1. Business Expansion

1. Business Expansion

As part of business expansion efforts at a company, your help is needed to find all pairs of customers and agents who have been in contact more than once. For each such pair, display the user id, first name, and last name, and the customer id, name, and the number of their contacts. Order the result by user id ascending.

▾ Schema

There are 3 tables: customer, user_account, contact.

customer

Name	Type	Description
id	int	This is a primary key
customer_name	varchar(255)	Name of the customer
city_id	int	A unique id for the city in which the customer resides
customer_address	varchar(255)	Customer's address
contact_person	varchar(255)	Person of contact. Can be null.
email	varchar(128)	Email address of the customer
phone	varchar(128)	Phone number for the customer
is_active	int	Boolean to denote if the customer is active

user_account

Name	Type	Description
id	int	This is a primary key
first_name	varchar(64)	First name of the user
last_name	varchar(64)	Last name of the user
user_name	varchar(128)	user name for the user
password	varchar(255)	password of the account
email	varchar(128)	email address of the user
phone	varchar(128)	contact number for the user. Can be null.
is_active	int	Boolean to denote if the user_account is active

contact

Name	Type	Description
id	int	This is a primary key
user_account_id	int	Foreign key referencing user_account.id
customer_id	int	Foreign key referencing customer.id
contact_type_id	int	Type of the contact
contact_outcome_id	int	Can be null
additional_comment	varchar(255)	Can be null
initiated_by_customer	int	Boolean
initiated_by_user	int	Boolean

▸ Sample Data Tables

Language: MySQL | Environment | Autocomplete Ready

```
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT ua.id, ua.first_name, ua.last_name, cu.id, cu.customer_name, COUNT(cu.id)
FROM customer cu, user_account ua, contact c
WHERE cu.id = c.customer_id AND ua.id = c.user_account_id
GROUP BY ua.id, ua.first_name, ua.last_name, cu.id, cu.customer_name
HAVING COUNT(cu.id) > 1;
```

Line: 9 Col: 25

Test Results | Run Query | Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+----+------------+-----------+----+--------------------+----------
| id | first_name | last_name | id | customer_name      | COUNT(cu
+----+------------+-----------+----+--------------------+----------
|  4 | Alex       | Ferguson  |  7 | Natural Cosmetics  |
+----+------------+-----------+----+--------------------+----------
```

Expected Output | Download

```
4 Alex Ferguson 7 Natural Cosmetics 2
```

Language: MySQL | Environment | Autocomplete Ready

```
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT ua.id, ua.first_name, ua.last_name, cu.id, cu.customer_name, COUNT(cu.id)
FROM customer cu, user_account ua, contact c
WHERE cu.id = c.customer_id AND ua.id = c.user_account_id
GROUP BY ua.id, ua.first_name, ua.last_name, cu.id, cu.customer_name
HAVING COUNT(cu.id) > 1;
```

Line: 9 Col: 25

Test Results | Run Query | Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+----+------------+-----------+----+--------------------+----------
| id | first_name | last_name | id | customer_name      | COUNT(cu
+----+------------+-----------+----+--------------------+----------
|  4 | Alex       | Ferguson  |  7 | Natural Cosmetics  |
+----+------------+-----------+----+--------------------+----------
```

Expected Output | Download

```
4 Alex Ferguson 7 Natural Cosmetics 2
```

▼ Sample Data Tables

customer

id	customer_name	city_id	customer_address	contact_person	email
1	Drogerie Wien	1	Deckergasse 15A	Emil Steinbach	emil@drogeriewie
2	Cosmetics Store	4	Watling Street 347	Jeremy Corbyn	jeremy@c-store.
3	Kosmetikstudio	3	Rothenbaumchaussee 53	Willy Brandt	willy@kosmetikstud
4	Neue Kosmetik	1	Karlsplatz 2	NULL	info@neuekosmeti
5	Bio Kosmetik	2	Motzstraße 23	Clara Zetkin	clara@biokosmeti
6	K-Wien	1	Kärntner Straße 204	Maria Rauch-Kallat	maria@kwien.c
7	Natural Cosmetics	4	Clerkenwell Road 14B	Glenda Jackson	glena.j@natural-cosm
8	Kosmetik Plus	2	Unter den Linden 1	Angela Merkel	angela@k-plus.c
9	New Line Cosmetics	4	Devonshire Street 92	Oliver Cromwell	oliver@nlc.or

