QUESTION 1  
Table: Person

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| personId | int |

| lastName | varchar |

| firstName | varchar |

+-------------+---------+

personId is the primary key column for this table.

This table contains information about the ID of some persons and their first and last names.

Table: Address

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| addressId | int |

| personId | int |

| city | varchar |

| state | varchar |

+-------------+---------+

addressId is the primary key column for this table.

Each row of this table contains information about the city and state of one person with ID = PersonId.

Write an SQL query to report the first name, last name, city, and state of each person in the Person table. If the address of a personId is not present in the Address table, report null instead.

Return the result table in **any order**.

The query result format is in the following example.

ANS:

QUESTION2  
Table: Courses

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| student | varchar |

| class | varchar |

+-------------+---------+

In SQL, (student, class) is the primary key column for this table.

Each row of this table indicates the name of a student and the class in which they are enrolled.

Find all the classes that have **at least five students**.

Return the result table in **any order**.

The result format is in the following example

**Example 1:**

**Input:**

Courses table:

+---------+----------+

| student | class |

+---------+----------+

| A | Math |

| B | English |

| C | Math |

| D | Biology |

| E | Math |

| F | Computer |

| G | Math |

| H | Math |

| I | Math |

+---------+----------+

**Output:**

+---------+

| class |

+---------+

| Math |

+---------+

**Explanation:**

- Math has 6 students, so we include it.

- English has 1 student, so we do not include it.

- Biology has 1 student, so we do not include it.

- Computer has 1 student, so we do not include it.

ANS:

QUESTION3

Table: Logs

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| id | int |

| num | varchar |

+-------------+---------+

id is the primary key for this table.

id is an autoincrement column.

Write an SQL query to find all numbers that appear at least three times consecutively.

Return the result table in **any order**.

The query result format is in the following example.

**Example 1:**

**Input:**

Logs table:

+----+-----+

| id | num |

+----+-----+

| 1 | 1 |

| 2 | 1 |

| 3 | 1 |

| 4 | 2 |

| 5 | 1 |

| 6 | 2 |

| 7 | 2 |

+----+-----+

**Output:**

+-----------------+

| ConsecutiveNums |

+-----------------+

| 1 |

+-----------------+

**Explanation:** 1 is the only number that appears consecutively for at least three times.

ANS:

QUESTION 4

Table: Sales

+-------------+-------+

| Column Name | Type |

+-------------+-------+

| sale\_id | int |

| product\_id | int |

| year | int |

| quantity | int |

| price | int |

+-------------+-------+

(sale\_id, year) is the primary key of this table.

product\_id is a foreign key to Product table.

Each row of this table shows a sale on the product product\_id in a certain year.

Note that the price is per unit.

Table: Product

+--------------+---------+

| Column Name | Type |

+--------------+---------+

| product\_id | int |

| product\_name | varchar |

+--------------+---------+

product\_id is the primary key of this table.

Each row of this table indicates the product name of each product.

Write an SQL query that selects the **product id**, **year**, **quantity**, and **price** for the **first year** of every product sold.

Return the resulting table in **any order**.

The query result format is in the following example.

**Example 1:**

**Input:**

Sales table:

+---------+------------+------+----------+-------+

| sale\_id | product\_id | year | quantity | price |

+---------+------------+------+----------+-------+

| 1 | 100 | 2008 | 10 | 5000 |

| 2 | 100 | 2009 | 12 | 5000 |

| 7 | 200 | 2011 | 15 | 9000 |

+---------+------------+------+----------+-------+

Product table:

+------------+--------------+

| product\_id | product\_name |

+------------+--------------+

| 100 | Nokia |

| 200 | Apple |

| 300 | Samsung |

+------------+--------------+

**Output:**

+------------+------------+----------+-------+

| product\_id | first\_year | quantity | price |

+------------+------------+----------+-------+

| 100 | 2008 | 10 | 5000 |

| 200 | 2011 | 15 | 9000 |

+------------+------------+----------+-------+

ANS: