

```

1  /*
2  drop procedure if exists pro1;
3  delimiter $
4  create procedure pro1()
5  BEGIN
6      select "Hello World" as "Message Box";
7  end $
8  delimiter ;
9
10
11 drop procedure if exists pro3;
12 delimiter $
13 create procedure pro3(x int,y int)
14 BEGIN
15 declare z1 int;
16 declare z2 int ;
17 set z1:=x+y;
18 set z2:=x-y;
19 select z1 as "z1";
20 select z2 as "z2";
21
22 end $
23 delimiter ;
24
25
26
27 drop procedure if exists pro5;
28 delimiter $
29 create PROCEDURE pro5(x int)
30 begin
31     if x>10
32     THEN
33         select "Good" as "marks";
34     else
35         select "bad" as "marks";
36     end if;
37 end $
38 delimiter ;
39
40
41
42 drop procedure if exists pro6;
43 delimiter $
44 create procedure pro6()
45 begin
46
47     declare x int;
48     set x:=0;
49     lb11:LOOP
50         set x:=x+1;
51         select x;
52         if x>10 THEN
53             leave lb11;
54         end if;
55     end loop lb11;
56 end $
57 delimiter ;
58
59
60
61 drop procedure if exists pro7;
62 delimiter $
63 create procedure pro7()
64 begin
65     declare x int;
66     set x:=0;
67     lb11:loop
68
69     set x:=x+1;

```

```

70      select x;
71      if x>20 THEN
72      leave lb11;
73      end if;
74      end loop lb11;
75      end $
76      delimiter ;
77
78
79
80 drop procedure if exists pro8;
81 delimiter $
82 create procedure pro8()
83 begin
84     INSERT INTO proc1 VALUES (2,"MK","REDDY");
85 END $
86 DELIMITER ;
87
88
89
90
91
92
93 drop procedure if exists pro9;
94 delimiter $
95 create procedure pro9()
96 begin
97     declare exit handler for 1062 select "Data already present" as "error window";
98     INSERT INTO proc1 VALUES (2,"MK","REDDY");
99 end $
100 delimiter ;
101
102
103
104
105
106
107 drop procedure if exists pro9;
108 delimiter $
109 create procedure pro9()
110 begin
111     declare exit handler for 1062 select "Data already present" as "error window";
112     INSERT INTO proc1 VALUES (3,"MK","REDDY");
113     select "data successfully entered";
114 end $
115 delimiter ;
116
117
118
119
120 drop procedure if exists pro10;
121 delimiter $
122 create procedure pro10(p1 int,p2 varchar(20),p3 varchar(30))
123 begin
124     declare exit handler for 1062 select "Data already present";
125     if p3="Pune"
126     THEN
127     insert into proc2 values (p1,p2,p3);
128     select "Data Successfully inserted";
129     else
130     select "invalid location";
131     end if;
132     end $
133     delimiter ;
134
135
136 drop procedure if exists pro11;
137 delimiter $
138 create procedure pro11(_deptno int )

```

```

139 begin
140     declare flag bool;
141     select true into flag from dept where deptno=_deptno;
142
143     if flag then
144         select * from dept where deptno=_deptno;
145     else
146         select "Record not found";
147     end if;
148 end $
149 delimiter ;
150
151
152
153 drop procedure if exists pro12;
154 delimiter $
155 create procedure pro12(_deptno int,_empno int,ename varchar(30),_city varchar(30))
156 BEGIN
157
158 declare flag bool ;
159     select true into flag from dept where _deptno=deptno;
160     if flag then
161         insert into emp values(_empno,_ename,_city,_deptno);
162         select "record inserted successfully";
163     else
164         select "deptno not found into department table";
165     end if;
166 end $
167 delimiter ;
168
169
170
171
172 drop procedure if exists pro9;
173 delimiter $
174 create procedure pro9(_tname varchar(20))
175 begin
176     set @x:=concat("select* from ",_tname);
177     prepare z from @x;
178     execute z;
179     end $
180     delimiter ;
181
182
183
184
185
186 drop procedure if exists pro10;
187 delimiter $
188 create procedure pro10(_tname varchar(20))
189 begin
190     set @x=concat("select * from ",_tname);
191     prepare z from @x;
192     execute z;
193 end $
194 delimiter ;
195
196
197
198
199
200 drop procedure if exists pro11;
201 delimiter $
202 create procedure pro11(_coname varchar(20),_tablename varchar(20))
203 begin
204     set @x = concat('select ', _coname, ' from ', _tablename);
205     prepare z from @x;
206     execute z;
207     end $