user_account

id	first_name	last_name	user_name	password	email
1	Jürgen	Klopp	jklopp	d0406ee29d848474a1897058df4634ad	jklopp@gmail.con
2	José	Mourinho	jmourinho	64bd7f38bef780050096bfb872084ac7	jmourinho@gmail.c
3	Josep	Guardiola	jpuardiola	98a87d2631b766915c9d1bf584c3fd7b	jpuardiola@gmail.c
4	Alex	Ferguson	aferguson	bda97a3a31b44206e6bc3d5f40f6023e	aferguson@gmail.c

contact

id	user_account_id	customer_id	contact_type_id	contact_outcome_id	additional_commer
1	4	7	1	4	customer agreed to our
2	1	2	1	1	don't want our produ
3	2	9	1	4	interested in our offe
4	3	1	1	2	call him tomorrow

```
Language MySQL   Environment   Autocomplete Ready
1  /*
2  Enter your query below.
3  Please append a semicolon ";" at the end of the query
4  */
5  SELECT ua.id, ua.first_name, ua.last_name, cu.id, cu.customer_name, COUNT(cu.id)
6  FROM customer cu, user_account ua, contact c
7  WHERE cu.id = c.customer_id AND ua.id = c.user_account_id
8  GROUP BY ua.id, ua.first_name, ua.last_name, cu.id, cu.customer_name
9  HAVING COUNT(cu.id) > 1;
Line: 9 Col: 25
```

Test Results — Run Query — Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+----+------------+-----------+----+--------------------+----------
| id | first_name | last_name | id | customer_name      | COUNT(cu
+----+------------+-----------+----+--------------------+----------
| 4 | Alex       | Ferguson  | 7 | Natural Cosmetics |
+----+------------+-----------+----+--------------------+----------
```

Expected Output — Download

```
4 Alex Ferguson 7 Natural Cosmetics 2
```

2m 30s left

ALL

oad 14B | Glenda Jackson | glena.j@natural-cosmetics.com | 093555123 | 1
inden 1 | Angela Merkel | angela@k-plus.com | 094727727 | 1
treet 92 | Oliver Cromwell | oliver@nlc.org | 093202404 | 0

user_account

password	email	phone	is_active
5ee29d848474a1897058df4634ad	jklopp@gmail.com	0933336777	1
7f38bef780050096bfb872084ac7	jmourinho@gmail.com	093825125	1
7d2631b766915c9d1bf584c3fd7b	jpuardiola@gmail.com	NULL	1
7a3a31b44206e6bc3d5f40f6023e	aferguson@gmail.com	093925645	0

contact

pe_id	contact_outcome_id	additional_comment	initiated_by_customer	initiated_by_user
	4	customer agreed to our offer	0	1
	1	don't want our products	0	1
	4	interested in our offer	0	1
	2	call him tomorrow	0	1
	4	purchased our products	0	1
	4	customer agreed to our offer	0	1
	4	loves our products	0	1
	4	customer agreed to our offer	0	1
	3	call him next month	0	1
	2	call him next month	0	1
	3	visit him next week	0	1

er_account_id, customer_id) who have been in contact
h other twice.

Language MySQL   Environment   Autocomplete Ready

