



**Do not award half marks.
In all cases give credit for appropriate alternative answers.**

Question 1 (Compulsory)

- (a) Define the following terms.
- (i) PL/SQL [2]
 - (ii) Index [2]
 - (i) **PL/SQL is an Oracle procedural language that extends SQL [1], with design features of programming languages [1].**
 - (ii) **An index is an object that can speed up the retrieval of rows [1] and improve query performance [1].**
- (b) List *four* categories of data type supported by PL/SQL. [4]
Scalar [1]; composite; reference [1]; and LOB [1].
- (c) List *three* types of exception found in Oracle database. [3]
User-defined error [1]; predefined Oracle Server error; non-predefined Oracle server error [1].
- (d) In which section of a PL/SQL block would you find functions for error trapping? [1]
Exception [1].

- (e) (i) What is a *system privilege*? [1]
- (ii) Give *two* example of a system privilege command. [2]
- (i) **A system privilege is a privilege required to gain access to the database (1 mark).**
- (ii) **One mark should be awarded for each example (up to a maximum of two marks). Examples include the following.**
- **CREATE USER.**
 - **DROP USER.**
 - **DROP TABLE.**
 - **BACKUP TABLE.**
- (f) (i) What is an *object privilege*? [1]
- (ii) Give *two* examples of an object privilege command. [2]
- (i) **An object privilege is a privilege or right to perform a particular action on a specific table, view, sequence, or procedure.**
- (ii) **One mark should be awarded for each example (up to a maximum of two marks). Examples include the following.**
- **SELECT.**
 - **ALTER.**
 - **UPDATE.**
 - **REFERENCES.**
 - **INSERT.**
- (g) (i) In the context of PL/SQL, what are *constraints*? [2]
- (ii) List *three* types of PL/SQL constraints. [3]
- (i) **Constraints enforce rules at the table level (1 mark), and prevent the deletion of a table if there are dependencies (1 mark).**
- (ii) **One mark should be awarded for each type of constraint named (up to a maximum of three marks). The constraints are as follows.**
- **NOT NULL.**
 - **UNIQUE.**
 - **PRIMARY KEY.**
 - **FOREIGN KEY.**
 - **CHECK.**

- (h) Referring to the ENROLMENT table given at the end of this paper, create a PL/SQL block to determine the *average* grade for a particular course. [7]

SET VERIFY OFF
SET SERVEROUTPUT ON [1]

ACCEPT p_courseno PROMPT 'Enter the course number: ' [1]

DECLARE
 v_average_grade **NUMBER;** [1]

BEGIN
 SELECT **AVG(grade)**
 INTO **v_average_grade** [1]

FROM **enrolment**
 WHERE **course_no = &p_courseno;** [1]

Dbms_output.put_line('The average grade: ' ||
 TO_CHAR(v_average_grade)); [1]

END; [1]
/

Do not award half marks.
In all cases give credit for appropriate alternative answers.

Question 2

- (a) What is the default date format used in an Oracle database? [1]
DD-MON-YY [1]
- (b) In the context of Oracle database, what is *SYSDATE*? [2]
SYSDATE is a function [1] that returns the current date and time. [1]
- (c) List *three* SQL DML statements that are supported in Oracle database. [3]
The statements are INSERT [1], UPDATE [1], and DELETE [1].

(d) Consider the tables provided at the end of this paper (COURSE, SECTION, ENROLMENT, STUDENT and INSTRUCTOR).

(i) Create a script to insert a new student into the STUDENT table:

- use SQL*Plus substitution variables for the student id and student name.
- join date shall be the current date and time.
- leave the address as null.

