INFORMATICS PRACTICES

Time allowed : 3 hours ***Maximum marks: 70***

***General Instructions**:*

(i) This question paper is divided into 3 sections.

*(ii) **Section A** consists of 30 marks.*

*(iii) **Section B** and **Section C** are of 20 marks each.*

(iv) Answer the questions after carefully reading the text.

QUESTION PAPER CODE 90

SECTION A

1. Answer the following questions :

(a) Expand the following abbreviations and explain in brief : 2

(i) FLOSS

(ii) GNU

(b) Write one point of difference between a freeware and free software. 2

(c) Write one example to show how Data mining is helpful in decision making. 2

(d) How is Post implementation review important during System Development Life Cycle ? 2

(e) Identify the type of relationship represented in the following statement and draw an Entity Relationship Diagram to show it : 2

"A customer can buy many items."

2. Answer the following questions :

(a) Write one difference and one common characteristic between a control and a variable in VB. 2

(b) Why do we write comments in a program ? What are the two ways of writing comments in VB ? 2

(c) Write the purpose of Select Case statement with the help of an example. Which VB statement can be used in place of Select Case statement ? In the Select Case statement, what happens if every case fails and there is no Case Else option ? 3

(d) What is the difference between a function and a sub procedure ? Write one example of each. 3

3. Answer the following questions :

(a) In a PL/SQL code, DBMS_OUTPUT . PVT_LINE is used to display output but the output is not getting displayed on screen. What might have gone wrong ? 2

(b) Write one point of difference between Row level Triggers, and Statement level Triggers. 2

(c) Explain in brief the purpose of Savepoint statement with the help of PL/SQL code example. 2

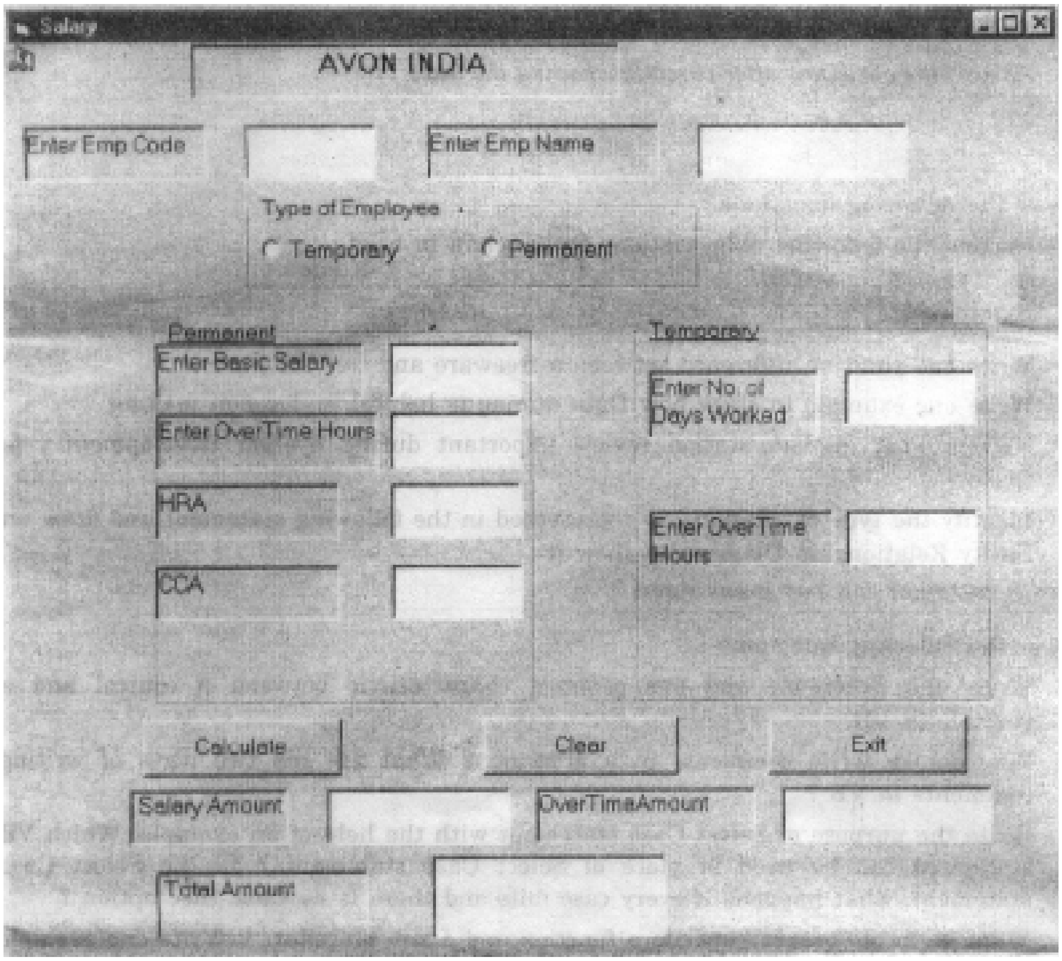
(d) What is a view ? Why does a view not require any physical storage ? 2

(e) Why are named procedures called stored procedures ? Write one advantage of stored procedures. 2

SECTION B

4. Read the following case study and answer the questions that follow :

AVON INDIA has computerized its Payroll. The following is the Data Entry screen used by them :



The form details for the above form are given in the following table :

Object Type	**Object Name**	**Description**
Form	FrmSalary	The main Form Object.
Text box	txtEmpCode	To enter Employee Code.
	txtName	To enter name of employee.
	txtBasic	To enter Basic Salary of Permanent Employee.
	TxtOverPerm	To enter overtime hours worked by Permanent Employee.
	txtHRA	To display House Rent Allowance.
	txtCCA	To display City Compensatory Allowance.
	TxtNumDays	To enter number of days worked by Temporary Employee.
	TxtOverTemp	To enter number of hours worked by Temporary Employee.
	TxtSalary	To display Salary earned by Temporary or Permanent employee.
	TxtOverTime	To display OverTime amount earned by Temporary or Permanent employee.
	TxtTotal	To display Total amount to be paid to Temporary or Permanent Employee.
Option Button	OptTemp OptPerm	To provide information whether Employee is of Temporary Category or Permanent Category.
Frame	FraTemp	To group controls related to Temporary Employee.
	FraPerm	To group controls related to Permanent Employee.
Command Buttons	cmdCalculate	To calculate HRA, CCA, OverTime Amount, Salary Amount and Total Amount in. case of Permanent Employee. To calculate OverTime Amount, Salary Amount and Total Amount in case of Temporary Employee.
	cmdClear	To clear all the values in Text boxes, Option button in the form.
	cmdExit	To close the application.