```
1  /*
2  Enter your query below.
3  Please append a semicolon ";" at the end of the query
4  */
5  SELECT ua.id, ua.first_name, ua.last_name, cu.id, cu.customer_name, COUNT(cu.id)
6  FROM customer cu, user_account ua, contact c
7  WHERE cu.id = c.customer_id AND ua.id = c.user_account_id
8  GROUP BY ua.id, ua.first_name, ua.last_name, cu.id, cu.customer_name
9  HAVING COUNT(cu.id) > 1;
Line: 9 Col: 25
```

Test Results — Run Query — Submit

Compiled successfully. Correct answer.

Test case 0

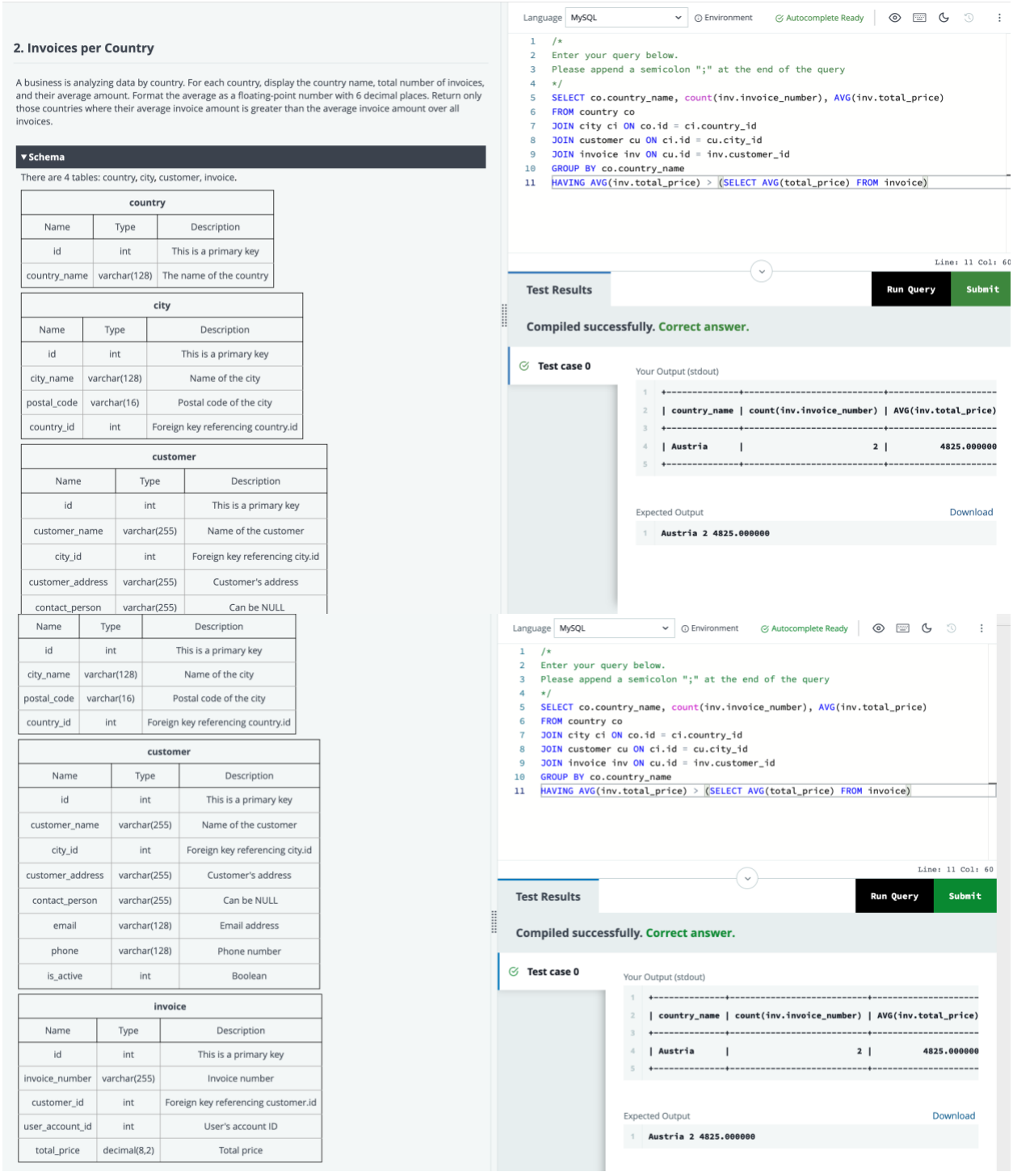
Your Output (stdout)

```
+----+------------+-----------+----+--------------------+----------
| id | first_name | last_name | id | customer_name      | COUNT(cu
+----+------------+-----------+----+--------------------+----------
| 4 | Alex       | Ferguson  | 7 | Natural Cosmetics |
+----+------------+-----------+----+--------------------+----------
```

Expected Output — Download

```
4 Alex Ferguson 7 Natural Cosmetics 2
```

2. Invoice Per Country



2. Invoices per Country

A business is analyzing data by country. For each country, display the country name, total number of invoices, and their average amount. Format the average as a floating-point number with 6 decimal places. Return only those countries where their average invoice amount is greater than the average invoice amount over all invoices.

▾ Schema

There are 4 tables: country, city, customer, invoice.

country

Name	Type	Description
id	int	This is a primary key
country_name	varchar(128)	The name of the country

city

Name	Type	Description
id	int	This is a primary key
city_name	varchar(128)	Name of the city
postal_code	varchar(16)	Postal code of the city
country_id	int	Foreign key referencing country.id

customer

Name	Type	Description
id	int	This is a primary key
customer_name	varchar(255)	Name of the customer
city_id	int	Foreign key referencing city.id
customer_address	varchar(255)	Customer's address
contact_person	varchar(255)	Can be NULL

Name	Type	Description
id	int	This is a primary key
city_name	varchar(128)	Name of the city
postal_code	varchar(16)	Postal code of the city
country_id	int	Foreign key referencing country.id

customer

Name	Type	Description
id	int	This is a primary key
customer_name	varchar(255)	Name of the customer
city_id	int	Foreign key referencing city.id
customer_address	varchar(255)	Customer's address
contact_person	varchar(255)	Can be NULL
email	varchar(128)	Email address
phone	varchar(128)	Phone number
is_active	int	Boolean

invoice

Name	Type	Description
id	int	This is a primary key
invoice_number	varchar(255)	Invoice number
customer_id	int	Foreign key referencing customer.id
user_account_id	int	User's account ID
total_price	decimal(8,2)	Total price

Language: MySQL | Environment | Autocomplete Ready

```
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT co.country_name, count(inv.invoice_number), AVG(inv.total_price)
FROM country co
JOIN city ci ON co.id = ci.country_id
JOIN customer cu ON ci.id = cu.city_id
JOIN invoice inv ON cu.id = inv.customer_id
GROUP BY co.country_name
HAVING AVG(inv.total_price) > (SELECT AVG(total_price) FROM invoice)
```

Line: 11 Col: 60

Test Results | Run Query | Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+--------------+----------------------------+---------------------
| country_name | count(inv.invoice_number) | AVG(inv.total_price)
+--------------+----------------------------+---------------------
| Austria      |                          2 |          4825.000000
+--------------+----------------------------+---------------------
```

Expected Output | Download

```
Austria 2 4825.000000
```


Language: MySQL | Environment | Autocomplete Ready

```
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT co.country_name, count(inv.invoice_number), AVG(inv.total_price)
FROM country co
JOIN city ci ON co.id = ci.country_id
JOIN customer cu ON ci.id = cu.city_id
JOIN invoice inv ON cu.id = inv.customer_id
GROUP BY co.country_name
HAVING AVG(inv.total_price) > (SELECT AVG(total_price) FROM invoice)
```

Line: 11 Col: 60

Test Results | Run Query | Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+--------------+----------------------------+---------------------
| country_name | count(inv.invoice_number) | AVG(inv.total_price)
+--------------+----------------------------+---------------------
| Austria      |                          2 |          4825.000000
+--------------+----------------------------+---------------------
```

Expected Output | Download

