Sample data analyst questions

Question 1) Departments

Given a list of departments and their employees, compute the number of employees in each department and the sums of their salaries.

Task description:

You are given two tables, department and employee, with the following structure:

```
create table department (
dept_id integer not null,
dept_name varchar(30) not null,
dept_location varchar(30) not null,
unique(dept_id)
);
create table employee (
emp_id integer not null,
emp_name varchar(50) not null,
dept_id integer not null,
salary integer not null,
unique(emp_id)
);
```

Each record in the table department represents a department which might hire some employees. Each record in the table employee represents an employee who works for one of the departments from the table department. The salary of each employee is known.

Write an SQL query that returns a table comprising all the departments (dept_id) in the Table department that hire at least one employee, the number of people they employ, and the sum of salaries in each department. The table should be ordered by dept_id (in increasing order).

For example, given these tables:

```
INSERT INTO department (dept_id, dept_name, dept_location) VALUES
('10', 'Accounts', 'Delhi'),
('20', 'Marketing ', 'Delhi'),
('40', 'IT', 'Warsaw'),
('30', 'Production', 'Hyderabad'),
('50', 'Sales', 'Bengaluru');
INSERT INTO employee (emp_id, emp_name, dept_id, salary) VALUES
('1', 'Jojo', '20', '5000'),
('2', 'Popat Lal', '30', '15000'),
('3', 'Santa Singh', '40', '25000'),
```

('4', 'Banta Singh', '20', '7500'),
('5', 'Sohan Lal', '20', '15000'),
('6', 'Kk', '10', '12000'),
('7', 'Bob', '20', '35000'),
('8', 'John', '30', '25000'),
('9', 'Smith', '40', '5000');
Your query should return:
dept_id, count, sum_of_salary
10 1 12000
20 4 62500
30 2 40000
40 2 30000

Question 2) Elevator

Find the last person in the queue who will fit into an elevator without exceeding the weight restriction.
Task description:
You are presented with a table of data detailing people queueing for an elevator. All the people want to enter the elevator, but it can only hold a maximum weight of 1000 lbs. Your task is to find the last person who will fit in the elevator without exceeding the weight restriction.
Your are given this table line with the following structure and values:
create table line (
id int not null PRIMARY KEY,
name varchar(255) not null,
weight int not null,
turn int unique not null,
check (weight > 0)
);
The column turn represents the postigion of each person in line.
Positions are numbered from 1
INSERT INTO line (id, name, weight, turn) VALUES
('5', 'George Washington', '250', '1'),
('4', 'Thomas Jefferson', '175', '5'),
('3', 'John Adams', '350', '2'),
('6', 'Thomas Jefferson', '400', '3'),
('1', 'James Elephant', '500', '6'),
('2', 'Will Johnliams', '200', '4');
Your query should return a table containing exactly one record:
Thomas Jefferson
as the first three people will fit (in order: George Washington, John Adams, Thomas Jefferson). The sum of their weight is 250+350+400 = 1000.

Assume that:
the turn column contains all integers from 1 to N;
the first person always fits in the elevator;

Question 3) Countries

You are given the transaction history between companies from different countries.
Calculate the total import and export value of every country.
Task description:
You have details of some companies that trade internationally. You know the nationality of each company and you have a list of trades between the various companies. Your task is to generate a summary that consists of the sums of the value of goods imported and exported by every country. Note that when a company buys some goods, it contributes to its country's total import. When the company sells some goods, it contributes to its country's total export.
You are given two tables: companies and trades, with the following structure:

```
create table companies (
name varchar(30) not null,
country varchar(30) not null,
unique(name)
);
create table trades (
id integer not null,
seller varchar(30) not null,
buyer varchar(30) not null,
value integer not null,
unique(id)
);
```

A row in table companies contains the name of a company and the nationality of that company. A row in table trades contains the unique ID of the trade, the name of the selling company, the name of the buying company, and the value of the traded goods.
Write a SQL query that returns a table consisting of three columns: country, export, and import. The table should contain the sums of the values of the exported (sold to other countries) and imported (purchased from other countries) goods for each country. Each country should appear in this table. The result table should be sorted increasingly by country.
Data Analyst - Take home assignment 5
For example, given these tables:

```
INSERT INTO companies (name, country) VALUES
('Alice s.p.', 'Wonderland'),
('Y-zap', 'Wonderland'),
('Absolute', 'Mathlands'),
('Arcus t.g.', 'Mathlands'),
('Lil Mermaid', 'Underwater Kingdom'),
```

('None at all', 'Nothingland');
INSERT INTO trades (id, seller, buyer, value) VALUES
('20121107', 'Lil Mermaid', 'Alice s.p.', '10'),
('20123112', 'Arcus t.g.', 'Y-zap', '30'),
('20120125', 'Alice s.p.', 'Arcus t.g.', '100'),
('20120216', 'Lil Mermaid', 'Absolute', '30'),
('20120217', 'Lil Mermaid', 'Absolute', '50');
your query should return:
country exports imports
Mathlands 30 180
Nothingland 0 0
Underwater Kingdom 90 0
Wonderland 100 40
Assume that:
There is no trade between companies within a single country
Every company in the table trades also appears in the table companies
Every company appears in the table companies exactly once
Data Analyst - Take home assignment 6

Question 4) Buses

Given a table of buses and passengers, compute how many people will board each bus.
Task Description:
You are given two tables, buses and passengers, with the following structure:
create table buses (
id integer primary key,
origin varchar not null,
destination varchar not null,
time varchar not null
);
create table passengers (
id integer primary key,
origin varchar not null,
destination varchar not null,
time varchar not null
);
Each row in table buses contains information about a single bus's origin, destination, and time of departure (time).
Each row in table passengers describes a single passenger and contains information about the stating they're travelling from (origin), the stating they're travelling to (destination) and the time they will arrive at the departure stating (time).
Passengers will board the earliest possible bus that travels directly to their desired destination.
Passengers can still board a bus if it departs in the same minute that they arrive at the

station.
All passengers who are still at the station at 23:59 and don't board any of the 23:59 buses will leave the platform without boarding any bus.
You can assume that no two busses with the same origin and destination depart at the same time.
Time is represented as a string in the format HH:MM.

Write a SQL query that, for each bus, returns the number of passengers boarding it. For each bus you should provide its id (id) and the number of passengers on board (passengers_on_board). Rows should be ordered by the id column in ascending order.
For example, given these tables:
INSERT INTO buses (id, origin, destination, time) VALUES
('10', 'Warsaw', 'Berlin', '10:55'),
('20', 'Berlin', 'Paris', '06:20'),
('21', 'Berlin', 'Paris', '14:00'),
('22', 'Berlin', 'Paris', '21:40'),
('30', 'Paris', 'Madrid', '13:30');
INSERT INTO passengers (id, origin, destination, time) VALUES
('1', 'Paris', 'Madrid', '13:30'),
('2', 'Paris', 'Madrid', '13:31'),
('10', 'Warsaw', 'Paris', '10:00'),
('11', 'Warsaw', 'Berlin', '22:31'),
('40', 'Berlin', 'Paris', '06:15'),
('41', 'Berlin', 'Paris', '06:50'),
('42', 'Berlin', 'Paris', '07:12'),
('43', 'Berlin', 'Paris', '12:03'),
('44', 'Berlin', 'Paris', '20:00');
your query should return:
id passengers_on_board
10 0
20 1
21 3
22 1
30 1