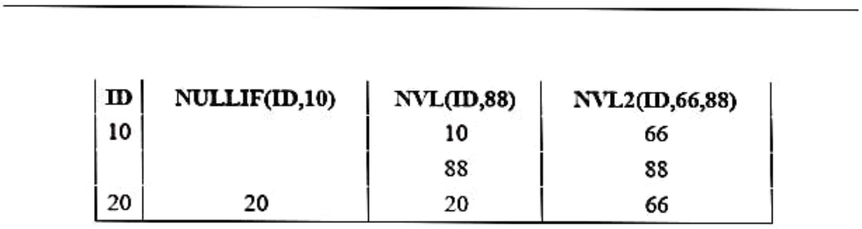
select * from cal

Output:

Query:

select id,nullif(id,10),nvl(id,88),nvl2(id,66,88) from cal

Output:

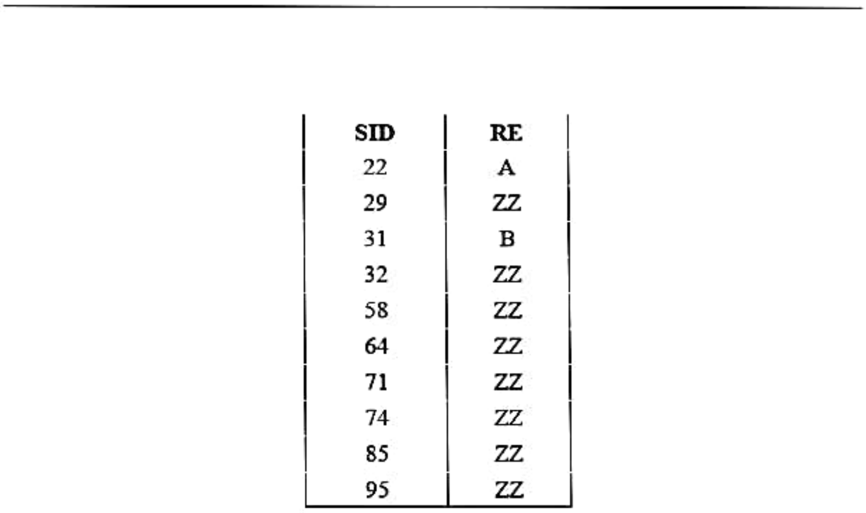


ID	NULLIF(ID,10)	NVL(ID,88)	NVL2(ID,66,88)
10		10	66
		88	88
20	20	20	66

Query:

select sid,decode(sid,22,'A',31,'B','ZZ') results from sailors

Output:



SID	RE
22	A
29	ZZ
31	B
32	ZZ
58	ZZ
64	ZZ
71	ZZ
74	ZZ
85	ZZ
95	ZZ

List 5

1. Write a PL/SQL program to perform the following:

   Insert <99,'John',9,25.6> into the Sailors table, Update sname 'John' to 'Joe' in the Sailors table, Delete tuple from the Sailors table whose sname is 'Joe'

2. Retrieve tuple from the Sailors table whose sname is 'Lubber' using PL/SQL program
3. Write a PL/SQL program to show handling of Pre-defined exception
4. Write a PL/SQL program to show handling of User-defined exception
5. Write a PL/SQL program to print the following values

   0 if a=0
   1 if a>0
   -1 if a<0

6. Write a PL/SQL program to retrieve records from the sailors table using cursor
7. Write a PL/SQL procedure to find the rating of a sailor with the given sailor id.
8. Write a PL/SQL function to return sailor name with the given sailor id.
9. Create a trigger to convert sname to upper case before inserting a record.
10. Create a trigger to display the rating difference between the new values and old values.
11. Explain Packages with a PL/SQL program

List-5 Solutions

1. Write a PL/SQL program to perform the following: Insert <99,'John',9,25.6> into the Sailors table, Update sname 'John' to 'Joe' in the Sailors table, Delete tuple from the Sailorstable whose sname is 'Joe'