Write code to implement the following :

(a) If option button for Permanent Employee is chosen then Controls in the Frame related to Temporary Employee should be disabled and if option button for Temporary Employee is chosen then Controls in the Frame related to Permanent Employee should be disabled. 2

(b) When the user clicks Clear button, all the values stored in text boxes and option button should be cleared. 2

(c) Check that in the text box for Employee code (txtEmpCode) only numeric data is entered. 2

(d) When the command button with caption "Calculate" (cmdCalculate) is clicked, HRA, CCA, OverTime Amount, Salary Amount and Total Amount should be calculated in case of Permanent Employee and OverTime Amount, Salary Amount and Total Amount should be calculated in case of Temporary Employee. 4

The criterion for calculation is as given below :

For Temporary Employee	Salary is Rs. 250 per day.
	OverTime is Rs. 50 per overtime hour
For Permanent Employee	HRA is 10% of Basic Salary
	CCA is Rs. 500
	OverTime allowance is Rs. 75 per overtime hour.

Total Amount is the sum total of Salary Amount (TxtSalary) and OverTime Amount (TxtOverTime).

5. Answer the following questions :

(a) Write the following code segment using For Loop : 2

```
I=6
x=I
Do While I>=1
     x=x-2
     If x=0 Then
          Print "Zero"
     End If
     Print I
     I=I-2
Loop
```

(b) Write the output that the following code segment will generate : 2

```
x=”Class”
Print Mid (LTrim( “Computer Science” ),1,4)+”One”
Print(7*4>8+3) And (3^2 < 6/2)
Print Val(“l6 oranges”) + Len(x)
Print InStr(“Computers”, “ut”)
```

(c) Rewrite the following code segment using Select Case statement : 2

```
If ch = “A” Then
   countA = countA + 1
ElseIf ch = “B” Then
   countB = countB + 1
ElseIf ch = “C”  Then
   countC = countC + 1
Else
   countE = countE + 1
End If
```

(d) Write a procedure in VB that takes in year of birth of a person as an argument. It displays a message box indicating how many years are left for the person to reach the retirement age of 60. If the user is older than 65, wish him/her for a long life ! 4

SECTION C

6. Read the questions given below and answer accordingly :

(a) Rewrite the following PL/SQL loop so that you do not use a loop at all. 2

```
FOR i IN 1. . 2
LOOP
   IF i = l THEN
      calc_bonus(president_id, 2000000);
   ELSIF i=2 THEN
      calc_bonus(ceo_id, 5000000);
   END IF;
END LOOP;
```

(b) Write the output produced by the following part of code in PL/SQL. 2

```
declare
   i number;
   q number;
   p number;
begin
   p:=-2;
   q:=5;
   for i in 5. . 7 loop
      q:=q+2;
      p:=q;
      if p>9 then
           dbms_output.put_line(p);
      end if;
      dbms_output.put_line(i);
   end loop;
end;
```

(c) Find error(s) in the following PL/SQL code and rewrite the correct code after underlining the corrections made. 2

```
Declare
P Number;
Pr Number:=0;
   FOR X IN 3. . 6 LOOP
      IF MOD(X,3):=0 THEN
            PR:=PR*X;
         ELSE
            PR:=PR-X;
      DBMS_OUTPUT.PUT_LINE(TO_CHAR(PR));
   END OF LOOP;
END
```

(d) Write a PL/SQL procedure that takes employee code of an employee as a parameter. In the table "Employee", if the commission field (comm) of that employee is empty then set it to 1000. (Field name of employee code : Empno, Commission : Comm) 4

7. Answer the questions based on the table **SalesPerson** given below :

Table : SalesPerson

Column Name	Data Type	Size	Description	Constraint
SNum	NUMBER	6	Salesperson's Identification number	PRIMARY KEY
SFName	VARCHAR2	25	First Name of SalesPerson	NOT NULL
SLName	VARCHAR2	25	Last Name of SalesPerson	NOT NULL
City	VARCHAR2	10	City where SalesPerson works	
Sales	NUMBER	9,2	Sales achieved by SalesPerson	
Comm	NUMBER	8,2	Commission earned by Sales Person	

(a) Write the SQL command to Create the above table with constraints. 2

(b) Write SQL command to Create a view consisting of all the Salespersons working in Delhi city. 2

(c) Write an SQL command to display each city along with total sales in that city. 2

(d) Create a Trigger to change the commission amount to 5000 every time the commission amount entered by user exceeds 5000. An appropriate message should also be displayed. 4

QUESTION PAPER CODE 90/1

SECTION A

1. Answer the following questions :

(a) Define (i) Freeware (ii) Shareware 2

(b) Write one point of difference between front end and back end with one example of each. 2

(c) Write one example each of "one to many" and "many to many" relationships between two entity sets. 2

(d) What is the purpose of Data Mining? 2

(e) During System Development Life Cycle, in which phase is the scope and true nature of problem determined? Explain in brief with the help of an example. 2

2. Answer the following questions :

(a) How many value(s) does a Procedure and a Function return ? 2

(b) Distinguish between Single Document Interface and Multiple Document Interface. Write one example each of SDI and MDI type application. 2

(c) What is Selection statement ? Name any two selection statements that VB provides. 2

(d) What are Events ? What are Event Procedures ? How are they related ? 2

(e) What is a Bound Control? Name any two dataware properties of bound controls. 2

3. Answer the following questions :

(a) Which statement must be present in a simple loop (Loop.. Endloop) so that it does not become an infinite loop ? 1

(b) Suggest parameters mode (IN/OUT/IN OUT) for the following : 1

(i) Parameter can be used as a normal variable.

(ii) Passed value must not get changed.

(c) Write two points of differences between SQL and PL/SQL. 2

(d) How are triggers similar to Constraints ? Write one point. 2

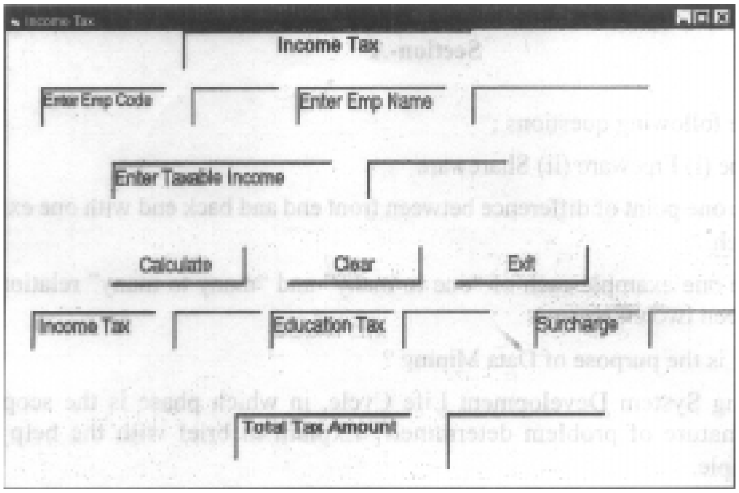
(e) What is the purpose of cursor in PL/SQL ? Name the types of cursors used in PL/SQL. 2

(f) Why do we use Roll back statement ? Explain in brief with the help of an example. 2

SECTION-B

4. Read the following case study and answer the questions that follow :

ABC Company has developed the following interface to enter and display data related to Income tax of employees.