```
Austria 2 4825.000000
```

▾ Sample Data Tables

country

id	country_name
1	Austria
2	Germany
3	United Kingdom

city

id	city_name	postal_code	country_id
1	Wien	1010	1
2	Berlin	10115	2
3	Hamburg	20095	2
4	London	EC4V 4AD	3

customer

id	customer_name	city_id	customer_address	contact_person	email
1	Drogerie Wien	1	Deckergasse 15A	Emil Steinbach	emil@drogeriewie
2	Cosmetics Store	4	Watling Street 347	Jeremy Corbyn	jeremy@c-store.
3	Kosmetikstudio	3	Rothenbaumchaussee 53	Willy Brandt	willy@kosmetikstud
4	Neue Kosmetik	1	Karlsplatz 2	NULL	info@neuekosmeti
5	Bio Kosmetik	2	Motzstraße 23	Clara Zetkin	clara@biokosmeti
6	K-Wien	1	Kärntner Straße 204	Maria Rauch-Kallat	maria@kwien.c
7	Natural Cosmetics	4	Clerkenwell Road 14B	Glenda Jackson	glena.j@natural-cosm
8	Kosmetik Plus	2	Unter den Linden 1	Angela Merkel	angela@k-plus.c
9	New Line Cosmetics	4	Devonshire Street 92	Oliver Cromwell	oliver@nlc.or

invoice

4	London	EC4V 4AD	3

customer

id	customer_name	city_id	customer_address	contact_person	email
1	Drogerie Wien	1	Deckergasse 15A	Emil Steinbach	emil@drogeriewie
2	Cosmetics Store	4	Watling Street 347	Jeremy Corbyn	jeremy@c-store.
3	Kosmetikstudio	3	Rothenbaumchaussee 53	Willy Brandt	willy@kosmetikstud
4	Neue Kosmetik	1	Karlsplatz 2	NULL	info@neuekosmeti
5	Bio Kosmetik	2	Motzstraße 23	Clara Zetkin	clara@biokosmeti
6	K-Wien	1	Kärntner Straße 204	Maria Rauch-Kallat	maria@kwien.c
7	Natural Cosmetics	4	Clerkenwell Road 14B	Glenda Jackson	glena.j@natural-cosm
8	Kosmetik Plus	2	Unter den Linden 1	Angela Merkel	angela@k-plus.c
9	New Line Cosmetics	4	Devonshire Street 92	Oliver Cromwell	oliver@nlc.or

invoice

id	invoice_number	customer_id	user_account_id	total_price
1	in_25181b07ba800c8d2fc967fe991807d9	7	4	1436
2	8fba0000fd456b27502b9f81e9d52481	9	2	1000
3	3b6638118246b6bcfd3dfcd9be487599	3	2	360
4	dfe7f0a01a682196cac0120a9adbb550	5	2	1675
5	2a24cc2ad4440d698878a0a1a71f70fa	6	2	9500
6	cbd304872ca6257716bcab8fc43204d7	4	2	150

The average invoice amount is 2353.5. The average invoice amount of Country with ids 1, 2, and 3 are 4825, 1017.5, and 1218 respectively. Hence, the only country to report is Austria.

Language MySQL | Environment | Autocomplete Ready

```
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT co.country_name, count(inv.invoice_number), AVG(inv.total_price)
FROM country co
JOIN city ci ON co.id = ci.country_id
JOIN customer cu ON ci.id = cu.city_id
JOIN invoice inv ON cu.id = inv.customer_id
GROUP BY co.country_name
HAVING AVG(inv.total_price) > (SELECT AVG(total_price) FROM invoice)
```

Line: 11 Col: 60

Test Results | Run Query | Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+--------------+----------------------------+---------------------
| country_name | count(inv.invoice_number) | AVG(inv.total_price)
+--------------+----------------------------+---------------------
| Austria      |                          2 |          4825.000000
+--------------+----------------------------+---------------------
```

Expected Output | Download

```
Austria 2 4825.000000
```

Language MySQL | Environment | Autocomplete Ready

```
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT co.country_name, count(inv.invoice_number), AVG(inv.total_price)
FROM country co
JOIN city ci ON co.id = ci.country_id
JOIN customer cu ON ci.id = cu.city_id
JOIN invoice inv ON cu.id = inv.customer_id
GROUP BY co.country_name
HAVING AVG(inv.total_price) > (SELECT AVG(total_price) FROM invoice)
```

Line: 11 Col: 60

Test Results | Run Query | Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+--------------+----------------------------+---------------------
| country_name | count(inv.invoice_number) | AVG(inv.total_price)
+--------------+----------------------------+---------------------
| Austria      |                          2 |          4825.000000
+--------------+----------------------------+---------------------
```

Expected Output | Download

```
Austria 2 4825.000000
```