Q.1: Assume you are given the table below on Uber transactions made by users. Write a query to obtain the third transaction of every user. Output the user id, spend and transaction date.

`transactions` Table:

Column Name	Type
user_id	integer
spend	decimal
transaction_date	timestamp

`transactions` Example Input:

user_id	spend	transaction_date
111	100.50	01/08/2022 12:00:00
111	55.00	01/10/2022 12:00:00
121	36.00	01/18/2022 12:00:00
145	24.99	01/26/2022 12:00:00
111	89.60	02/05/2022 12:00:00

Example Output:

user_id	spend	transaction_date
111	89.60	02/05/2022 12:00:00

Q.2: Assume you're given tables with information on Snapchat users, including their ages and time spent sending and opening snaps.

Write a query to obtain a breakdown of the time spent sending vs. opening snaps as a percentage of total time spent on these activities grouped by age group. Round the percentage to 2 decimal places in the output.

Notes:

- Calculate the following percentages:
  - time spent sending / (Time spent sending + Time spent opening)
  - Time spent opening / (Time spent sending + Time spent opening)
- To avoid integer division in percentages, multiply by 100.0 and not 100.

`activities` Table

Column Name	Type
activity_id	integer
user_id	integer
activity_type	string ('send', 'open', 'chat')
time_spent	float
activity_date	datetime

`activities` Example Input

activity_id	user_id	activity_type	time_spent	activity_date
7274	123	open	4.50	06/22/2022 12:00:00
2425	123	send	3.50	06/22/2022 12:00:00
1413	456	send	5.67	06/23/2022 12:00:00
1414	789	chat	11.00	06/25/2022 12:00:00
2536	456	open	3.00	06/25/2022 12:00:00

`age_breakdown` Table

Column Name	Type
user_id	integer
age_bucket	string ('21-25', '26-30', '31-25')

`age_breakdown` Example Input

user_id	age_bucket
123	31-35
456	26-30
789	21-25

Example Output

age_bucket	send_perc	open_perc
26-30	65.40	34.60
31-35	43.75	56.25

Q.3: Given a table of tweet data over a specified time period, calculate the 3-day rolling average of tweets for each user. Output the user ID, tweet date, and rolling averages rounded to 2 decimal places.
Notes:

- A rolling average, also known as a moving average or running mean, is a time-series technique that examines trends in data over a specified period of time.
- In this case, we want to determine how the tweet count for each user changes over a 3-day period.

`tweets` Table:

Column Name	Type
user_id	integer
tweet_date	timestamp
tweet_count	integer

`tweets` Example Input:

user_id	tweet_date	tweet_count
111	06/01/2022 00:00:00	2
111	06/02/2022 00:00:00	1
111	06/03/2022 00:00:00	3
111	06/04/2022 00:00:00	4
111	06/05/2022 00:00:00	5

Example Output:

user_id	tweet_date	rolling_avg_3d
111	06/01/2022 00:00:00	2.00
111	06/02/2022 00:00:00	1.50
111	06/03/2022 00:00:00	2.00
111	06/04/2022 00:00:00	2.67
111	06/05/2022 00:00:00	4.00

Q.4 : Assume you're given a table with information on Amazon customers and their spending on products in different categories, write a query to identify the top two highest-grossing products within each category in the year 2022. The output should include the category, product, and total spend.

`product_spend` Table:

Column Name	Type
category	string
product	string
user_id	integer
spend	decimal
transaction_date	timestamp

`product_spend` Example Input:

category	product	user_id	spend	transaction_date
appliance	refrigerator	165	246.00	12/26/2021 12:00:00
appliance	refrigerator	123	299.99	03/02/2022 12:00:00
appliance	washing machine	123	219.80	03/02/2022 12:00:00
electronics	vacuum	178	152.00	04/05/2022 12:00:00
electronics	wireless headset	156	249.90	07/08/2022 12:00:00
electronics	vacuum	145	189.00	07/15/2022 12:00:00

Example Output:

category	product	total_spend
appliance	refrigerator	299.99
appliance	washing machine	219.80
electronics	vacuum	341.00
electronics	wireless headset	249.90

Q.5: Assume there are three Spotify tables containing information about the artists, songs, and music charts. Write a query to find the top 5 artists whose songs appear most frequently in the Top 10 of the `global_song_rank` table.