```

```

208         delimiter ;
209
210     /*
211     DROP PROCEDURE IF EXISTS pro11;
212     DELIMITER $
213
214     CREATE PROCEDURE pro11(_coname VARCHAR(20), _tablename VARCHAR(20))
215     BEGIN
216         SET @x = CONCAT('SELECT ', _coname, ' FROM ', _tablename);
217         PREPARE stmt FROM @x;
218         EXECUTE stmt;
219         DEALLOCATE PREPARE stmt;
220     END $
221
222     DELIMITER ;
223
224
225
226
227
228
229     drop procedure if exists pro11;
230     delimiter $
231     create procedure pro11(_cname varchar(200),_tname varchar(200))
232     begin
233         set @x=concat("select ", _cname ," from ", _tname );
234         prepare z from @x;
235         execute z;
236     end $
237     delimiter ;
238
239
240
241
242
243     _____FUNCTION_____
244
245
246
247
248
249     DROP FUNCTION IF EXISTS f1;
250     DELIMITER $
251
252     CREATE FUNCTION f1() RETURNS INT
253     DETERMINISTIC
254     BEGIN
255         RETURN 10;
256     END $
257
258     DELIMITER ;
259
260
261     _____Procedure 1) asignment_____
262
263
264
265     drop procedure if exists adduser;
266     delimiter $
267     create procedure adduser(username varchar(20),password varchar(20),email varchar(20))
268     begin
269         insert into login(username,password,email) VALUES (username, password, email);
270
271     end $
272     delimiter ;
273
274
275
276

```

```

277
278      _____Procedure 2) assignment_____
279
280
281
282
283
284
285
286
287
288 DROP PROCEDURE IF EXISTS checkuser;
289 DELIMITER $
290
291 CREATE PROCEDURE checkuser(_emailid VARCHAR(30))
292 BEGIN
293     DECLARE p BOOLEAN;
294
295     SELECT TRUE INTO p
296     WHERE _emailid = (SELECT emailid FROM login);
297
298     IF p THEN
299         SELECT username, password FROM login1 WHERE emailid = _emailid;
300     ELSE
301         INSERT INTO login1 (curr_date, curr_time, msg) VALUES (CURRENT_DATE(),
302             CURRENT_TIME(), 'Default');
303     END IF;
304 END $
305
306 DELIMITER ;
307
308
309
310
311
312
313 drop procedure if exists pro11;
314 delimiter $
315 create procedure pro11(id int)
316 begin
317     set id:=5;
318     select id;
319 end $
320 delimiter ;
321
322
323
324 drop procedure if exists pro13;
325 delimiter $
326 create procedure pro13()
327 begin
328     declare exit handler for 1050 select "Table already exists";
329     create table c1(id int primary key auto_increment,name varchar(20));
330 end $
331 delimiter
332
333
334
335
336 drop procedure if exists pro14;
337 delimiter $
338 create procedure pro14()
339 begin
340     declare exit handler for 1050 select "Table already present";
341     create table c2(id int,ename varchar(20));
342 end $
343 delimiter ;
344

