

## Day 23 – Passengers in each bus

### Problem Statement:

Buses and passengers arrive at the LeetCode station. If a bus arrives at the station at a time  $t_{bus}$  and a passenger arrived at a time  $t_{passenger}$  where  $t_{passenger} \leq t_{bus}$  and the passenger did not catch any bus, the passenger will use that bus. In addition, each bus has a capacity. If at the moment the bus arrives at the station there are more passengers waiting than its capacity, only capacity passengers will use the bus.

- Write a solution to report the number of users that used each bus.
- Return the result table ordered by `bus_id` in ascending order.

### Input:

```
create table buses
(
    bus_id          int unique,
    arrival_time    int,
    capacity        int
);

drop table if exists Passengers;
create table Passengers
(
    passenger_id    int unique,
    arrival_time    int
);
```

Table: **Buses**

Column	Type
bus_id	int
arrival_time	int
capacity	int

`bus_id` contains unique values.

Each row of this table contains information about the arrival time of a bus at the LeetCode station and its capacity (the number of empty seats it has).

No two buses will arrive at the same time and all bus capacities will be positive integers.

Table: **Passengers**

Column	Type
passenger_id	int
arrival_time	int

passenger\_id contains unique values.

Each row of this table contains information about the arrival time of a passenger at the LeetCode station.

### Example 1:

#### Input:

Buses table:

bus_id	arrival_time	capacity
1	2	1
2	4	10
3	7	2

Passengers table:

passenger_id	arrival_time
11	1
12	1
13	5
14	6
15	7

#### Output:

bus_id	passengers_cnt
1	1
2	1
3	2

## SOLUTION

```
with cte
as(
    select row_number() over(order by arrival_time) as rn, bus_id,
    (select count(1) from passengers p where p.arrival_time <=
b.arrival_time) as total_passengers,
    capacity
    from buses b
),finalcte
as (
    select rn, bus_id, capacity, total_passengers,
case when capacity < total_passengers then capacity else
total_passengers end as onboarded_bus,

    case when capacity < total_passengers then capacity else
total_passengers end as total_onboarded

    from cte where rn=1

    union all

    select d.rn, d.bus_id, d.capacity, d.total_passengers,
    case when d.capacity < (d.total_passengers -
finalcte.total_onboarded) then d.capacity else (d.total_passengers -
finalcte.total_onboarded) end as onboarded_bus,

    finalcte.total_onboarded + case when d.capacity <
(d.total_passengers - finalcte.total_onboarded) then d.capacity else
(d.total_passengers - finalcte.total_onboarded) end as
total_onboarded

    from finalcte
    join cte d on d.rn = finalcte.rn+1
)
select bus_id, onboarded_bus as passengers_cnt
from finalcte
order by bus_id;
```

## OUTPUT

	bus_id	passengers_cnt
1	1	1
2	2	1
3	3	2