CASE STUDY

Consider the below schema for the given case study:

Programmer

Name	Varchar2(8)	
Emp_id	varchar2(4)	Primary key
DOB	Date	Not Null
DOJ	Date	Not Null
Sex	Varchar2(1)	
Prof1	Varchar2(8)	
Prof2	Varchar2(8)	
Salary	Number(5)	Not Null

Package

Emp_id	Varchar2(4)	F.K
Title	Varchar2(20)	Not Null
Dev_In	Varchar2(8)	Not Null
SCost	Number(8,2)	
DCost	Number(6)	
Sold	Number(3)	
available	number(3)	

Studies

Emp_id	Varchar2(4)	F.K
Institute	Varchar2(10)	Not Null
Course	Varchar2(5)	Not Null
start_date	Date	

Course

Institute	Varchar2(10)	Not Null
Course	Varchar2(5)	Not Null
duration_in_months	Number(3)	Not Null
ccost	Number(6)	
Launch_date	Date	
status	Char(1)	"**V**" for valid and "**I**" for Invalid

Create a package pkg_developer which has following components:

- Public procedure Add_Studies(Empid,Institute,course,status,start_dt) which carries out the functionality defined in (A)

- Public procedure Add_Developer which carries out the functionality as defined in (B)

- Public Procedure Show_royalty as defined in (D)

- Private function Calculate_royalty as defined in (C)

(A)
Create a public procedure Add_Studies as a part of package pkg_developer that accepts all the column values for the "studies" table as user inputs and validates the following :

- The Developer exists (emp_id value is there in the "developer" table)
- Institute and course combination exists (is there in the "course" table)
- The Course status is valid. (status is "V" in "course" table)
- Start date is less than system date, but not lesser than 2 months from system date.

If all the validations are satisfied then the block should insert a record in the studies table and display a message on the screen "1 Record added". The block should raise an exception and handle it by displaying an appropriate error message if a validation is not satisfied.

Note: An employee can register only if he is a developer. This restricted is implemented by trigger TR_VALIDATE_ENROLLMENT as defined in (E)

(B)
Create a public procedure Add_developer as a part of package pkg_developer PL/SQL block that accepts all the column values except for the emp_id column, for the "developer" table, as user inputs in the outer block. The outer block also declares two exceptions "Invalid_DOJ" and "Check_DOB". The inner block performs the following validations :

- The emp_id is autogenerated by making use of a sequence
- DOJ should be greater than '12-jan-1998' otherwise the user defined exception "Invalid_DOJ" is raised
- DOB should be greater than DOJ by 20 years otherwise the user defined exception "Check_DOB" is raised
- If both the validations are satisfied then the inner block should insert a record in the "developer" table and display a message "1 Record added'".
- The outer block should handle the exceptions fired by the inner block by displaying the appropriate messages

(C)
Create a private function named CALCULATE_ROYALTY(p_emp_no) as a part of package pkg_developer

- The function accepts the employee number of a developer
- Calculates and returns the royalty that the developer has got so far, for the software he/she has developed
- Royalty is 10% of software cost, for each copy of the software sold
- The function should return -1 if the developer does not exist. It should return -2 if the developer has not developed any software or the software developed by him/her have not been sold so far.

(D)
Create a procedure named SHOW_ROYALTY that

- Takes employee number of a developer as a parameter
- Gives a call to the function CALCULATE_ROYALTY by passing on the employee number value
- The procedure should display on the screen the royalty value returned by the function.
- The procedure should display the appropriate messages when the developer does not exist or when the developer has not developed any software or the software developed by him/her have not been sold so far.

(E)
Create type course_details of object type to store the details of the course

(F)
Create a nested table **course_type** to store the course_details.

Create a table **Course_tab** with the following attributes.

Course_id	number	Primary key
Course_det	Course_type	

(G)
Create a stored procedure **Disp_course_Det** which displays the details of all the courses which are valid.
The procedure should also display the count of employees enrolled for each course till date.

(H)
Create a procedure "**insrt_details**" which will take course id, Institute name, Course, duration in months, course cost, and status as parameter.
The procedure should do the following validations:

1. Course id must be unique.
2. Status must be "V" or "I". If not then raise the exception with message "**Invalid status**".
3. Launch date must be present date.
4. Insert record into Course_tab and then display all the records from the table.

(I)
Create a procedure "**updt_details**" which will take course id, course cost as parameter.
The procedure should update the cost of the particular course and then display all the details of the particular course id.