Display the top 5 artist names in ascending order, along with their song appearance ranking. Note that if two artists have the same number of song appearances, they should have the same ranking, and the rank numbers should be continuous (i.e. 1, 2, 2, 3, 4, 5).

For instance, if Ed Sheeran appears in the Top 10 five times and Bad Bunning four times, Ed Sheeran should be ranked 1st, and Bad Bunny should be ranked 2nd.

`artists` Table:

Column Name	Type
artist_id	integer
artist_name	varchar

`artists` Example Input:

artist_id	artist_name
101	Ed Sheeran
120	Drake

`songs` Table:

Column Name	Type
song_id	integer
artist_id	integer

`songs` Example Input:

song_id	artist_id
45202	101
19960	120

`global_song_rank` Table:

Column Name	Type
day	integer (1-52)
song_id	integer

rank	integer (1-1,000,000)

`global_song_rank` Example Input:

day	song_id	rank
1	45202	5
3	45202	2
1	19960	3
9	19960	15

Example Output:

artist_name	artist_rank
Ed Sheeran	1
Drake	2

Q.6: New TikTok users sign up with their emails. They confirmed their signup by replying to the text confirmation to activate their accounts. Users may receive multiple text messages for account confirmation until they have confirmed their new account.
A senior analyst is interested to know the activation rate of specified users in the `emails` table. Write a query to find the activation rate. Round the percentage to 2 decimal places.

Definitions:

- **`The emails` table contains the information of user signup details.**
- **`The text` table contains the users' activation information.**

Assumptions:

- **The analyst is interested in the activation rate of specific users in the `emails` table, which may not include all users that could potentially be found in the `texts` table.**
- **For example, user 123 in the `emails` table may not be in the `texts` table and vice versa.**

Effective April 4th 2023, we added an assumption to the question to provide additional clarity.

`emails` Table:

Column Name	Type
email_id	integer
user_id	integer
signup_date	datetime

`emails` Example Input:

email_id	user_id	signup_date
125	7771	06/14/2022 00:00:00
236	6950	07/01/2022 00:00:00
433	1052	07/09/2022 00:00:00

`texts` Table:

Column Name	Type
text_id	integer
email_id	integer
signup_action	varchar

`texts` Example Input:

text_id	email_id	signup_action
6878	125	Confirmed
6920	236	Not Confirmed
6994	236	Confirmed

'Confirmed' in `signup_action` means the user has activated their account and successfully completed the signup process.

Example Output:

confirm_rate
0.67

Q.7: A Microsoft Azure Supercloud customer is a company which buys at least 1 product from each product category.
Write a query to report the company ID which is a Supercloud customer.

As of 5 Dec 2022, data in the `customer_contracts` and `products` tables were updated.

`customer_contracts` Table:

Column Name	Type
customer_id	integer
product_id	integer
amount	integer

`customer_contracts` Example Input:

customer_id	product_id	amount
1	1	1000
1	3	2000
1	5	1500
2	2	3000
2	6	2000

`products` Table:

Column Name	Type
product_id	integer
product_category	string
product_name	string

`products` Example Input:

product_id	product_category	product_name
1	Analytics	Azure Databricks
2	Analytics	Azure Stream Analytics

4	Containers	Azure Kubernetes Service
5	Containers	Azure Service Fabric
6	Compute	Virtual Machines
7	Compute	Azure Functions

Example Output:

customer_id
1

Q.8: Assume you're given a table with measurement values obtained from a Google sensor over multiple days with measurements taken multiple times within each day. Write a query to calculate the sum of odd-numbered and even-numbered measurements separately for a particular day and display the results in two different columns. Refer to the Example Output below for the desired format.

Definition:

- Within a day, measurements taken at 1st, 3rd, and 5th times are considered odd-numbered measurements, and measurements taken at 2nd, 4th, and 6th times are considered even-numbered measurements.

