

```

1  SET SERVEROUTPUT ON
2
3  select user from dual;
4  SELECT NAME FROM v$database;
5  select table_name from dba_tables;
6  select owner, table_name from all_tables;
7  select table_name FROM user_tables;
8  SELECT table_name FROM user_tables ORDER BY table_name;
9
10 create table Stud(rollno int primary key,
11                  name char(10),
12                  mark1 float,
13                  mark2 float,
14                  mark3 float);
15
16  /* ROLLNO NAME MARK1 MARK2 MARK3
17  -----
18  1 aparna 80 90 78
19  2 amritha 90 92 81
20  3 binuja 23 18 20
21  4 cathy 49 50 50
22  5 danish 60 62 61
23  6 fayas 76 62 74 */
24
25  insert into Stud values(1,'aparna', 80, 90, 78);
26  insert into Stud values(&rollno, '&name', &mark1, &mark2, &mark3);
27  select * from Stud;
28  select name from Stud where rollno=1;
29
30
31  DECLARE
32      age integer;
33      name VARCHAR(20);
34  BEGIN
35      dbms_output.put_line('Hello world');
36      --dbms_output.put_line('age = ' || age);
37      --dbms_output.put_line('name = ' || name);
38      --insert into Stud values(&rollno, '&name', &mark1, &mark2, &mark3);
39  END;
40
41  select * from Stud;
42  delete from Stud;
43  drop table Stud purge;
44
45  /* addition of two numbers */
46  DECLARE
47      a integer := &a;
48      b integer := &b;
49      res int;
50  BEGIN
51      res := a + b;
52      dbms_output.put_line('result = ' || res);
53  END;
54
55  /*
56  IF-THEN, IF-THEN-ELSE, IF-THEN
57  ELSIF, CASE, WHILE
58  */
59
60  -- find the largest of two numbers
61  DECLARE
62      a integer := &a;
63      b integer := &b;
64  BEGIN
65
66      if (a > b) then
67          dbms_output.put_line(a || ' is the largest number');
68      else
69          dbms_output.put_line(b || ' is the largest number');

```

```

70     end if;
71 END;
72
73
74 /* Use of if elsif ladder */
75 /* calculate the range of input number */
76 DECLARE
77     c integer := &c;
78 BEGIN
79     if (c >= 0 and c < 10) then
80         dbms_output.put_line(' is less than 10');
81     elsif (c >= 10 and c < 20) then
82         dbms_output.put_line(' is less than 20');
83     elsif (c >= 20 and c < 30) then
84         dbms_output.put_line(' is less than 30');
85     else
86         dbms_output.put_line(' is grater than or equal 30');
87     end if;
88 END;
89
90
91 /* Case statement */
92 /*
93     Grading ...!
94     A- > print "Excellent"
95     B- > print "Very good"
96     C- > print "Well done"
97     D- > print "You passed"
98     F- > print "Better try again"
99     others - "No such grade"
100
101 */
102 DECLARE
103     c char(1) := '&c';
104 BEGIN
105     case c
106     when 'A' then dbms_output.put_line('Excellent');
107     when 'B' then dbms_output.put_line('Very good');
108     when 'C' then dbms_output.put_line('Well done');
109     when 'D' then dbms_output.put_line('You passed');
110     when 'F' then dbms_output.put_line('Better try again');
111     else dbms_output.put_line('No such grade');
112     end case;
113 END;
114
115
116 /* use of array*/
117 /* index starts with 1 .. not 0 */
118 DECLARE
119     type intArray IS VARRAY(10) OF INTEGER;
120     type namesArray IS VARRAY(5) OF VARCHAR2(5);
121
122     arr intArray;
123     names namesArray;
124     i integer;
125
126 BEGIN
127     arr := intArray(1,5,2,3,6,7,4,8,9,10);
128     names := namesArray('Alice', 'Bob', 'Cindy', 'Sam', 'Eric');
129     dbms_output.put_line('Integer array');
130     --dbms_output.put_line('arr[0]' || arr(0));
131     dbms_output.put_line('arr[1]' || arr(1));
132     dbms_output.put_line('arr[2]' || arr(2));
133     dbms_output.put_line('arr[3]' || arr(3));
134 END;
135
136 /* Use of loops */
137 DECLARE
138     type intArray IS VARRAY(10) OF INTEGER;

