

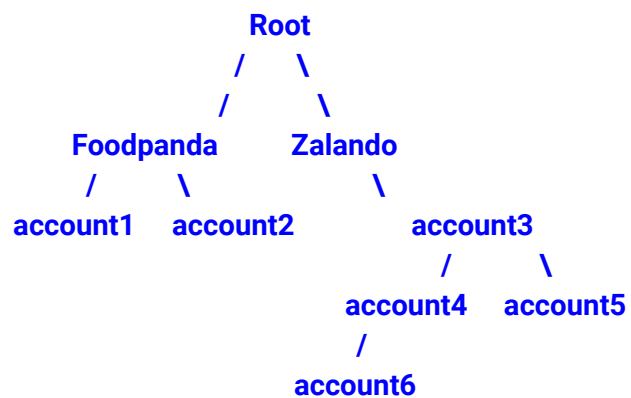
Please complete the exercises in python or SQL and submit runnable python files !
Disclosing of this exercises outside of this interview process is strictly forbidden.

Exercise 1:

Given a tree structure (not necessarily binary (!)) with three types nodes:

- **The root:** only node of the tree with no parents
- **Ventures:** all childrens of the root
- **Accounts:** all nodes which are not the root neither venture nodes

For example:



which is represented as a list of **Link** objects:

```
class Link(object):
    def __init__(self, parent_name, child_name):
        self.parent_name = parent_name
        self.child_name = child_name
```

for example the top left link could be expressed as:

```
>> top_left_link = Link("Root", "Foodpanda")
```

Write a function that given a list of links and an account name, finds the name of the venture it belongs. (Hint: ventures are always linked to the root)

The signature of this function should be:

```
def find_venture(list_of_links, account_name):
    #code
    return venture_name
```

example:

```
>> find_venture(list_of_links,"account5")  
"Zalando"
```

Exercise 2:

A time interval can be represented as:

```
>> t_interval = [ ( start_hr, start_minute ), ( end_hr, end_minute ) ]
```

where:

- start_hr in [0,23]
- start_minute in [0, 59]
- end_hr in [0,24]
- end_minute in [0, 59]

for example:

```
[(0,0),(24,0)] represents the entire day  
[(13,15),(13,30)] represents a quarter of an hour  
[(9,10), (8,10)] is an invalid time interval
```

Write and `__test__` a function that given two time intervals returns whether they overlap.

The signature of this function should be:

```
def have_overlap(t_interval1, t_interval2):  
    #code  
    return # boolean value true or false
```

example:

```
>> t1 = [(0,0),(24,0)]  
>> t2 = [(0,0),(12,0)]  
>> have_overlap(t1, t2)  
True
```

```
>> t1 = [(11,0), (24,0)]  
>> t2 = [(0,0), (10,0)]  
>> have_overlap(t1, t2)  
False
```

Exercise 3:

Given the below subset of Foodpanda schema, write executable SQL queries to answer the questions below. Please answer in a single query for each question and assume readonly access to the database (i.e. do not use CREATE TABLE).

1. For each of the cities 'Singapore' and 'Bangkok', calculate 90th percentile difference between Actual and Predicted ETA for all completed deliveries within the last 30 days.
2. A signup is defined as an event labeled 'sign_up_success' within the events table. For each city ('Singapore' and "Bangkok") and each day of the week, determine the percentage of signups in the first week of 2016 that resulted in completed delivery within 168 hours of the sign up date.

Table: **deliveries**

Column Name:	Datatype
id	INT
client_id	INT
rider_id	INT
requested_at	TIMESTAMP WITH TIMEZONE
predicted_eta	INT
actual_eta	INT
status	Enum('completed', 'cancelled_by_driver', 'cancelled_by_client')

Table: **cities**

Column name	Datatype
city_id	INT
city_name	STRING

Table: **events**

Column name	Datatype
device_id	INT
rider_id	INT
city_id	INT
event_name	Enum('sign_up_success', 'attempted_sign_up', 'sign_up_failure')
_ts	TIMESTAMP WITH TIMEZONE