```
:) CREATE TABLE hw7_table
^I(
^I^Iuser_id UInt64,
^I^Iaction String,
^I^Iexpense UInt64
^I)
^IENGINE = MergeTree()
^IPRIMARY KEY (user_id)
^IORDER BY (user_id);

CREATE TABLE hw7_table
(
    `user_id` UInt64,
    `action` String,
    `expense` UInt64
)
ENGINE = MergeTree
PRIMARY KEY user_id
ORDER BY user_id

Query id: 1a4f0cde-1e6d-4e5b-907d-bc86537387d5

Ok.
```

```
:) INSERT INTO hw7_table (user_id, action, expense)
^ISELECT
^I^I-- Генерируем 10 уникальных пользователей (1-10)
^I^I(number % 10) + 1 AS user_id,
^I^I
^I^I-- Низкокардинальное поле action: всего 4 возможных значения
^I^IarrayElement(['view', 'click', 'purchase', 'refund'], (number %
^I^I
^I^I-- Случайные расходы: 3 категории (низкие, средние, высокие)
^I^ImultiIf(
^I^I^Iaction = 'view', 10 + (rand() % 20),          -- 10-30
^I^I^Iaction = 'click', 50 + (rand() % 50),         -- 50-100
^I^I^Iaction = 'purchase', 500 + (rand() % 500),    -- 500-1000
^I^I^Iaction = 'refund', 200 + (rand() % 200),      -- 200-400
^I^I^I0
^I^I) AS expense
^IFROM numbers(30)  -- 30 строк
^IORDER BY user_id, action;

INSERT INTO hw7_table (user_id, action, expense) SELECT
    (number % 10) + 1 AS user_id,
    arrayElement(['view', 'click', 'purchase', 'refund'], (number %
    multiIf(action = 'view', 10 + (rand() % 20), action = 'click', 5
) AS expense
FROM numbers(30)
ORDER BY
    user_id ASC,
    action ASC

Query id: 028f7ef6-63f8-4e7e-a57e-ab9e67d2ce23

Ok.
```

```
CREATE DICTIONARY user_emails_dict
(
    `user_id` UInt64,
    `email` String
)
PRIMARY KEY user_id
SOURCE(CLICKHOUSE(QUERY 'SELECT user_id, email FROM file('/var/lib/clickhouse/user_files/emails.csv', 'CSVWithNames', 'user_id UInt64, email String')'))
LIFETIME(MIN 0 MAX 3600)
LAYOUT(HASHED())

Query id: 5599da15-60d2-4957-9bda-74908164c011

Ok.

0 rows in set. Elapsed: 0.008 sec.
```

```
SELECT
    dictGet(user_emails_dict, 'email', user_id),
    action,
    expense,
    SUM(expense) OVER (PARTITION BY action ORDER BY dictGet(user_emails_dict, 'email', user_id) ASC)
FROM hw7_table
ORDER BY dictGet(user_emails_dict, 'email', user_id) ASC

Query id: b5212084-4043-48d4-9392-e951d8e0d65d
```

	dictGet('user_…il', user_id)	action	expense	SUM(expense)…er_id) ASC)
1.		purchase	649	649
2.		view	23	41
3.		view	18	41
4.	alex.johnson@gmail.com	view	24	80
5.	alex.johnson@gmail.com	view	15	80
6.	alex.johnson@gmail.com	purchase	940	1589
7.	amelia.davis@aol.com	click	65	118
8.	amelia.davis@aol.com	refund	371	371
9.	amelia.davis@aol.com	click	53	118
10.	emma.watson@outlook.com	refund	228	819
11.	emma.watson@outlook.com	refund	220	819
12.	emma.watson@outlook.com	click	76	194
13.	ivan.petrov@mail.ru	purchase	515	2658
14.	ivan.petrov@mail.ru	purchase	554	2658
15.	ivan.petrov@mail.ru	view	10	90
16.	li.wei@qq.com	purchase	625	3283
17.	li.wei@qq.com	view	27	132
18.	li.wei@qq.com	view	15	132
19.	maria.garcia@yahoo.com	refund	285	1104
20.	maria.garcia@yahoo.com	click	98	347
21.	maria.garcia@yahoo.com	click	55	347
22.	noah.williams@protonmail.com	click	56	403
23.	noah.williams@protonmail.com	refund	298	1671
24.	noah.williams@protonmail.com	refund	269	1671
25.	olivia.brown@icloud.com	purchase	809	4892
26.	olivia.brown@icloud.com	purchase	800	4892
27.	olivia.brown@icloud.com	view	29	161
28.	sophie.martin@hotmail.com	click	50	509
29.	sophie.martin@hotmail.com	click	56	509
30.	sophie.martin@hotmail.com	refund	270	1941

```
30 rows in set. Elapsed: 0.004 sec.

:)
```