```

```

345
346
347
348 drop procedure if exists pro15;
349 delimiter $
350 create procedure pro15(_cname varchar(20),_tname varchar(20))
351 begin
352     set @x := concat("select  ", _cname  ," from  ",_tname ) ;
353     prepare z from @x;
354     execute z;
355 end $
356 delimiter ;
357
358 drop procedure if exists pro16;
359 delimiter $
360 create procedure  pro16(tname varchar (20))
361 begin
362     set @x := concat("select * from  ",tname);
363     prepare z from @x;
364     execute z;
365 end $
366 delimiter ;
367
368
369 drop procedure if exists pro16;
370 delimiter $
371 create procedure pro16(p1 int,p2 int ,p3 int)
372 begin
373
374
375     set @x1 :=p1;
376     set @x2 :=p2;
377     set @x3 :=p3;
378     set @y := "insert into table1(id,eid,did) values(?,?,?)";
379 prepare z from @y;
380 execute z using @x1,@x2,@x3;
381
382 end $
383 delimiter ;
384
385
386
387
388 /*
389 DROP PROCEDURE IF EXISTS pro16;
390 DELIMITER $
391
392 CREATE PROCEDURE pro16(p1 INT, p2 INT, p3 INT)
393 BEGIN
394     SET @x1 := p1;
395     SET @x2 := p2;
396     SET @x3 := p3;
397     SET @y := "INSERT INTO table1 (id, eid, did) VALUES (?, ?, ?)";
398
399     PREPARE z FROM @y;
400     EXECUTE z USING @x1, @x2, @x3;
401
402     DEALLOCATE PREPARE z;
403 END $
404
405 DELIMITER ;
406
407
408
409
410 drop procedure if exists pro1;
411 delimiter $
412 create procedure pro1()
413 begin

```

```

414         declare z int ;
415         select max(sal) into z from emp;
416     end $
417     delimiter ;
418
419 drop function if exists f1;
420 delimiter $
421 create function f1() returns int
422 DETERMINISTIC
423 begin
424     call prol()
425     return 10;
426 end $
427 delimiter ;
428
429 DROP PROCEDURE IF EXISTS prol;
430 DELIMITER $
431
432 CREATE PROCEDURE prol()
433 BEGIN
434     DECLARE z INT;
435     SELECT MAX(sal) INTO z FROM emp;
436 END $
437
438 DELIMITER ;
439
440 DROP FUNCTION IF EXISTS f1;
441 DELIMITER $
442
443 CREATE FUNCTION f1() RETURNS INT
444 DETERMINISTIC
445 BEGIN
446     CALL prol();
447     RETURN 10;
448 END $
449
450 DELIMITER ;
451
452
453
454
455 drop procedure if exists prol;
456 delimiter $
457 create procedure prol()
458 begin
459     insert into d values (2,2,2);
460 end $
461 delimiter ;
462
463
464
465 drop procedure if exists prol;
466 delimiter $
467 create procedure prol()
468 begin
469     declare exit handler for 1062 select "Data already present " as "error window";
470     insert into d values (2,2,2);
471 end $
472 delimiter ;
473
474 drop procedure if exists prol;
475 delimiter $
476 create procedure prol()
477 begin
478
479     insert into d values(3,7,4);
480 end $
481 delimiter ;
482

```

```

483
484
485
486
487 drop procedure if exists pro1;
488 delimiter $
489 create procedure pro1()
490 begin
491
492 declare exit handler for 1062 select "Data already present " as "error window";
493 insert into d values(3,7,4);
494 end $
495 delimiter ;
496
497
498
499 drop procedure if exists pro1;
500 delimiter $
501 create procedure pro1(p1 int,p2 int,p3 int)
502 begin
503 declare exit handler for 1062 select "Data already present" as "Error Window";
504 insert into d values(p1,p2,p3);
505 end $
506 delimiter ;
507
508
509 drop procedure if exists pro1;
510 delimiter $
511 create procedure pro1()
512 begin
513
514 declare exit handler for 1050 select "table already prsent" ;
515 create table d(a int,b int,c int);
516
517
518 end $
519 delimiter ;
520
521
522 drop procedure if exists pro2;
523 delimiter $
524 create procedure pro2(_cname varchar(20),_tname varchar(20))
525 begin
526
527
528 declare exit handler for 1054 select "invalid column";
529 set @x:=CONCAT("SELECT ", _cname, " FROM ", _tname);
530 prepare z from @x;
531 execute z;
532 end $
533 delimiter ;
534
535 */
536
537
538
539
540
541

```