The form details of the above form are given in the following table :

Object Type	**Object Name**	**Description**
Form	FrmSalary	The main Form Object
Text box	TxtEmpCode	To enter code of Employee.
	TxtName	To enter Name of Employee.
	Txtlncome	To enter Taxable Income of Employee.
	TxtlTax	To display Income tax
	TxtEdTax	To display Educational tax
	TxtSurcharge	To display Surcharge
	TxtTotalTax	To display Total Tax to be paid by the employee.
Command Buttons	cmdCalculate	To calculate Income tax, Education Tax, Surcharge and Total Tax
	cmdClear	To clear all the values in Text boxes
	cmdExit	To close the application.

Write code to implement the following :

(a) When the form loads text boxes for Income tax, Education Tax, Surcharge and Total Tax Amount should be disabled. They should be enabled only when Calculate command button is clicked. 2

(b) Taxable Income entered should be numeric data. 2

(c) When the user clicks the Clear command button, textboxes EmpCode and EmpName should be set to blank and other textboxes should be set to zero. 2

(d) When Calculate command button is clicked, Income tax, Education Tax, Surcharge and Total Tax (sum of Income Tax, Education Tax, Surcharge) is displayed in their respective text boxes based on the following criterion. 4

Taxable Income	**Income Tax**	**Education Tax**	**Surcharge**
UptoRs. 1,00,000	Nil	Nil	Nil
Rs. 1,00,001 to 1,50,000	10% of the amount exceeding Rs. 1,00,000	2% of Taxable Income	Nil
Rs. 1,50,001 to 2,50,000	Rs. 5,000+20% of amount exceeding Rs. 1,50,000	2% of Taxable Income	Nil
Rs. 2,50,001 and above	Rs. 25,000+30% of the amount exceeding Rs. 2,50,000	2% of Taxable Income	l% of Taxable Income.

5. Answer the following questions :

(a) How many times will the following loop execute ? 2

```
1 = 6
Do While I > = -1
I=I-2
Print I
Loop
```

(b) Write the output that the following code segment will generate 2

```
Stringl = “Class XII”
String2 = “XI”
Print Stringl + String2
Print InStr (Stringl, String2)
Print Mid (LCase(Stringl), 6, 3 )
Print Int(4.7) + Len(String2)
```

(c) Rewrite the following code using If Elself 2

```
Select Case Code
Case Is > 45
   Messagel = “Error”
CaselQTo20
   Message1 = “Accounts”
Case 21 To 30
   Message1 = “Personnel”
Case 31 To 45
   Message1 = “EDP”
Case Else
   Message1 = “Access Denied”
End Select
```

(d) Write a function that receives two integers as arguments and returns the following: 4

0 if both are equal

- 1 if first is greater than the second

1 if second is greater than the first

SECTION-C

6. Read the questions given below and answer accordingly :

(a) Write PL/SQL cursor declaration to store Employee code(EMPNO), Employee name(ENAME) of employees from Emp table who are located (LOC) in Delhi. 2

(b) Write the output produced by the following code in PL/SQL 2

```
declare
cnt number;
num number;
begin
cnt:= 8;
num := 4;
while cnt>4 loop
   num := num-2;
   cnt := cnt-1;
if cnt>4then
   dbms_output.put_line(cnt);
end if;
end loop;
dbms_output.put_line(num);
end;
```

(c) Find error(s) in the following PL/SQL code and rewrite the correct code after underlining the corrections made. 2

```
DECLARE
   CTR NUMBER;
SUM1 NUMBER;
BEGIN
   CTR : = 0;
   SUM1 : = 0;
LOOP
   SUM1 : = SUM1 + CTR;
   PRINT(TO_CHAR(SUM1));
   CTR : = CTR-2;
   IF CTR > 2 THEN
      EXIT;
END LOOP;
```

(d) Create a procedure called Select_employee that selects the first name, last name, join date, and salary for employee with employee code 100 from EMP table into variables and displays values stored in these variables.

Table Emp consists of fields with field names : ID,

FIRST_NAME, LAST_NAME, JOIN_DATE, SALARY 4

7. Answer questions based on the table **Student** given below :

Table : **Student**

Column Name	**Data** Type	Size	Description	**Constraint**
RollNumber	NUMBER	2	Student's Roll number	Primary key
Name	VARCHAR2	25	Name of Student	NOT NULL
Class	VARCHAR2	3	Class of Student	
Stream	VARCHAR2	15	Stream opted by the Student	
TotalMarks	NUMBER	8,2	Total marks scored by the student	
Grade	VARCHAR2	1	Grade scored by the student	Can be'A' or 'B' or 'C

(a) Write SQL command to display Roll numbers, names, Total marks and grades of all the students in "Nonmedical" stream sorted by Total marks in descending order. 2

(b) Write SQL command to create a view consisting of all students in "Medical" stream and who have scored "A" grade. 2

(c) Write PL/SQL Procedure that takes in Student's Roll number as a parameter and increases Total marks of that student by 2% (as he/she is a Sports person.) 3

(d) Create a Trigger to display the name and stream of Student(s) whose record(s) is/are being deleted from the Student table. 3

Marking Scheme — Informatics Practices

Important Note

- All answers provided in the marking scheme are SUGGESTIVE.
- Examiners are requested to accept all possible alternative correct answers.
- In Section C, semicolon to be ignored for terminating SQL statements.

QUESTION PAPER CODE 90

EXPECTED ANSWERS/VALUE POINTS

SECTION – A

1. (a) **(i)** **FLOSS** - Free/Libre/Open-Source Software refers to both Free Software and Open Source Software.

(ii) **GNU** - Gnu's Not UNIX is a UNIX-compatible operating system developed by the Free Software Foundation.

(½ Mark each for full form of each term)

(½ Mark each for correct definition of each term)

(b) **Freeware:** Software that is available free of charge for personal use. The author retains the copyright, which means that it cannot be altered or sold.

