So far, you have worked on MySQL for simple and complex queries. Now that you are familiar with the MySQL syntax attempt the following questions using MYSQL.

**Task:**

1. **Write PL/SQL code with an exception handler:-**

**Answer:-**

**query:-**

DECLARE

pppno PASSENGER.PPNO%type := 'PPNO00009';

pname PASSENGER.NAME%type;

BEGIN

SELECT name INTO pname

FROM passenger

WHERE ppno = pppno;

DBMS\_OUTPUT.PUT\_LINE ('Name: '|| pname);

EXCEPTION

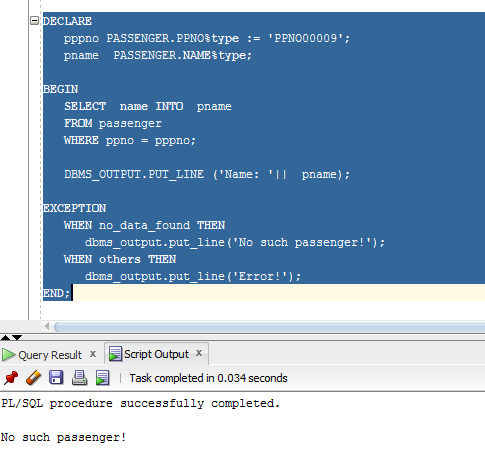
WHEN no\_data\_found THEN

dbms\_output.put\_line('No such passenger!');

WHEN others THEN

dbms\_output.put\_line('Error!');

END;



1. **List any few exception handlers with the corresponding comments when the exception will be raised?**

**Answer:-**

|  |  |  |
| --- | --- | --- |
| Sr.no | Exception | Raised |
|  | ZERO\_DIVIDE | Raised when any number is divided by zero |
|  | NO\_DATA\_FOUND | Raised when there is no record found |
|  | TOO\_MANY\_ROWS | Raised when more than one row is returned |
|  | VALUE\_ERROR | Raised when arithmetic, conversion, truncation error occurs |
|  | ROWTYPE\_MISMATCH | Raised when occurs host cursor variable and PL/SQL cursor variable involved in an assignment have incompatible return types |
|  | INVALID\_CURSOR | Raised when  a program attempts a cursor operation that is not allowed, such as closing an unopened cursor. |

1. **Create a user-defined exception handler.**

**Answer:-**

DECLARE

myexception EXCEPTION;

i NUMBER;

BEGIN

FOR i IN (SELECT \* FROM BUS)

LOOP

IF i.bus\_no = 103 THEN

RAISE myexception;

END IF;

END LOOP;

EXCEPTION

WHEN myexception THEN

dbms\_output.put\_line('Bus number already exists is bus table.');

END;

