```
-- Write a query to print the ID, FIRST_NAME, and LAST_NAMEs of the customers whose
-- combined name length, i.e., the sum of the length of
the FIRST_NAME and LAST_NAME,
-- is less than 12. The IDs and names should be printed in the ascending order of the
-- combined name length. If two or more customers have the same combined name length,
-- sort the result in lexicographical order of the full names, ignoring case. If two
-- or more customers have the same full name, sort those results by ID, ascending.

select ID, first_name, last_name, length(concat(first_name, last_name)) as No_of_values
from dummy.customer
having No_of_values < 12
order by No_of_values, first_name, last_name, id;

-- Given two tables, Employee and Department, generate a summary of how many employees are
-- in each department. Each department should be listed, whether they currently have any
-- employees or not. The results should be sorted from high to low by number of employees,
-- and then alphabetically by department when departments have the same number of employees.
-- The results should list the department name followed by the employee count.
-- The column names are not tested, so use whatever is appropriate.

select count(e.id) as Number_of_empl,
d.id,
d.name as Department_name
from employee as e
right join department as d
on e.dept_id = d.id
group by d.id
order by Number_of_empl desc, Department_name;

select * from department;

-- There are two data tables with employee information: EMPLOYEE and EMPLOYEE_UIN.
-- Query the tables to generate a list of all employees who are less than 25 years
-- old first in order of NAME, then of ID, both ascending.  The result should include
-- the UIN followed by the NAME.
select * from dummy2.employee;
select * from dummy2.employee_uin;

select e_u.uin, e.name
from dummy2.employee as e
join dummy2.employee_uin as e_u
on e.id = e_u.id
```

```
where age <25
order by name, e.id;

-- A company maintains an EMPLOYEE table with information for each of their
employees.
-- Write a query to produce a list containing two columns.  The first column should
include
-- the name of an employee who earns less than some other employee.  The second
column should
-- contain the name of a higher earning employee.  All combinations of lesser and
greater
-- earning employees should be included.  Sort ascending, first by the lower earning
employee's
 -- ID, then by the higher earning employee's SALARY.

 select * from employee;

select e1.name, e2.name, e1.salary, e2.salary
from dummy3.employee e1
join dummy3.employee e2
on e1.name <> e2.name
where e1.salary < e2.salary
order by e1.salary, e2.salary;

select e1.name, e2.name, e1.salary, e2.salary
from dummy3.employee as e1
join dummy3.employee as e2
on e1.id <> e2.id and e1.salary < e2.salary
order by e1.salary, e2.salary;
```