`measurements` Table:

Column Name	Type
measurement_id	integer
measurement_value	decimal
measurement_time	datetime

`measurements` Example Input:

measurement_id	measurement_value	measurement_time
131233	1109.51	07/10/2022 09:00:00
135211	1662.74	07/10/2022 11:00:00

523542	1246.24	07/10/2022 13:15:00
143562	1124.50	07/11/2022 15:00:00
346462	1234.14	07/11/2022 16:45:00

Example Output:

measurement_day	odd_sum	even_sum
07/10/2022 00:00:00	2355.75	1662.74
07/11/2022 00:00:00	1124.50	1234.14

Q.9: Assume you are given the table on Walmart user transactions. Based on a user's most recent transaction date, write a query to obtain the users and the number of products bought.
Output the user's most recent transaction date, user ID and the number of products sorted by the transaction date in chronological order.

P.S. As of 10 Nov 2022, the official solution was changed from output of the transaction date, number of users and number of products to the current output.

`user_transactions` Table:

Column Name	Type
product_id	integer
user_id	integer
spend	decimal
transaction_date	timestamp

`user_transactions` Example Input:

product_id	user_id	spend	transaction_date
3673	123	68.90	07/08/2022 12:00:00
9623	123	274.10	07/08/2022 12:00:00
1467	115	19.90	07/08/2022 12:00:00
2513	159	25.00	07/08/2022 12:00:00

1452	159	74.50	07/10/2022 12:00:00

Example Output:

transaction_date	user_id	purchase_count
07/08/2022 12:00:00	115	1
07/08/2022 12:00:000	123	2
07/10/2022 12:00:00	159	1

Q.10: Given a table containing the item count for each order and the frequency of orders with that item count, write a query to determine the mode of the number of items purchased per order on Alibaba. If there are several item counts with the same frequency, you should sort them in ascending order.
Effective April 22nd, 2023, the problem statement and solution have been revised for enhanced clarity.

`items_per_order` Table:

Column Name	Type
item_count	integer
order_occurrences	integer

`items_per_order` Example Input:

item_count	order_occurrences
1	500
2	1000
3	800
4	1000

Example Output:

mode
2
4

Q.11: Your team at JPMorgan Chase is soon launching a new credit card. You are asked to estimate how many cards you'll issue in the first month.
Before you can answer this question, you want to first get some perspective on how well new credit card launches typically do in their first month.

Write a query that outputs the name of the credit card, and how many cards were issued in its launch month. The launch month is the earliest record in the `monthly_cards_issued` table for a given card. Order the results starting from the biggest issued amount.

`monthly_cards_issued` Table:

Column Name	Type
issue_month	integer
issue_year	integer
card_name	string
issued_amount	integer

`monthly_cards_issued` Example Input:

issue_month	issue_year	card_name	issued_amount
1	2021	Chase Sapphire Reserve	170000
2	2021	Chase Sapphire Reserve	175000
3	2021	Chase Sapphire Reserve	180000
3	2021	Chase Freedom Flex	65000
4	2021	Chase Freedom Flex	70000

Example Output:

card_name	issued_amount
Chase Sapphire Reserve	170000
Chase Freedom Flex	65000

`Q.12:` A phone call is considered an international call when the person calling is in a different country than the person receiving the call.
What percentage of phone calls are international? Round the result to 1 decimal.

Assumption:

- The `caller_id` in the phone_info table refers to both the caller and receiver.

`phone_calls` Table:

Column Name	Type
caller_id	integer
receiver_id	integer
call_time	timestamp

`phone_calls` Example Input:

caller_id	receiver_id	call_time
1	2	2022-07-04 10:13:49
1	5	2022-08-21 23:54:56
5	1	2022-05-13 17:24:06
5	6	2022-03-18 12:11:49

`phone_info` Table:

Column Name	Type
caller_id	integer
country_id	integer
network	integer
phone_number	string

`phone_info` Example Input:

caller_id	country_id	network	phone_number
1	US	Verizon	+1-212-897-1964
2	US	Verizon	+1-703-346-9529
3	US	Verizon	+1-650-828-4774
4	US	Verizon	+1-415-224-6663
5	IN	Vodafone	+91 7503-907302
6	IN	Vodafone	+91 2287-664895

Example Output:

international_calls_pct
50.0