**4. Least Earning Locations**

A ride hailing company has their DB structured in 3 major tables as described in *SCHEMA* below.

Write a query to fetch the city names along with earnings from each city. 'Earnings' are calculated as the sum of fares of all the rides taken in that city. The output should be structured as: cities.name earnings

The output is sorted ascending by earnings, then ascending by the city name.

Schema

There are 3 tables: *CITIES*, *USERS*, and *RIDES*.

|  |  |  |
| --- | --- | --- |
| **CITIES** | | |
| **Name** | **Type** | **Description** |
| id | String | The assigned ID to the city presented as 32 character UUID. |
| name | String | The name of the city. |

|  |  |  |
| --- | --- | --- |
| **USERS** | | |
| **Name** | **Type** | **Description** |
| id | String | The assigned ID to the user presented as 32 character UUID. |
| city\_id | String | The id of the city in which this user resides. |
| name | String | The name of the user. |
| email | String | The email of the user. |

|  |  |  |
| --- | --- | --- |
| **RIDES** | | |
| **Name** | **Type** | **Description** |
| id | String | The assigned ID to the ride presented as 32 character UUID. |
| user\_id | String | The id of the user who took this ride. |
| distance | Integer | The travelled distance in this ride. |
| fare | Integer | The fare of this ride. |

Sample Data Tables

|  |  |
| --- | --- |
| **CITIES** | |
| **id** | **name** |
| 1 | Cooktown |
| 2 | South Suzanne |

|  |  |  |  |
| --- | --- | --- | --- |
| **USERS** | | | |
| **id** | **city\_id** | **name** | **email** |
| 1 | 2 | Robert Delgado | robertdelgado@hotmail.com |
| 2 | 2 | Thomas Williams | thomaswilliams@bradley.org |
| 3 | 1 | Michele Peterson | michelepeterson@hotmail.com |
| 4 | 1 | Bill Wheeler | billwheeler@gmail.com |
| 5 | 1 | David Lloyd | davidlloyd@gmail.com |
| 6 | 1 | Morgan Powers | morganpowers@hansen.biz |

|  |  |  |  |
| --- | --- | --- | --- |
| **RIDES** | | | |
| **id** | **user\_id** | **distance** | **fare** |
| 1 | 1 | 21 | 200 |
| 2 | 3 | 6 | 55 |
| 3 | 2 | 30 | 230 |
| 4 | 2 | 16 | 125 |
| 5 | 2 | 11 | 110 |
| 6 | 6 | 30 | 285 |
| 7 | 3 | 18 | 170 |
| 8 | 1 | 6 | 50 |
| 9 | 2 | 4 | 40 |
| 10 | 1 | 10 | 90 |
| 11 | 6 | 11 | 95 |
| 12 | 5 | 16 | 140 |
| 13 | 3 | 24 | 220 |
| 14 | 6 | 17 | 160 |
| 15 | 2 | 23 | 205 |
| 16 | 3 | 11 | 90 |
| 17 | 6 | 5 | 50 |
| 18 | 3 | 19 | 180 |
| 19 | 5 | 22 | 205 |
| 20 | 4 | 6 | 60 |

South Suzanne 1050

Cooktown 1710

**Explanation**

1. In the city *South Suzanne*, there are 2 users and the total fare of rides taken by those users is 1050.
2. In the city *Cooktown*, there are 4 users and the total fare of rides taken by those users is 1710.

Language: MS SQL ServerAutocomplete not supported

More

7

8

9

10

11

12

13

14

15

\*/

select c.name,SUM(r.fare)

from CITIES c

LEFT JOIN USERS u ON c.id=u.city\_id

LEFT JOIN RIDES r ON u.id=r.user\_id

GROUP BY c.name

ORDER BY SUM(r.fare) ASC

go



Line: 9 Col: 13

Test Results

Run Query

Submit

**Compiled successfully.Correct answer.**

**Test case 0**

Your Output (stdout)

* **Scottshire 81155**
* **North Danielfurt 100505**
* **Taylormouth 102090**
* **South Suzanne 104540**
* **Arnoldmouth 109040**
* **Cooktown 110000**

Expected Output

Download

* **Scottshire 81155**
* **North Danielfurt 100505**
* **Taylormouth 102090**
* **South Suzanne 104540**
* **Arnoldmouth 109040**
* **Cooktown 110000**

holes