-- QUESTION 3
/*
STUDENT DATABASE
class_(class_id, class_name, division, st_cnt)
student(st_id, st_fname, st_lname, addr, phone, email, class_id)
teacher(teacher_id, fname, lname, phone, subject)
student_class(st_id, class_id, teacher_id)
*/

-- 1)Create all the tables and insert data.

-- 2)List all classes where strength is greater than 50

-- 3)List the name of students of Lina teacher.

-- 4)List the names of all the English teachers.

-- 5)List the names of teachers who teach standard 9

-- 6)Find out all the classes that are taught by Jaya teacher

-- 7)List the names and details of all students in standard 10

-- 8) List all the students whose first name is the same along with their student id

-- 9)List the name of students whose name starts with 's'.

-- QUESTION 4 - PROCEDURE
/*
BANK DATABASE
branch(b_id, bname, city)
customer(c_id, cname, city)
deposit(acc_no, c_id, b_id, amount, date)
borrow(loan_no, c_id, b_id, amount, date)
*/

-- 1) Find the number of customers who have loan in each branch

-- 2)List all details of all customers

--QUESTION 5 - PROCEDURE
/*
EMPLOYEE DATABASE
employee(emp_id, name, dob, doj, sal, dept_id)
department(dept_id, dept_name)
*/

-- 1)Write a function for updating the salary of employees working in the department with dept_id=10 by 20%

-- 2)Write a function for employee table which accepts dept_id and return the highest salary in that department. Handle the error if the dept_id does not exist or if the query return more than

one maximum

-- 3)Write a function which accepts emp_id and returns employee experience

--QUESTION 6 - TRIGGER
/*
TRIGGERS
department(dept_id, dept_name)
employee(emp_id, name, dob, doj, sal, dept_id)
emp_backup(emp_id, name, dob, doj, sal, dept_id, date_of_op, type_of_op)
income_tax(emp_id, name, dob, doj, sal, dept_id, tax_amount)
*/

-- 1) Write a trigger which converts employee name into upper case if it is entered in lower case.

-- 2)Write a trigger that stores that data of emplyee table in emp_backup for every delete operation and store the old data for every update operation

-- 3)Write a trigger which displays the message 'updating', 'deleting' or 'inserting' when the corresponding operation is performed on the employee table.

-- 4) Write a trigger that ensures that the emp_id is of the form 'E_____'. If the inserted id is not in this form convert it and then inserted

-- 5)Write a trigger that checks the age of the employee while inserting the record in employee table. If age is negative generate error and display proper message.

-- 6)Create and execute a trigger that allows updation only if the new salary is 80% more than the original salary.

-- 7)Write a trigger to calculate income tax and insert it into income_tax table. Calculate income tax as follows if the annual income is: a)Greater than 15000 but less than 100000 tax is 10% of annual salary. b)Greater than 100000 but less than 200000 tax is 15% of annual salary. c)Greater than 200000 tax is 20$ of annual salary.