

## Lab Practice Session II (3 Hours)

### Note:

**It is important that students complete all the problems provided in this document.**

### Problem 1

Write a PL/SQL program to generate a multiplication table of 8 up to 10 rows. Your output should look like:

1	8	8
2	8	16
--	--	--
--	--	--
10	8	80

### Problem 2

Use the script (provided below) to create the table 'multiply\_table'. Write a PL/SQL program (or you can modify the program you wrote for problem 1) to generate multiplication table of 8 and then insert the results into the 'multiply\_table' up to 20 rows.

```
create table multiplication_table(  
tab_number number,  
multiplier number,  
result number);
```

### Problem 3

Modify your program in Problem 2, to generate the multiplication table of 7 and then all the 20 rows in the table 'multiply\_table' are updated properly.

Then modify your code again to delete the odd rows in the table (like 1, 3, 5, 7 ....19).

### Problem 4

Write a PL/SQL program using the table 'moviestar' (database script is provided below. You do not have to run the script if the table is already existing in your database). to determine –

1. how many female and male stars are there.
2. What is the age of the each star as on today? (assume that all stars are alive now)

```
CREATE TABLE moviestar (  
ms_id NUMBER(5),  
ms_name VARCHAR2(20),  
ms_address VARCHAR2(30),
```

```

ms_gender CHAR(1) DEFAULT 'F' CONSTRAINT CK_ms_gender CHECK
(ms_gender IN ('M', 'F')),
ms_birthdate DATE,
CONSTRAINT pk_moviestar PRIMARY KEY (ms_id)
);

```

```

INSERT INTO moviestar(ms_id, ms_name, ms_gender, ms_birthdate)
VALUES(1, 'Henry FONDA', 'M', TO_DATE('16/05/1905', 'dd/mm/yyyy'));

```

```

INSERT INTO moviestar(ms_id, ms_name, ms_gender, ms_birthdate)
VALUES(2, 'Vivien LEIGH', 'F', TO_DATE('05/11/1913', 'dd/mm/yyyy'));

```

```

INSERT INTO moviestar(ms_id, ms_name, ms_gender, ms_birthdate)
VALUES(3, 'Harrison FORD', 'M', TO_DATE('13/07/1942', 'dd/mm/yyyy'));

```

```

INSERT INTO moviestar(ms_id, ms_name, ms_gender, ms_birthdate)
VALUES(4, 'Mark HAMIL', 'M', TO_DATE('25/09/1951', 'dd/mm/yyyy'));

```

```

INSERT INTO moviestar(ms_id, ms_name, ms_gender, ms_birthdate)
VALUES(5, 'Kate WINSLET', 'F', TO_DATE('05/10/1975', 'dd/mm/yyyy'));

```

```

INSERT INTO moviestar(ms_id, ms_name, ms_gender, ms_birthdate)
VALUES(6, 'Leonardo DI CAPRIO', 'M', TO_DATE('11/11/1974', 'dd/mm/yyyy'));

```

```

INSERT INTO moviestar(ms_id, ms_name, ms_gender, ms_birthdate)
VALUES(7, 'Sean CONNERY', 'M', TO_DATE('30/08/1930', 'dd/mm/yyyy'));

```

```

INSERT INTO moviestar(ms_id, ms_name, ms_gender)
VALUES(8, 'Naomie WATTS', 'F');

```

### Problem 5

A pharmacy orders different drugs on whole-sale basis and sells them to the customers on retail basis. Use the script (provided below) to create the table and insert the data. Then write a PL/SQL program to output – drug id, drug name, how many units purchase, how many sold, how much the profit was made. Also determine the below items.

1. Overall profit
2. Drugs that were not sold at all
3. Drugs that were completely sold and out of stock.

Note: you have to use any PL/SQL Loop statement and Cursor Statement and anchored variables.

```

create table pharmacy_purchases(
drug_id number (4),

```

```

drug_name varchar2(25),
unit_purchase_price number(8,2),
qty_ordered number(5),
constraint pk_drug primary key(drug_id));

```

```

Insert into pharmacy_purchases values (101, 'Panadol', 1.25, 7220);
Insert into pharmacy_purchases values (103, 'Asprin', 3.68, 2100);
Insert into pharmacy_purchases values (105, 'Ciprolet', 4.70, 5010);
Insert into pharmacy_purchases values (113, 'Eteroxoib', 2.27, 4015);
Insert into pharmacy_purchases values (119, 'Nerogaban', 4.45, 6011);
Insert into pharmacy_purchases values (122, 'Roxid', 4.45, 2864);

```

```

Commit;

```

```

create table pharmacy_sales(
drug_id number (4),
unit_sale_price number(8,2),
qty_sold number(5),
constraint fk_drug foreign key(drug_id)
references pharmacy_purchases (drug_id));

```

```

insert into pharmacy_sales values (101, 2.00, 250);
insert into pharmacy_sales values (103, 4.50, 800);
insert into pharmacy_sales values (105, 6.00, 400);
insert into pharmacy_sales values (113, 3.50, 380);
insert into pharmacy_sales values (119, 6.00, 900);
commit;

```

### Problem 6

It is required to write a PL/SQL program to prepare grade sheet for a given course in a semester. The course has 1 mid term examination, 2 assignments and a final end-term examination. The assessment weighting is shown below.

Assessment Component	Marks
Mid Term Examination	30
Assignment 1	10
Assignment 2	10
Final end term Examination	50
<b>Total</b>	<b>100</b>

The following Grading scheme needs to be used to determine the grade in the course.

