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.

Example 1:

```
Input:
Person table:
+----------+----------+-----------+
| personId | lastName | firstName |
+----------+----------+-----------+
| 1        | Wang     | Allen     |
| 2        | Alice    | Bob       |
+----------+----------+-----------+
Address table:
+-----------+----------+---------------+------------+
| addressId | personId | city          | state      |
```

```
+-----------+----------+---------------+------------+
| 1         | 2        | New York City | New York   |
| 2         | 3        | Leetcode      | California |
+-----------+----------+---------------+------------+
Output:
+-----------+----------+---------------+----------+
| firstName | lastName | city          | state    |
+-----------+----------+---------------+----------+
| Allen     | Wang     | Null          | Null     |
| Bob       | Alice    | New York City | New York |
+-----------+----------+---------------+----------+
Explanation:
There is no address in the address table for the personId = 1 so we return null in
their city and state.
addressId = 1 contains information about the address of personId = 2.
```

True
Table: `Person`

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| id          | int     |
| email       | varchar |
+-------------+---------+
id is the primary key column for this table.
Each row of this table contains an email. The emails will not contain uppercase
letters.
```

Write an SQL query to delete all the duplicate emails, keeping only one unique email with the smallest `id`. Note that you are supposed to write a `DELETE` statement and not a `SELECT` one.

After running your script, the answer shown is the `Person` table. The driver will first compile and run your piece of code and then show the `Person` table. The final order of the `Person` table does not matter.

The query result format is in the following example.

Example 1:**

```
Input:
Person table:
+----+------------------+
| id | email            |
+----+------------------+
| 1  | john@example.com |
| 2  | bob@example.com  |
| 3  | john@example.com |
+----+------------------+
Output:
```

```
+----+------------------+
| id | email            |
+----+------------------+
| 1  | john@example.com |
| 2  | bob@example.com  |
+----+------------------+
```

Explanation: john@example.com is repeated two times. We keep the row with the smallest Id = 1.