Examples of Freeware:

Name of any proprietary software developed by individual/organisation to be used by others and distributed free of cost.

Free software: Software that can be used, copied, studied, modified and redistributed.

Examples of Free Software:

GNU/Linux, OpenSolaris, Apache web server, MySQL and PostgreSQL, Panjo, Tomcat, Perl, PHP, Python, X Window System, GNOME, KDE, OpenOffice.org office suite, Mozilla and Firefox web browsers

(2 Marks for any one correct point of difference)

OR

(1 Mark each for individual definitions)

OR

(½ Mark each if only examples are given)

(c) Any example which covers any one of the points mentioned below:

- Establishes relationships between market parameters
- Identifies patterns in customer spending
- Predicts behavior in customers

- Solves business decision problems
- Predicts future trends
- Allows businesses to take pro-active knowledge–driven decisions

(2 Marks for any one correct example of Data mining)

OR

(2 Marks if any one of the above points is explained)

(d) A post implementation review helps in evaluating the system in terms of response time, ease of use and reliability.

OR

A post implementation review is used to document implementation experiences to recommend system enhancements and provide guidance for future projects.

OR

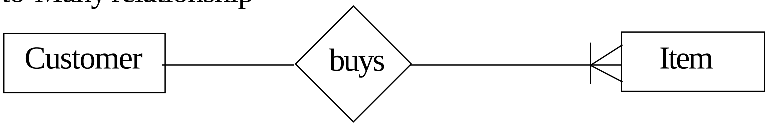
A post implementation review is an evaluation of a system in terms of the extent to which the system accomplishes stated objectives and actual project costs exceed initial estimates.

OR

A post implementation review helps to evaluate the monetary and non-monetary benefits achieved by the system.

(2 Marks for any equivalent explanation)

(e) One-to-Many relationship



(1 Mark for identifying relationship)

(1 Mark for ER diagram)

2. (a)

	Control	Variable
Difference	It is a graphical feature, which may be drawn on forms to allow use interaction and thereby enabling occurence of an event .eg. command buttons, option buttons, checkboxes etc.	It is a named memory location whose contents can be changed during the execution of a program
Similarity	Can hold a value at compile time, which may be modified at run time .	

OR

A variable can store only one value at a time while a control has a number of values in its properties.

Both a variable and control can hold value(s) at compile time, which can be modified during run time.

OR

A user can interact directly with a control by performing events on it whereas the same is not true with variables. Controls exist both in design and code whereas variables exist only in code.

Both controls and variables have to be loaded in the memory and are initialized with values.

(1 Mark for any equivalent difference)

(1 Mark for any equivalent similarity)

(b) We write comments in a program:

- To document the program
- To write non-executable statement(s)
- To improve program readability
- To help other people to understand the program code

The two ways of writing comments in VB are :

- ' (Single quote)
- Rem (short for remark)

(1 mark for writing any one reason for comments)

(1 mark any one way of writing comments)

(c) Select Case executes one of several groups of statements, depending on the value of an expression.

OR

Select case is used to handle multiple-choice conditions.

Example :

```
Select case x
   Case 1
      y = z
   Case 2
      y = z^2
   Case 3
      y = z^3
   Case else
      Msgbox " Try Again !!!"
End Select
```

If-Then-Elseif or **switch** statement can be used in place of Select-Case

If every case fails and if there is no Case Else option then control will exit from the Select Case construct and the statement after the End Select would be executed.

(1 mark for stating purpose of Select Case)

(1 mark for any correct example)

(½ mark for mentioning If-then-elseif or switch)

(½ mark for an *appropriate explanation*)

Note: If the example contains a Case Else option in it, ½ mark of *appropriate explanation* should be given.

(d) A function returns a value while a sub procedure does not return a value.

Example of a function:

```
Function SquareArea(Side As Double)
SquareArea = Side * Side
End Function
```

Example of a sub procedure:

```
Sub SquareArea(Side As Double)
Dim Area
Area = Side * Side
Print Area
End Sub
```

(1 Mark for correct difference)

(1 Mark each for any correct example of sub procedure and function)

Note: Full 3 marks to be given if the difference is explained with the help of valid example of each.

3. (a) Before using DBMS_OUTPUT.PUT_LINE we need to execute the following command:

SET SERVER OUTPUT ON

(2 Marks for correct answer)

OR

(Full 2 marks to be given if the student has identified the error "DBMS_OUTPUT.PVT_LINE")

(b)

Row level	Statement level
1) It is fired once for each row, affected by the triggering SQL DML statement (INSERT/ UPDATE/DELETE)	1) It is fired only once , either before or after the triggering SQL statement executes irrespective of the number of rows modified by it .
2) It can refer to any values in the rows affected by the statement.	2) It cannot refer to any values in the the rows affected by the statement .
3) They are used for generating/ checking the values being inserted/ updated or to enforce integrity of individual rows .	3) They are used for data replication or for generation of summary information .
4) The inclusion of FOR EACH ROW clause in the CREATE	4) The omission of FOR EACH ROW clause in the CREATE

(2 Marks for any one correct difference)

OR

(1 Mark <u>each</u> for individual definitions)

(c) The SAVEPOINT statement is used to identify a point in a transaction to which you can later ROLLBACK.

OR

It is used to give a name and mark a specific point in the processing of current transaction. This marker allows ROLLBACK upto that point. i.e. the RDBMS will undo any change(s) and release locks issued after that SAVEPOINT.

OR

SAVEPOINT establishes a mark, which allows one to perform partial ROLLBACK.

Example:

```
SQL Statement 1;
Savepoint S1;
SQL Statement2;
Rollback to S1;
SQL Statement3;
```

In the above example any changes made to the database in Statement2 will be undone while changes done in Statement 1 and Statement3 will be saved.

OR

Example:

```
INSERT INTO emp(empno,ename,job) VALUES (8000,'SHAH','ANALYST) ;
UPDATE EMP SET SAL = 9000 WHERE empno = 8000;
SAVEPOINT point1;
DELETE FROM EMP WHERE empno = 8000;
ROLLBACK to point1;
DBMS_OUTPUT.PUT_LINE('Partial rollback occured...');
```

(2 marks for explaining purpose of SAVEPOINT with the help of an example)

OR

(2 marks for an example explaining the purpose of SAVEPOINT)

OR

(1 mark only for explanation only without example)

OR

(1 mark only for example without any explanation)

(d) A **view** is a virtual or logical table composed of the result set of a query.

OR

A view is a virtual table that does not exist, but is a logical definition of a set of related columns, usually from multiple tables.

OR

A view presents data to the end user of an application system the way the end user is used to seeing the data.

Why does a view not require any physical storage?

A view definition is permanently stored as part of the database. Since no stored file is created to store the contents of a view , a view would not require any physical storage as only its definition is stored.

OR

A view does not require any physical storage as it is not part of the physical schema: it is a dynamic, virtual table computed or collated from data in the database.

(1 Mark for correct definition of View)

(1 Mark for correct reason)

(e) Named procedures are called stored procedures because they are compiled and stored as schema objects in the Oracle database.

Advantages of Stored Procedures

- Bigger programs can be broken into smaller, manageable units
- They are compiled at the time of their creation so they enhance the performance.
- Stored procedure enhance reusability
- With Stored procedure, a better database security is applied
- Procedure take advantage of shared memory resources

(1 Mark for correct reason)

(1 Mark for any one correct advantage)

Section – B

4. (a)

```
Private Sub OptPerm_Click()
TxtNumDays.Enabled = False    OR   TxtNumDays.locked = True
TxtOverTemp.Enabled = False        TxtOverTemp.locked = True
End Sub
Private Sub OptTemp_Click()
txtBasic.Enabled = False
TxtOverPerm.Enabled = False
txtHRA.Enabled = False
txtCCA.Enabled = False
End Sub
```

OR

```
Private Sub OptPerm_Click()
FraPerm.Enabled = True
FraTemp.Enabled = False
End Sub
Private Sub OptTemp_Click()
FraPerm.Enabled = False
Fratemp.Enabled = True
End Sub
```

OR

```
If OptTemp.Value = True Then
FraPerm.Enabled = False
Else
FraPerm.Enabled = True
End If
If OptPerm.Value = True Then
FraTemp.Enabled = False
Else
FraTemp.Enabled = True
End If
```

```
OR
If OptTemp.Value = True Then
TxtNumDays.Enabled = False    OR   TxtNumDays.locked = True
TxtOverTemp.Enabled = False        TxtOverTemp.locked = True
End If
If OptPerm.Value = True Then
txtBasic.Enabled = False
TxtOverPerm.Enabled = False
txtHRA.Enabled = False
txtCCA.Enabled = False
End Sub
```

(½ mark for checking for Temporary employees)

(½ mark for enabled/locked property)

(½ mark for checking for Permanent employees)

(½ mark for enabled/locked property)

Note:

- **0 and 1 instead of False and True to be considered correct**
- **Any additional statements in the code may be ignored**
- **Ignore Cases (Lower/Upper) for all identifiers in the above Codes**
- **Ignore use of VAL functions used in above code**

(b)

```
txtEmpCode.Text = 0
txtName.Text = “”
txtBasic.Text = “”
TxtOverPerm.Text = “”
txtHRA.Text = “”
txtCCA.Text = “”
TxtNumDays.Text = “”
TxtOverTemp.Text = “”
TxtSalary.Text = “”
TxtOverTime.Text = “”
TxtTotal.Text = “”
OptTemp.Value = False
OptPerm.Value = False
```

OR

```
txtEmpCode.Text = Empty
txtName.Text = Empty
txtBasic.Text = Empty
TxtOverPerm.Text = Empty
txtHRA.Text = Empty
txtCCA.Text = Empty
```

```
TxtNumDays.Text = Empty
TxtOverTemp.Text = Empty
TxtSalary.Text = Empty
TxtOverTime.Text = Empty
TxtTotal.Text = Empty
OptTemp.Value = False
OptPerm.Value = False
```

(½ Mark each for any four correct statements)

Note:

- **Ignore Cases (Lower/Upper) for all identifiers in the above Codes**

(c)
```
If Not(IsNumeric(txtEmpCode.Text)) Then
   Msgbox (“Please enter numbers only”)
End If
```

OR

```
If Not(Right(txtEmpCode.text,1) >= “0” And
   Right(txtEmpCode.text,1) <=”9”) then
txtEmpCode.text = “”    or msgbox(“Enter only numeric value”)
End if
```

OR

Any other equivalent code

(1 Mark for correct If statement)

(1 Mark for displaying error message or assigning txtEmpcode as blank)

Note:

- **Ignore Cases (Lower/Upper) for all identifiers in the above Codes**

(d)
```
If OptTemp.Value = True Then   OR   if OptTemp.value = 1 then
  TxtSalary.Text = 250 * Val(TxtNumDays)
  TxtOverTime.Text = 50 * Val(TxtOverTemp)
  TxtTotal = Val(TxtSalary.Text) + Val(TxtOverTime.Text)
End If
If OptPerm.Value = True Then
  txtHRA.Text = txtBasic * 10 / 100
  txtCCA.Text = 500
  TxtSalary.Text=Val(txtBasic.Text)+Val(txtHRA.Text)+
                                 Val(txtCCA.Text)
  TxtOvertime.Text = 75 * Val(TxtOverPerm)
  TxtTotal = Val(TxtSalary.Text) + Val(TxtOverTime.Text)
End If
```

OR

```
If  OptTemp.Value =True then
   TxtSalary=250* val(TxtNumDays.Text)
   TxtOvertime= val(TxtOverTemp.Text) *50
   TxtTotal=val(TxtSalary)+val(TxtOverTime)
else
   txtHRA=0.1*val(txtBasic)
   txtCCA=500
   TxtOverTime= val(TxtOverPrem) *75
   TxtTotal=val(txtHRA)+val(txtCCA)+val(TxtOverTime)
endif
```

OR

Any other equivalent code.

(1 mark for if statement(s) the correct option button)
(½ mark to calculate salary)
(½ mark to calculate the overtime pay)
(½ mark to calculate the hra)
(½ mark to calculate the CCA)
(1 mark to calculate the total pay)

Note:

- **Ignore Cases (Lower/Upper) for all identifiers in the above Codes**
- **Ignore use of VAL functions used in above code**

5. (a)

```
x = 6
For I = 6 To 1 Step -2
   x = x – 2
   If x = 0 Then
      Print "zero"
   End If
   Print I
Next
```

(½ Mark for initializing x)
(½ Mark each for (a) correct use of Initialisation of I in FOR loop (b) correct condition in FOR loop (c) decrementing the value of I in FOR loop)

(b) CompOne

False **OR** 0

21

5

(½ mark for each line of correct output)

Note:

Deduct ½ mark, if the correct values of output is shown in a single line

(c)
```
Select case ch
    Case “A”
        countA = countA + 1
    Case “B”
        countB = countB + 1
    Case “C”
        countC = countC + 1
    Case  Else
        countE = countE + 1
End Select
```

(½ mark for select statement)

(½ mark for any one correct case value)

(½ mark for case else statement)

(½ mark for end select statement)

(d)
```
Sub Check(Year_Birth As Integer)
    Dim Age, Retire
    Age = Year(Date()) – Year_Birth
    Retire = 60 – Age
    If Retire > 0 Then
        Msgbox “Years left for retirement:” + Retire
    End If
    If Age > 65 Then
        Msgbox “Wish you a long life!”
    End If
End Sub
```

OR

```
Private Sub checkage(birthyear As Integer)
    Dim age As Integer
    age = Year(Now) - birthyear
    If age > 65 Then
        MsgBox “Have a long Life !”
    End If
    If age <= 60 Then
        MsgBox “There are “& 60-age & “ Years left for Retirement”
    End If
End Sub
```

OR

```
Sub retire(bday As Date)
    myage = Abs(DateDiff(“yyyy”, bday, Now))
    If myage <= 60  Then
```

```
        Print “Years left for retirement : “ & 60 – myage
    End If
    If myage > 65 Then
        MsgBox (“Happy long life...”)
    End If
End Sub
```

OR

Any other equivalent code

(1 mark for Sub routine header)
(1 mark for calculating present age)
(1 mark for checking and displaying years left)
(1 mark for checking and displaying if older than 65 yrs)

Section - C

6. (a)

```
calc_bonus(president_id, 2000000);
calc_bonus(ceo_id, 5000000);
```

OR

```
I = 1 ;
If  I = 1 then
    calc_bonus(president_id,2000000);
End if ;
I := I + 1 ;
If  I = 2 then
    calc_bonus(ceo_id,5000000)  ;
End if ;
```

(2 Marks for calling calc_bonus(president_id, 2000000) and calc_bonus(ceo_id, 5000000) in the correct sequence)
OR
(1 Mark if I is assigned the correct values in the correct sequence)

(b) Output:

```
5
6
11
7
```

(½ Mark for each line of correct output)

Note:
Deduct ½ mark, if the correct values of **output is shown in a single line**

(c)

```
Declare
P Number;
Pr Number:=0;
BEGIN                                    —Error 1
   FOR X IN 3..6 LOOP
      IF MOD(X, 3)=0 THEN                —Error 2
          PR:=PR*X;
      ELSE
          PR:=PR-X;
      END IF;                            —Error 3
      DBMS_OUTPUT.PUT_LINE(TO_CHAR(PR));
   END LOOP;                             —Error 4
END;                                     —Error 5
```

(½ Mark each for identifying <u>and</u> correcting any four errors)

OR

(1 Mark to be given if all the errors are identified without correction)

(d)

```
CREATE OR REPLACE PROCEDURE ShowEmp(P_Eno NUMBER) AS
   V_eno Employee.EmpNo%TYPE ;
   V_comm Employee.Comm%TYPE ;
BEGIN
   SELECT EmpNo, Comm. INTO V_no , V_comm
   FROM Employee  WHERE Empno = P_Eno;
   IF V_Comm IS NULL THEN
      UPDATE Employee SET Comm = 1000
      WHERE EmpNo = P_Eno;
   END IF;
END ShowEmp;
```

OR

Any other equivalent code

(½ mark for procedure header)
(½ mark for using correct parameters)
(½ mark for declaration of local variables)
(1 mark for the SELECT query OR cursor)
(½ mark for IF statement)
(1 mark for the correct statement for updation)

7. (a)

```
Create table SalesPerson
(
   SNUM NUMBER(6) PRIMARY KEY ,
   SFNAME VARCHAR2(25)  NOT NULL ,
   SLNAME  VARCHAR2(10) NOT NULL ,
   CITY VARCHAR2(10) ,
   SALES NUMBER(9,2) ,
   COMM NUMBER(8,2)
);
```

OR

```
Create table SalesPerson
(
   SNUM NUMBER(6) CONSTRAINT PK PRIMARY KEY ,
   SFNAME VARCHAR2(25)  NOT NULL ,
   SLNAME  VARCHAR2(10) NOT NULL ,
   CITY VARCHAR2(10) ,
   SALES NUMBER(9,2) ,
   COMM NUMBER(8,2)
);
```

(½ mark for CREATE TABLE SalesPerson)
(½ mark for fields with data types)
(½ mark for PRIMARY KEY constraint)
(½ mark for NOT NULL constraint)

(b)

```
CREATE VIEW sales_view AS
SELECT * FROM SalesPerson
WHERE City= ‘Delhi’;
```

OR

```
CREATE VIEW Sales(SNnum,SFName,SLName,City,Sales,Comm ) AS
SELECT SNum,SFName,SLName,City,Sales,Comm
FROM SalesPerson
WHERE City = ‘Delhi’ ;
```

(1 Mark for Create View command)
(½ Mark for Select)
(½ Mark for Where clause)

(c) SELECT City, SUM(Sales)

FROM SalesPerson

GROUP BY City;

(½ mark for SELECT statement)

(½ mark for SUM function)

(½ mark for FROM)

(½ mark for GROUP BY clause)

(d)
```
CREATE OR REPLACE TRIGGER upd_comm
BEFORE INSERT OR UPDATE ON SalesPerson
FOR EACH ROW
  BEGIN
    IF :new.comm > 5000  THEN
      :new.comm := 5000;
      DBMS_OUTPUT.PUT_LINE('Commission cannot be more than Rs.5000');
    END IF;
END upd_comm;
```

OR

```
CREATE OR REPLACE TRIGGER upd_comm
BEFORE INSERT OR UPDATE OF COMM ON SalesPerson
FOR EACH ROW WHEN (new.comm > 50000)
BEGIN
  :new.comm := 5000;
  DBMS_OUTPUT.PUT_LINE('Commission cannot be more than Rs.5000');
END upd_comm;
```

OR

Any other equivalent code

(½ mark for create)

(1 mark for triggering event and time)

(½ mark for row level)

(1 mark for condition check)

(½ mark for assigning new value)

(½ displaying error message)

QUESTION PAPER CODE 90/1

EXPECTED ANSWERS/VALUE POINTS

SECTION – A

1. (a) **Freeware**: Software application programs those are free to use and distribute. The author retains the copyright, which means that it cannot be altered or sold.

Shareware: Copyrighted software that is available free of charge on a trial basis, which usually expires after a period of time.

(1 Mark each for correct definition of Freeware and Shareware)

(b) **Front-end:** An application that users directly interact with for data entry, querying etc. For example Visual Basic, Web Browser.

Back-end: A program or application which services the requests of the front-end. For example – Oracle, Web Server.

(½ Mark each for correct definition of Front-end and Back-end)

(½ Mark each for any correct example of Front-end and Back-end)

(c) One-to-Many relationship: Country → State

Many-to-Many relationship: Country → River

(1 Mark each for any correct relationship depicting one-to-many and many-to-many)

(d) The purpose of data mining is to:

- Solve business decision problems
- Identify patterns in customer spending
- Predict behavior in customers
- Establish relationships between market parameters
- Predict future trends
- Allows businesses to take pro-active knowledge–driven decisions

(2 Marks for stating any one correct purpose of data mining)

(e) **Analysis Phase / Preliminary Study / Initial Survey / First Phase**

This phase gathers the requirements for the system. This stage includes a detailed study of the business needs of the organization.

(1 Mark for identifying the phase)

(1 Mark for explaining with OR without an example)

2. (a) A procedure does not return any value.

A function returns only one value.

(1 Mark each for number of values returned by procedure and function)

(b) **MDI:** A function that allows an application to display and lets the user work with more than one document at the same time.

Example: MS Word OR MS Excel

SDI: A function that allows an application to display only one document at a time.

Example: Notepad OR MS-Paint

(½ Mark each for correct definition of MDI and SDI)

(½ Mark each for any correct example of MDI and SDI)

(c) Selection statements are used to perform different actions based on different conditions.

OR

Selection statement allows to choose a set of instructions for execution from the given alternatives depending on a condition.

Selection statements available in VB are:

- if statement
- if...then...else statement
- if...then...elseif statement
- select case statement

(1 Mark for correct definition of selection statement)

(½ Mark each for naming ANY TWO selection statements)

(d) An event is a message sent by an object announcing that something has happened.

OR

Event refers to the occurrence of an activity usually in response to the user’s action, such as a key press, mouse button click or a keystroke.

Event procedures are procedures that are called when a corresponding event occurs.

OR

An event procedure is executed when the event connected with it occurs.

(1 Mark each for correct definition of event and event procedure)

OR

(2 Marks for explanation of Event and Event Procedure with the help of any valid example)

(e) Bound Control is a control that can provide access to a specific column or columns in a Data source through a data control.

OR

A bound control is a control on a form or report that gets its contents from a field in the underlying table, query, or SQL statement.

Data aware properties of bound controls:

- Data Source
- Data Field
- Data Member
- Record Source

(1 Mark for correct definition of bound control)

(½ Mark each for naming ANY TWO data aware properties)

3. (a) EXIT

OR

EXIT WHEN

(1 mark for correct answer)

(b) (i) IN OUT

(ii) IN

(½ Mark each)

(c)

SQL	PL/SQL
Non-procedural language.	Procedural Language
4th generation language.	3rd generation language
Single line statements	Block of statements
Exception handling is not possible	Exception handling is possible
Time consuming due to Network traffic. Each SQL statement is sent to database server at a time.	Reduced network traffic as entire block of statements is sent for processing at a time

(1 mark for each for any two points of difference)

(d) Triggers are similar to constraints as they are used to preserve data integrity by consistently checking on or changing data.

OR

Triggers like constraints can be used for data validations or to control the content of database tables.

(2 Marks for any correct similarity)

(e) A cursor in PL/SQL is a mechanism that provides a way to select multiple rows of data from a database table and then process each row individually inside a PL/SQL block.

OR

A cursor is a name assigned to a specific private SQL area that contains information about the SQL statement and the set of data returned.

Types of cursors used in PL/SQL are:

- Implicit Cursors
- Explicit Cursors

(1 mark for any purpose or definition conveying similar meaning)

(½ mark each for two types of cursors)

(f) ROLLBACK statement is used to undo all changes made since the last COMMIT command.

Example:

```
SQL Statement 1;
Commit;
SQL Statement2;
SQL Statement3;
Rollback;
```

In the above example any changes made to the database Statement2 and Statement3 will be undone when Rollback statement is executed.

(1 mark for purpose)

(1 mark for any correct example)

SECTION – B

4. (a)

```
TxtITax.Enabled = False
TxtEdTax.Enabled = False
TxtSurcharge.Enabled = False
```

OR

```
TxtITax.Locked = True
TxtEdTax.Locked = True
TxtSurcharge.Locked = True
```

(1 Mark each for any two correct commands)

Note:

- **0 and 1 instead of False and True to be considered correct**
- **Any additional statements in the code may be ignored**
- **Ignore Cases (Lower/Upper) for all identifiers in the above Codes**
- **Ignore use of VAL functions used in above code**

(b)

```
If Not(IsNumeric(TxtIncome.Text)) Then
   Msgbox (“Please enter numbers only”)
End If
```

OR

```
If Not (KeyAscii >= Asc(“0”) And KeyAscii <= Asc(“9”) Then
   Msgbox (“Please enter numbers only”)
End If
```

OR

```
If VarType(TxtIncome.Text) <> vbInteger Then
   Msgbox (“Please enter numbers only”)
End If
```

OR

```
If Not(Right(TxtIncome.text,1) >= “0” And
      Right(TxtIncome.text,1) <=”9” then
TxtIncome.text = “”   or msgbox(“Enter only numeric value”)
End if
```

OR

Any other equivalent code

(1 Mark for correct If statement)

(1 Mark for displaying error message or assigning TxtIncome as blank)

Note:

- **Ignore Cases (Lower/Upper) for all identifiers in the above Codes**

(c)

```
TxtEmpCode.Text = “ “
TxtEmpName.Text = “ “
TxtIncome.Text = 0
TxtITax.Text = 0
TxtEdTax.Text = 0
TxtSurcharge.Text = 0
TxtTotalTax.Text = 0
```

OR

Any other equivalent code

(½ Mark each for setting EmpCode and EmpName to blank)

(½ Mark each for setting any two of the other textboxes to 0)

Note:

- **Ignore Cases (Lower/Upper) for all identifiers in the above Codes**

(d)

```
Dim Income, Itax, Edtax, Surcharge, Totaltax
Income=Val(TxtIncome.Text)
If Income>250000 Then
   Itax=25000 + 0.3*(Income-250000)
   Edtax=0.02 * Income
   Surcharge=0.01 * Income
Elseif Income>150000 Then
   Itax=0.2*(Income-150000)
   Edtax=0.02 * Income
   Surcharge=0
Elseif Income>100000 Then
   Itax=5000 + 0.1*(Income-100000)
   Edtax=0.02 * Income
   Surcharge=0
Else
   Itax=0
   Edtax=0
   Surcharge=0
End If
TxtITax.Text = Itax
TxtEdTax.Text = Edtax
TxtSurcharge.Text = Surcharge
TxtTotalTax.Text = Itax + Edtax + Surcharge
```

OR

Any other equivalent code

(1 mark for IF-THEN-ELSEIF statement)
(1 mark for calculating Income Tax for any one slab)
(½ mark for calculating Education Tax for any one slab)
(½ mark for calculating Surcharge for any one slab)
(½ mark for calculating Total Tax)
(½ mark for displaying the values in correct textboxes)

Note:

- **Ignore Cases (Lower/Upper) for all identifiers in the above Codes**
- **Ignore use of VAL functions used in above code**

5. (a) 4 times

OR

Output :

4

2

0

-2

(2 Marks for correct answer)

Note: Full marks to be given if student has written the output instead of number of times.

(b) Class XIIXI

7 **OR** 6

xi **OR** xii

6

(½ mark for each line of correct output)

Note:

Deduct ½ mark, if the correct values of output is shown in a single line

(c)

```
If Code>45 Then
    Message1="Error"
Elseif Code>=10 And Code<=20 Then
    Message1="Accounts"
Elseif Code>=21 And Code<=30 Then
    Message1="Personnel"
Elseif Code>=31 And Code<=45 Then
    Message1="EDP"
Else
    Message1="Access Denied"
End If
```

OR

```
If Code>45 Then
   Message1=”Error”
Elseif Code>=31 Then
   Message1=”EDP”
Elseif Code>=21 Then
   Message1=”Personnel”
Elseif Code>=10 Then
   Message1=”Accounts”
Else
   Message1=”Access Denied”
End If
```

(½ mark for IF statement)

(½ mark for any one correct ELSEIF statement)

(½ mark for ELSE statement)

(½ mark for END IF statement)

(d)

```
Function Check(a As Integer, b As Integer)
If a = b Then
   Check = 0
Elseif a > b Then
   Check = -1
Else
   Check = 1
End If
End Function
```

(1 Mark for header of Function)

(1 Mark for checking equlity and assigning value to be returned)

(1 Mark for checking first argument is greater than second and assigning value to be returned)

(1 Mark for checking first argument is lesser than second and assigning value to be returned)

SECTION - C

6. (a)

```
DECLARE
   CURSOR emp_cursor IS
   SELECT Eno, Ename FROM Emp
   WHERE Loc=’Delhi’;
```

(½ Mark for DECLARE)

(½ Mark for creating CURSOR)

(½ Mark for SELECT)

(½ Mark for WHERE)

(b) Output:

7

6

5

-4

(½ Mark for each correct line of output)

Note:

Deduct ½ mark, if the correct values of output is shown in a single line

(c)

```
DECLARE
  CTR NUMBER;
  SUM1 NUMBER;
BEGIN
  CTR:=0;
  SUM1:=0;
LOOP
  SUM1:=SUM1 + CTR;
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(SUM1));  —Error 1
  CTR:=CTR-2;
  IF CTR>2 THEN
     EXIT;
  END IF;                               —Error 2
END LOOP;
END;                                    —Error 3
```

(1 Mark for identifying <u>and</u> correcting any one error)

(1 ½ Marks for identifying <u>and</u> correcting any two errors)

(2 Marks for identifying <u>and</u> correcting all three errors)

OR

(1 Mark to be given if all the errors are identified without correction)

OR

(1 Mark for only identifying Logical Error as Infinite Loop due to condition CTR>2)

Note: Identification or Correction of Logical error to be ignored, if Syntax Errors are identified/corrected.

(d)

```
CREATE OR REPLACE PROCEDURE Select_employee IS
  v_id Number:=100;
  v_fname Emp.FIRST_NAME%TYPE;
  v_lname Emp.LAST_NAME%TYPE;
  v_joindt Emp.JOIN_DATE%TYPE;
  v_sal Emp.SALARY%TYPE;
```

```
BEGIN
  SELECT FIRST_NAME, LAST_NAME, JOIN_DATE, SALARY INTO v_fname,
  v_lname, v_joindt, v_sal FROM Emp  WHERE EMployee_ID=v_id;
DBMS_OUTPUT.PUT_LINE(v_fname||’‘||v_lname||’‘||v_joindt||’ ‘||v_sal);
END Select_employee;
```

OR

```
CREATE OR REPLACE PROCEDURE Select_employee IS
  v_id Number:=100;
  v_fname VARCHAR2(30);
  v_lname VARCHAR2(30);
  v_joindt DATE;
  v_sal NUMBER(11,2);
BEGIN
  SELECT FIRST_NAME, LAST_NAME, JOIN_DATE, SALARY INTO  v_fname,
  v_lname, v_joindt, v_sal FROM Emp  WHERE EMployee_ID=v_id;
  DBMS_OUTPUT.PUT_LINE(v_fname||’ ‘||v_lname);
  DBMS_OUTPUT.PUT_LINE(to_char(v_joindt,’DD-Mon-YYYY’));
  DBMS_OUTPUT.PUT_LINE(to_char(v_sal,’999,990.99'));
END Select_employee;
```

OR

Any other equivalent code

(1 Mark for header of procedure)
(1 Mark for declaring variables)
(1 Mark for correct select statement)
(1 Mark for display)

7. (a)

```
SELECT  RollNumber,  Name,  TotalMarks,  Grade  FROM
Student
WHERE Stream= ‘Nonmedical’
ORDER BY TotalMarks DESC;
```

(½ Mark for selection)
(½ Mark for Where clause)
(½ Mark for Order By clause)
(½ Mark for DESC)

(b)

```
CREATE VIEW stud_view AS
SELECT * FROM Student
WHERE Stream= ‘Medical’ and Grade=’A’;
```

(1 Mark for Create View command)
(½ Mark for Select)
(½ Mark for Where clause)

(c)

```
CREATE OR REPLACE PROCEDURE inc_marks(rno NUMBER) IS
BEGIN
   UPDATE Student
   SET TotalMarks = TotalMarks + 0.02 * TotalMarks
   WHERE RollNumber = rno;
END inc_marks;
```

OR

Any other equivalent code

(1 mark for procedure header)
(1 mark for UPDATE- SET command)
(½ mark for correct calculation of total marks)
(½ mark for WHERE clause)

(d)

```
CREATE TRIGGER Change_comm
BEFORE DELETE ON Student    OR  AFTER DELETE ON Student
FOR EACH ROW
BEGIN
DBMS_OUTPUT.PUT_LINE(:old.name);
DBMS_OUTPUT.PUT_LINE(:old.Stream);
END;
```

OR

Any other equivalent code

(1 Mark for CREATE TRIGGER trigger_name)
(½ Mark for BEFORE DELETE ON table_name)
(½ Mark for FOR EACH ROW)
(½ Mark for displaying name and stream)
(½ Mark for BEGIN and END)