Make the changes permanent. [4]

```
ACCEPT p_student_id PROMPT 'Enter the student id: '
ACCEPT p_name PROMPT 'Enter the name: '
INSERT INTO student
VALUES (&p_student_id, '&p_name', NULL, sysdate)
COMMIT; [1]
[1]
[1]
[1]
```

(ii) Write a SQL statement to reduce the cost of all courses by 10%. [2]

```
UPDATE    courses
SET       cost = cost * 0.9; [1]
[1]
```

(iii) Explain the error encounter when executing the statement below. [3]

```
SQL> DELETE FROM    section
      WHERE          section_id = 107;
```

```
DELETE FROM    section
              *
```

```
ERROR;
ORA-02292: integrity constraint (ENROLMENT_SECTION_ID_FK)
Violated – child record found
```

Section ID is used as a foreign key in another (ENROLMENT) table.
[1] Hence, the parent record that you attempt to delete has child records, [1] and therefore deletion is not allowed [1].

Do not award half marks.
In all cases give credit for appropriate alternative answers.

Question 3

(a) (i) What is a view? [1]

(ii) List *three* advantages of creating views. [3]

Advantages are:

- **Restrict database access**
- **Simplify queries**
- **Provide data independence**
- **Allow multiple views of the same data**
- **Can be removed without affecting the underlying data**

[1 mark for each valid advantage, max 3 marks awarded]

(b) (i) Explain the OR REPLACE option when creating a view. [1]

OR REPLACE option allows a view to be created even if one exists with this name already, thus replacing the old version of the view [1].

(ii) Explain the WITH READ ONLY clause when creating a view. [1]

To disallow any DML operations on the view [1].

(c) Explain what happens to the view when the table on which the view is based on is dropped? [1]

The view will remain but is invalid [1]

(d) Create the following queries from the tables provided at the end of this paper (COURSE, SECTION, ENROLMENT, STUDENT and INSTRUCTOR).

(i) Determine the five highest grades of section 107. [4]

```
SELECT ROWNUM, grade [1]  
FROM (SELECT DISTINCT grade  
FROM enrolment  
WHERE section_id = 107 [1]  
ORDER BY grade DESC) [1]  
WHERE ROWNUM <= 5; [1]
```

(ii) Create a view named SECTION107 that contains the student id, student name and grade for all students enrolled in section 107. Do not allow any DML operations on the view. [4]

```
CREATE VIEW section107 AS [1]  
SELECT student_id, name, grade [1]  
FROM enrolment, student  
WHERE section_id = 107  
AND enrolment.student_id = student.student_id [1]  
WITH READ ONLY; [1]
```

Do not award half marks.
In all cases give credit for appropriate alternative answers.

Question 4

(a) Evaluate each of the following declarations. Determine whether or not each is legal. Justify your answer.

(i) DECLARE

v_a, v_b, v_c VARCHAR2(20); [2]

Illegal [1], because only one identifier per declaration is allowed [1].

(ii) DECLARE

v_grade BOOLEAN :=1; [2]

Illegal [1], because 1 is not a Boolean expression [1].

(iii) DECLARE

stud_name%type; [3]

Illegal [1], because this is an invalid declaration [1]. The proper declaration would have to include a predeclared variable OR a table and column reference [1] such as

stud_name student.name%type;

(b) Referring to the tables at the end of this paper, create a PL/SQL block that display a letter grade based on the student's grade. It will accept the student id and course number as user inputs with SQL*Plus substitution variables. In addition:

- If the grade is between 90 and 100, the letter grade is 'A'.
- If the grade is between 80 and 89, the letter grade is 'B'.
- If the grade is between 70 and 79, the letter grade is 'C'.
- If the grade is between 60 and 69, the letter grade is 'D'.
- If the grade is less than 60, the letter grade is 'F'. [8]

ACCEPT p_student_id PROMPT 'Please enter student id: '
ACCEPT p_course_no PROMPT 'Please enter course no: ' [1]

DECLARE
 v_student_id **enrolment.student_id%TYPE := &p_student_id;**
 v_course_no **enrolment.course_no%TYPE := &p_course_no;**
 v_grade **enrolment.grade%TYPE;**
 v_letter_grade **CHAR(1);** [1]

BEGIN
 SELECT **grade**
 INTO **v_grade**
 FROM **enrolment**
 WHERE **student_id= v_student_id** [2]
 AND **course_no=v_course_no**