Range of Marks	Grade
80 - 100	A+ (Excellent)
70 - 79	A
67 - 69	A-
63 - 66	B+
60 - 62	B
57 - 59	B-
47 - 56	C
40 - 46	D (Threshold)
0 - 39	F (Fail)

The database script for this problem is provided below. Write a PL/SQL program to update the column 'grades' column. Use only IF statements.

```
drop table course_marks;
```

```
create table course_marks(  
  student_id   number(9),  
  student_name varchar2(20),  
  quiz_marks   number(2),  
  a1_marks     number(2),  
  a2_marks     number(2),  
  final_exam  number(2),  
  total_marks  number(3));
```

```
insert into course_marks values (0112141,'John Freeman',3,6,2,11,"");  
insert into course_marks values (0112142,'Ali Khan',5,10,10,25,"");  
insert into course_marks values (0112151,'Kit Pun',9,18,19,48,"");  
insert into course_marks values (0113121,'Linda George',0,2,1,7,"");  
insert into course_marks values (0113143,'Ken Balfour',7,14,15,34,"");  
insert into course_marks values (0112144,'Daina Wong',6,12,14,32,"");  
insert into course_marks values (0112145,'Susila Wong',8,16,17,30,"");  
insert into course_marks values (0112146,'May Ann George',3,6,2,11,"");  
insert into course_marks values (0112147,'Kelly Edward',9,18,19,40,"");  
insert into course_marks values (0114152,'Kevin Khan',6,11,12,26,"");  
insert into course_marks values (0114153,'Frederick Foreman',5,11,11,27,"");  
insert into course_marks values (0114154,'Tom Ridgeman',3,6,2,45,"");  
insert into course_marks values (0114155,'David Lee',4,8,12,32,"");  
insert into course_marks values (0114156,'Leena George',7,6,12,38,"");
```

### Problem 7

Modify your program in problem 6 with PL/SQL 'Case' statement (instead of 'IF'). And to determine how many students got A+ grade, how many students got A grade and so on for all grades.

### Problem 8

A small library manages the book lending to the students using a database. See the script below. It is required to determine

1. Determine the usage of each and every book by listing each book, how many times it was borrowed, minimum days of usage by the borrowers, maximum days of usage by the borrowers.
2. Determine how the students are using the book store facility by listing each student, how many times the student borrowed the books, average days of reading per book
3. What is the most popular book?
4. Who is the active borrower? Name and ID are required.
5. List the names and IDs of students who did not return the books on time.

```
create table book_base(  
book_id number(9),  
title varchar2(90),  
date_published date,  
pages number(5),  
cost number(9,2),  
num_of_copies number(3),  
constraint pk_bookid primary key (book_id));
```

```
create table student(  
student_id number(9),  
name varchar2(50),  
mobile number(7),  
constraint pk_studid primary key (student_id));
```

```
create table books_borrowed(  
book_id number(9),  
student_id number(9),  
date_borrowed date,  
return_before date,  
actual_date_returned date,  
constraint fk_bookid foreign key(book_id) references book_base(book_id),  
constraint fk_student_id foreign key(student_id) references student(student_id));
```

```
insert into student values(101,'Carlos Williams', 292344);  
insert into student values(102,'Lucey Duggon', 2875454);  
insert into student values(201,'Marylin Waltrust', 2671213);  
insert into student values (112, 'Victor Henry', 2547862);  
insert into student values (115, 'Rose-Marry Wilson', 2587654);  
insert into student values (108, 'Robert Winters', 2579201);
```

```
INSERT INTO BOOK_BASE VALUES (22154, 'Introduction C++ Programming',  
TO_DATE('12-03-1994', 'DD-MM-YYYY'), 452, 20, 25);  
INSERT INTO BOOK_BASE VALUES (31225, 'Database Management Systems',  
TO_DATE('17-08-2006', 'DD-MM-YYYY'), 312, 23, 20);
```

```
INSERT INTO BOOK_BASE VALUES (67123, 'Java Programming', TO_DATE('29-04-2008', 'DD-MM-YYYY'), 387, 25, 30);
```

```
insert into books_borrowed VALUES (22154, 112, TO_DATE('12-10-2015', 'DD-MM-YYYY'), TO_DATE('12-11-2015', 'DD-MM-YYYY'), TO_DATE('2-11-2015', 'DD-MM-YYYY'));
```

```
insert into books_borrowed VALUES (31225, 201, TO_DATE('11-1-2015', 'DD-MM-YYYY'), TO_DATE('12-2-2015', 'DD-MM-YYYY'), TO_DATE('2-2-2015', 'DD-MM-YYYY'));
```

```
insert into books_borrowed VALUES (22154, 112, TO_DATE('6-4-2015', 'DD-MM-YYYY'), TO_DATE('6-5-2015', 'DD-MM-YYYY'), TO_DATE('12-4-2015', 'DD-MM-YYYY'));
```

```
insert into books_borrowed VALUES (67123, 115, TO_DATE('20-11-2015', 'DD-MM-YYYY'), TO_DATE('20-12-2015', 'DD-MM-YYYY'),");
```

```
insert into books_borrowed VALUES (22154, 108, TO_DATE('16-7-2015', 'DD-MM-YYYY'), TO_DATE('16-8-2015', 'DD-MM-YYYY'), TO_DATE('2-8-2015', 'DD-MM-YYYY'));
```

```
insert into books_borrowed VALUES (31225, 101, TO_DATE('13-3-2015', 'DD-MM-YYYY'), TO_DATE('12-4-2015', 'DD-MM-YYYY'), TO_DATE('24-3-2015', 'DD-MM-YYYY'));
```

```
insert into books_borrowed VALUES (31225, 102, TO_DATE('22-2-2015', 'DD-MM-YYYY'), TO_DATE('22-5-2015', 'DD-MM-YYYY'), TO_DATE('24-6-2015', 'DD-MM-YYYY'));
```