CO2 Lab Assignments
Procedures and Functions

1. Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation
   Hint: CREATE TABLE T2 (a INTEGER, b CHAR(10));

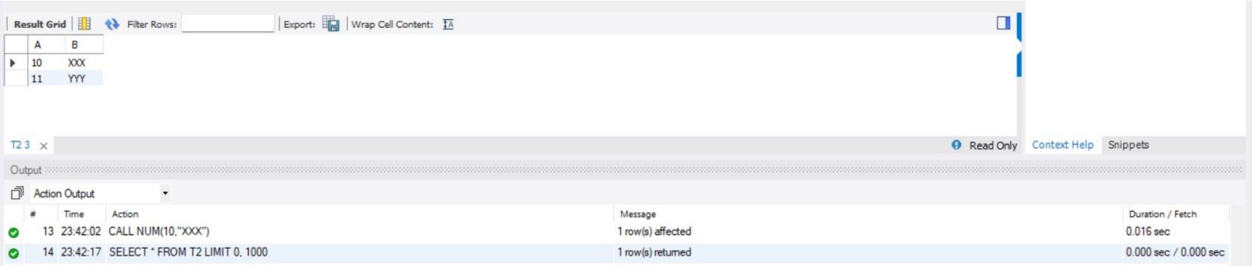
   ANSWER

```
CREATE DATABASE NUMBER;
USE NUMBER;
CREATE TABLE T2(A INT ,
                                    B CHAR(10) );
DROP TABLE T2;
CALL NUM(10,"XXX");
CALL NUM(11,"YYY");

SELECT * FROM T2;
SHOW DATABASES;
```

   STORED PROCEDURE :

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `NUM`(I INT,J CHAR(10))
BEGIN
IF(SELECT A FROM T2 WHERE A LIKE (I))
THEN
INSERT INTO T2 (A,B) VALUES (NULL,NULL);
ELSE
INSERT INTO T2 (A,B) VALUES (I,J);
END IF;
END
```

   OUTPUT



A	B
10	XXX
11	YYY

#	Time	Action	Message	Duration / Fetch
13	23:42:02	CALL NUM(10,"XXX")	1 row(s) affected	0.016 sec
14	23:42:17	SELECT * FROM T2 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

2. **W**rite a PL/SQL block to calculate the incentive of an employee whose ID is 110

   ANSWER

Table creation and insertion

```
CREATE TABLE employee(id int,basic double,hra double);
INSERT INTO employee VALUES(101,12000,3200),(102,15000,3200);
```

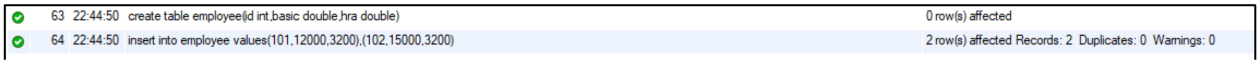
Function call

```
SELECT *,incentive(id) FROM employee;
```

FUNCTION :

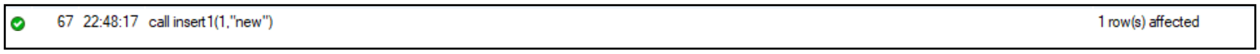
```
CREATE DEFINER=`root`@`localhost` FUNCTION `incentive`(id1 int) RETURNS
double
BEGIN
 DECLARE bp double;
 DECLARE h double;
 DECLARE inc double;
 SELECT basic INTO bp
          FROM employee
          WHERE id=id1;
 SELECT hra INTO h
          FROM employee
          WHERE id=id1;
 if(bp>10000) then
          set inc=bp+h+1200;
 else
          set inc=bp+h+4500;
 end if;
RETURN inc;
END
```

OUTPUT



```
✓  63  22:44:50  create table employee(id int,basic double,hra double)               0 row(s) affected
✓  64  22:44:50  insert into employee values(101,12000,3200),(102,15000,3200)        2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0
```

Function call :



```
✓   67  22:48:17  call insert1(1,"new")                                              1 row(s) affected
```

3. Create the Book database and do the following: (Consider the attributes based on the question given)
   book(book_name, author_name, price,quantity)

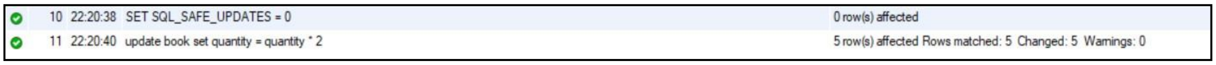
   a. Write a query to update the quantity by double in the table book.
   b. List all the book_name whose price is greater than those of book named "Database for Dummies"
   c. Retrieve the list of author_name whose first letter is 'a' along with the book_name and price (Explore more about *Like* keyword)
   d. Write a PL/SQL Procedure to find the total number of books of same author

```
create table book(book_name varchar(20),author_name varchar(20),price int,quantity int);
```

ANSWERS:

a.

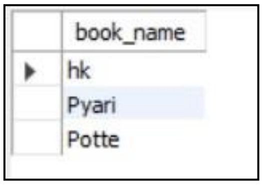
```
update book set quantity = quantity * 2;
```



```
✓  10  22:20:38  SET SQL_SAFE_UPDATES = 0                      0 row(s) affected
✓  11  22:20:40  update book set quantity = quantity * 2       5 row(s) affected Rows matched: 5  Changed: 5  Warnings: 0
```

b.

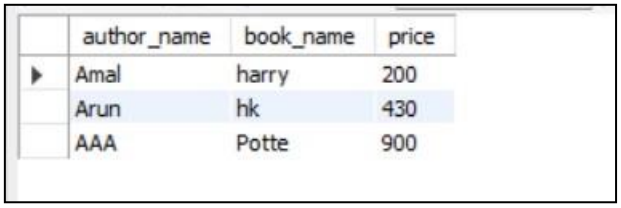
select book_name from book where price > (select price from book where book_name = 'database for dummies');



	book_name
▶	hk
	Pyari
	Potte

c.

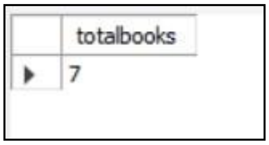
select author_name,book_name,price from book where author_name like 'a%';



	author_name	book_name	price
▶	Amal	harry	200
	Arun	hk	430
	AAA	Potte	900

d.

```
call total('Stanlee');

CREATE DEFINER=`root`@`localhost` PROCEDURE `total`(auth varchar(20))

BEGIN

        select sum(quantity) as totalbooks from book where author_name like auth;

END
```



	totalbooks
▶	7

4. Create the Company database with the following tables and do the following:
   Administration (employee_salary, development _cost, fund_ amount, turn_over,bonus)
   Emp_details (emp_no, emp_name, DOB, address, doj, mobile_no, dept_no, salary).

   a. Calculate the total and average salary amount of the employees of each department.

   b. Display total salary spent for employees.

   c. Develop a PL/SQL function to display total fund_amount spent by the administration department

ANSWERS

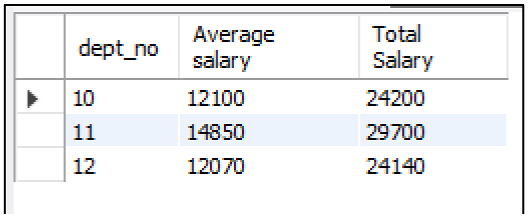
Table creation and insertion

```
CREATE TABLE Administration (
employee_salary double,
development_cost double,
fund_amount double,
turn_over double,
bonus double);

CREATE TABLE Emp_details(
emp_no int,
emp_name varchar(20),
DOB date,
address varchar(20),
doj date,
mobile_no int(12),
dept_no int,
salary double);

INSERT INTO Administration VALUES
(12000,25000,560000,65000,5000),
(70000,55000,860000,15000,1000),
(18000,45000,160000,75000,7000),
(10000,27000,520000,60000,5000),
(18000,27000,360000,35000,3000);

INSERT INTO Emp_details VALUES
(1,"Ram","1999-10-10","Street - 2,vallakadavu","2020-10-10",9865986598,10,12000),
(2,"manoharan","1997-10-10","Street - 2,vallakadavu","2020-10-10",9865986598,10,12200),
(3,"mani","1996-10-10","Street - 2,vallakadavu","2020-10-10",9865986598,11,12500),
(4,"moran","1957-10-10","Street - 2,vallakadavu","2020-10-10",9865986598,11,17200),
(5,"sasi","1948-10-10","Street - 2,vallakadavu","2020-10-10",9865986598,12,12090),
(6,"kaka","1988-10-10","Street - 2,vallakadavu","2020-10-10-",9865986598,12,12050);
```

a.

```
SELECT
        dept_no,
        avg(salary) 'Average salary',
        sum(salary) 'Total Salary'
FROM Emp_details
GROUP BY dept_no;
```

OUTPUT

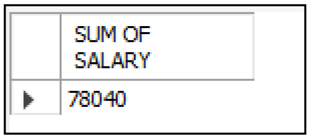


	dept_no	Average salary	Total Salary
▶	10	12100	24200
	11	14850	29700
	12	12070	24140

b.

```
SELECT
        sum(salary) 'SUM OF SALARY'
FROM Emp_details;
```

OUTPUT



	SUM OF SALARY
▶	78040

c.

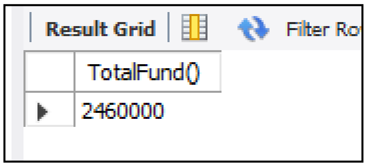
FUNCTION :

```
CREATE DEFINER=`root`@`localhost` FUNCTION `TotalFund`() RETURNS double
BEGIN
        DECLARE f DOUBLE;
   DECLARE i DOUBLE;
   SELECT SUM(fund_amount) INTO f
                FROM Administration;
RETURN f;
END
```

Function call :

```
SELECT TotalFund() from Administration LIMIT 1;
```

OUTPUT



	TotalFund()
▶	2460000