IF **v_grade BETWEEN 90 AND 100**
 THEN **v_letter_grade :='A';**
 ELSIF **v_grade BETWEEN 80 AND 89**
 THEN **v_letter_grade :='B';**
 ELSIF **v_grade BETWEEN 70 AND 79**
 THEN **v_letter_grade :='C';**
 ELSIF **v_grade BETWEEN 60 AND 69**
 THEN **v_letter_grade :='D';**
 ELSIF **v_grade <60**
 THEN **v_letter_grade :='F';**
 END IF; [3]

DBMS_OUTPUT.PUT_LINE
 ('Letter grade is : " || v_letter_grade); [1]

END;
/

Do not award half marks.
In all cases give credit for appropriate alternative answers.

Question 5

- (a) (i) Differentiate between implicit cursor attributes and explicit cursor attributes. [2]

Implicit cursor attributes verify the outcome of DML statements in a PL/SQL block [1]. Explicit cursor attributes are used to obtain status information about a cursor [1].

- (ii) List *two* examples of implicit cursor attributes. [2]

Examples: SQL%NOTFOUND, SQL%FOUND, SQL%ROWCOUNT, SQL%ISOPEN

[Award 1 mark for any one valid example]

- (iii) List *two* example of an explicit cursor attributes. [2]

Examples: %NOTFOUND, %FOUND, %ROWCOUNT, %ISOPEN

[Award 1 mark for any one valid example]

- (b) Create a PL/SQL block that uses explicit cursor and cursor attributes to retrieve the first 10 instructors' instructor id and instructor name from the INSTRUCTOR table. Print them to the screen, using DBMS_OUTPUT.PUT_LINE. [9]

```
DECLARE
    v_instructorid      instructor.instructor_id%TYPE;
    v_name              instructor.name%TYPE;

    CURSOR instructor_cursor IS
        SELECT instructor_id, name
        FROM instructor;

BEGIN
    OPEN instructor_cursor;

    LOOP
        FETCH instructor_cursor INTO v_instructorid, v_name;
        EXIT WHEN instructor_cursor%ROWCOUNT>10 OR
        instructor_cursor%NOTFOUND
        DBMS_OUTPUT.PUT_LINE(TO_CHAR(v_instructorid) || ', ' ||
        v_name);
    END LOOP;

    CLOSE instructor_cursor;

END;
```

- END OF PAPER -

STRICTLY CONFIDENTIAL
CS216 - August 2003 - Marking Scheme

COURSE Table

Information for one course.

Column Name	Null?	Type
COURSE_NO	NOT NULL	NUMBER(6)
DESCRIPTION		VARCHAR2(50)
PREREQUISITE		NUMBER(6)
COST		NUMBER(7,2)

SECTION Table

Represents an individual section (class) of a particular course.

Column Name	Null?	Type
SECTION_ID	NOT NULL	NUMBER(6)
COURSE_NO	NOT NULL	NUMBER(6)
START_DATE		DATE
LOCATION		VARCHAR2(50)
INSTRUCTOR_ID		NUMBER(6)
CAPACITY		NUMBER(3)

ENROLMENT Table

Shows information about one student registered to a particular section(class).

Column Name	Null?	Type
COURSE_NO	NOT NULL	NUMBER(6)
SECTION_ID	NOT NULL	NUMBER(6)
STUDENT_ID	NOT NULL	NUMBER(6)
GRADE		NUMBER(3)
ENROLL_DATE		DATE

STUDENT Table

Profile information about a student.

Column Name	Null?	Type
STUDENT_ID	NOT NULL	NUMBER(8)
NAME		VARCHAR2(30)
ADDRESS		VARCHAR2(50)
JOIN_DATE		DATE

INSTRUCTOR table

Profile information about a instructor.

Column Name	Null?	Type
INSTRUCTOR_ID	NOT NULL	NUMBER(8)
NAME		VARCHAR2(30)
ADDRESS		VARCHAR2(50)
PHONE		VARCHAR2(15)