Program:

```
DECLARE
s_id number(2):=99;
  s_name varchar2(20):='john';
  s_rating number(2):=9;
  s_age number(3,1):=25.6;
BEGIN
  INSERT INTO Sailors VALUES(s_id,s_name,s_rating,s_age);
end;
```

Output:

```
SID   SNAME        RATING  AGE
22    Dustin  7           45
29    Brutus  1           33
31    Lubber       8           55.5
32    Andy         8           25.5
58    Rusty  10           35
64    Horatio      7           35
71    Zorba  10           16
74    Horatio      9           35
85    Art          3           25.5
95    Bob          3           63.5
99    john         9           25.6
11 rows selected.
```

Program:

```
begin
 update sailors set sname='joe' where sname='john';
end;
```

Output:

```
SID   SNAME        RATING     AGE
22    Dustin 7            45
29    Brutus 1            33
31    Lubber       8           55.5
32    Andy         8           25.5
58    Rusty  10           35
64    Horatio       7          35
71    Zorba  10           16
74    Horatio      9           35
85    Art          3           25.5
95    Bob          3           63.5
99    joe          9           25.6
11 rows selected.
```

Program:

```
begin
   delete from sailors where sname='joe';
end;
```

Output:

```
SID   SNAME        RATING     AGE
22    Dustin 7            45
29    Brutus 1            33
31    Lubber       8           55.5
32    Andy         8           25.5
58    Rusty  10           35
64    Horatio       7          35
71    Zorba  10           16
74    Horatio      9           35
85    Art          3           25.5
95    Bob          3           63.5
10 rows selected.
```

2. Retrieve tuple from the Sailors table whose sname is 'Lubber' using PL/SQL program

Program:

```
declare
   s_id sailors.sid%type;
   s_sname sailors.sname%type;
   s_rating sailors.rating%type;
   s_age sailors.age%type;
begin
select sid,sname,rating,age into s_id,s_sname,s_rating,s_age from sailors          where sname='lubber';
dbms_output.put_line('sid is '||s_id||' sname is'||s_sname||' rating'||s_rating||' age i     s'||s_age);
end;
```

Output:

```
sid is 31 sname is lubber rating is 8 age is 55.5
```

3. Write a PL/SQL program to show handling of Pre-defined exception Program:

```
declare
a number:=4;
c number;
begin
c:=a/0;
exception
when ZERO_DIVIDE then
dbms_output.put_line('divide by zero exception');
end;
```

Output:

```
divide by zero exception
```

4. Write a PL/SQL program to show handling of User-defined exception

Program:

```
declare
a number(2);
b number(2);
c number(2);
ex exception;
begin
a:=&a;
b:=&b;
if(b=0) then
raise ex;
else
c:=a/b;
dbms_output.put_line(c);
end if;
exception
when ex then
dbms_output.put_line('divide by zero');
end;
```

Output:

```
old 8: b:=&b;
new 8: b:=0;
divide by zero
```

5. Write a PL/SQL program to print the following values

```
 0   if a=0
 1   if a>0
-1   if a<0
```

Program:

```
declare
a number:=&a;
begin
if a=0 then
dbms_output.put_line('0');
elsif a>0 then
dbms_output.put_line('1');
else
dbms_output.put_line('-1');
end if;
end;
```

Output:

```
Enter value for a: 5
old   2: a number:=&a;
new   2: a number:=5;
1
```

6. Write a PL/SQL program to retrieve records from the sailors table using cursor

Program:

```
declare
s_id number(2);
s_name varchar2(20);
s_rating number(2);
s_age number(3,1);
cursor c_sailors is select sid,sname,rating,age from sailors;
begin
open c_sailors;
loop
fetch c_sailors into s_id,s_name,s_rating,s_age;
dbms_output.put_line(s_id||' '||s_name||' '||s_rating||' '||s_age);
exit when c_sailors%notfound;
end loop;
close c_sailors;
end;
```

Output:

```
22 dustin 7 45
29 brutus 1 33
31 lubber 8 55.5
32 andy 8 25.5
58 rusty 10 35
64 horatio 7 35
71 zorba 10 16
74 horatio 9 35
85 art 3 25.5
95 bob 3 63.5
```