```

```

139     type namesArray IS VARRAY(5) OF VARCHAR2(5);
140
141     arr intArray;
142     names namesArray;
143     i integer;
144
145 BEGIN
146     arr := intArray(1,5,2,3,6,7,4,8,9,10);
147     names := namesArray('Alice', 'Bob', 'Cindy', 'Sam', 'Eric');
148
149     i := 1;
150     /*While loop...! */
151     while( i <= 10) loop
152         dbms_output.put_line('arr[' || i ||'] =' ||arr(i));
153         i := i+1;
154     end loop;
155
156     /*For loop...! */
157     for i in 1 .. 10 loop
158         dbms_output.put_line('arr[' || i ||'] =' ||arr(i));
159     end loop;
160
161     i := 1;
162     while( i <= 5) loop
163         dbms_output.put_line('names[' || i ||'] =' ||names(i));
164         i := i+1;
165     end loop;
166
167     /*For loop...! */
168     for i in 1 .. 5 loop
169         dbms_output.put_line('names[' || i ||'] =' ||names(i));
170     end loop;
171 END;
172
173
174 /* Use of function in pl/sql */
175 /* Program - findMax value*/
176 DECLARE
177     a int;
178     b int;
179     c number;
180 FUNCTION findMax(x int, y int)
181 RETURN int
182 IS
183     z int;
184 BEGIN
185     IF x > y THEN
186         z:= x;
187     ELSE
188         Z:= y;
189     END IF;
190     RETURN z;
191 END;
192
193 BEGIN
194     a := &a;
195     b := &b;
196
197     c := findMax(a, b);
198     dbms_output.put_line(' Maximum of (' || a ||',' ||b || '): ' || c);
199 END;
200
201
202 /* Sorting algorithm...! */
203 DECLARE
204     type intArray IS VARRAY(10) OF INTEGER;
205     arr1 intArray;
206     i int;
207     j int;

```

```

208     c int;
209     temp INTEGER := 0;
210 FUNCTION sortArray(arr IN OUT intArray, len integer)
211 RETURN int
212 IS
213     ret int;
214 BEGIN
215     ret := 1;
216     dbms_output.new_line;
217     for i in 1 .. len loop
218         dbms_output.put_line('arr[' || i ||'] =' ||arr(i));
219     end loop;
220
221     for i in 1 .. len loop
222         for j in 1+i .. len loop
223             if (arr(i) > arr(j)) then
224                 temp := arr(i);
225                 arr(i) := arr(j);
226                 arr(j) := temp;
227             end if;
228         end loop;
229     end loop;
230
231     for i in 1 .. len loop
232         dbms_output.put_line('arr[' || i ||'] =' ||arr(i));
233     end loop;
234
235     return ret;
236 END;
237
238 BEGIN
239     arr1 := intArray(1,5,2,3,6,7,4,8,9,10);
240     c := sortArray(arr1, 10);
241 END;
242
243 /* SQL Operations */
244 /* ROLLNO NAME MARK1 MARK2 MARK3
245 -----
246 1 aparna 80 90 78
247 2 amritha 90 92 81
248 3 binuja 23 18 20
249 4 cathy 49 50 50
250 5 danish 60 62 61
251 6 fayas 76 62 74 */
252 create table Stud(rollno int primary key,
253                  name char(10),
254                  mark1 float,
255                  mark2 float,
256                  mark3 float);
257 insert into Stud values(1,'aparna', 80, 90, 78);
258 insert into Stud values(2,'amritha', 90, 92, 81);
259 insert into Stud values(3,'binuja', 23, 18, 20);
260 insert into Stud values(4,'cathy', 49, 50, 50);
261 insert into Stud values(5,'danish', 60, 62, 61);
262 insert into Stud values(6,'fayas', 76, 62, 74);
263
264 select * from Stud;
265
266 /*
267 Write a PL/SQL program to grade the student according to the following rules
268 Student(name,rollno,mark1,mark2,mark3)
269 TOTAL MARKS GRADE
270     >=250 Distinction
271     180-250 First Class
272     120-179 Second Class
273     80-119 Third Class
274     <80 Fail
275 The result should be in the following Format
276 STUDENT NAME:

```

```

277     ROLL NO :
278     TOTAL MARKS :
279     GRADE :
280 */
281 DECLARE
282     roleNo integer;
283     No integer;
284     Name varchar(20);
285     total integer;
286 BEGIN
287     -- get role no from user
288     roleNo := &roleNo;
289
290     --execute sql query
291     select rollno,name,(mark1+mark2+mark3) into No, Name, total from Stud where
        rollno=roleNo;
292
293     --print output
294     dbms_output.put_line('-----');
295     dbms_output.put_line('Roll no : ' || No);
296     dbms_output.put_line('Name : ' || Name);
297     dbms_output.put_line('Total Marks : ' || total);
298     dbms_output.put_line('-----');
299
300     --calculate & print grade
301     if (total >=250 ) then
302         dbms_output.put_line('Grade = Distinction');
303     elsif (total < 250 and total >= 180) then
304         dbms_output.put_line('Grade = First Class');
305     elsif (total < 180 and total >= 120) then
306         dbms_output.put_line('Grade = Second Class');
307     elsif (total < 120 and total >= 80) then
308         dbms_output.put_line('Grade = Third Class');
309     else
310         dbms_output.put_line('Grade = FAIL');
311     end if;
312 END;
313 /

```