7. Write a PL/SQL procedure to find the rating of a sailor with the given sailor id.

Procedure:

```
create or replace procedure findrating(s_id in number, s_rating out number) is
begin
select rating into s_rating from sailors where sid=s_id;
end findrating;
```

Program:

```
declare
id sailors.sid%type;
srating sailors.rating%type;
begin
id:=&id;
findrating(id,srating);
dbms_output.put_line(id||' rating is '||srating);
end;
```

Output:

```
Enter value for id: 22
old   5: id:=&id;
new   5: id:=22;
22 rating is 7
```

8. Write a PL/SQL function to return sailor name with the given sailor id.

Function:

```
create or replace function findname(s_id in number) return varchar2 is
name sailors.sname%type;
begin
select sname into name from sailors where sid=s_id;
return name;
end findname;
```

Program:

```
declare
id sailors.sid%type;
name sailors.sname%type;
begin
id:=&id;
name:=findname(id);
dbms_output.put_line('Sailor name is '||name);
end;
```

Output:

```
Enter value for id: 31
old   5: id:=&id;
new   5: id:=31;
Sailor name is Lubber
```

9. Create a trigger to convert sname to upper case before inserting a record.

Program:

```
create or replace trigger triuppersname
before
insert on sailors
for each row
begin
:new.sname := upper(:new.sname);
end;
```

Query: insert into sailors values(99,'uwxyz',7,43);

Query: select * from sailors;

Output:

```
SID   SNAME       RATING      AGE
22    Dustin 7          45
29    Brutus 1          33
31    Lubber      8           55.5
32    Andy        8           25.5
58    Rusty  10         35
64    Horatio      7          35
71    Zorba  10         16
74    Horatio     9           35
85    Art         3           25.5
95  Bob           3           63.5
99   UWXYZ    7         43
```

10. Create a trigger to display the rating difference between the new values and old values.

Program:

```
create or replace trigger displaydiff
before insert or update or delete on sailors
for each row
when (new.sid>0)
declare
 rating_diff number;
begin
 rating_diff := :new.rating - :old.rating;
 dbms_output.put_line('Old rating: '|| :old.rating);
 dbms_output.put_line('New rating: '|| :new.rating);
 dbms_output.put_line('Rating difference: '|| rating_diff);
end;
```

Query: insert into sailors values(97,'abc',8,32);

Output:
```
Old rating:
 New rating: 8
Rating difference:
```

Query:
```
update sailors
 set rating = rating + 2
 where sid = 99;
```

Output:
```
Old rating: 7
 New rating: 9
Rating difference: 2
```

11. Explain Packages with a PL/SQL program

Package Specification:

```
CREATE OR REPLACE PACKAGE DEMO AS
-- Function specs goes here
FUNCTION findname(s_id in number) return varchar2;
-- Procedure specs goes here
PROCEDURE findrating(s_id in number, s_rating out number);
END DEMO;
```

Output:

```
Package created.
```

Package Body:

```
CREATE OR REPLACE PACKAGE BODY DEMO AS
FUNCTION findname(s_id in number) return varchar2 is
name sailors.sname%type;
begin
select sname into name from sailors where sid=s_id;
return name;
END findname;
PROCEDURE findrating(s_id in number, s_rating out number) is
begin
select rating into s_rating from sailors where sid=s_id;
END findrating;
END DEMO;
```

Output:

```
Package body created.
```

Program:

```
DECLARE
    id sailors.sid%type;
    name sailors.sname%type;
    srating sailors.rating%type;
BEGIN
    id:=&id;
    demo.findrating(id,srating);
    dbms_output.put_line('Sailor rating is '||srating);
    name:=demo.findname(id);
    dbms_output.put_line('Sailor name is '||name);
END;
```

Output:

```
Enter value for id: 22
old   6:    id:=&id;
new   6:    id:=22;
Sailor rating is 7
Sailor name is Dustin
```