### **Data Science Analyst (Software Product Ops Analyst) Assessment Please do not modify this document. Create a copy if you think necessary.**

### **Section 1. Problem- solving logic**

### **Problem 1**

### In Bright we use a PostgreSQL 13 database with about 120 tables. We use it for everything from sales to installation to customer service. Many of our core business processes rely on this database, and we also use data analytics to drive business decisions. In other words, we use it for OLTP and OLAP Of course, we can’t expect you to understand a data model of 120 tables - and countless more columns. So we have prepared an extremely simplified dump of Bright’s database as a [SQLite](https://www.sqlite.org/index.html) file as well as a .sql text file (zipped) which should be compatible with most database engines. This dump is sampled, anonymized, and simplified to be easier to use in this interview context. **Download the file of your choice here:** [Bright.sqlite](https://ops.thinkbright.mx/api/files/eaf67029-57f1-4f5a-83b6-d97e1484682d?download=true&name=bright.sqlite) [Bright.sql.zip](https://ops.thinkbright.mx/api/files/18e99b5d-ad39-44f0-9baa-74037079215b?download=true&name=bright.sql.zip) You **may use any database / analysis technology you are comfortable** with to analyze this data. That said, we recommend using SQLite directly. If you download the .sqlite file above, you can load it into any sqlite program (recommend <https://sqlitestudio.pl/> or <https://sqlitebrowser.org/>). Some things in SQLite are a bit different from PostgreSQL, Redshift, or MSSQL - but the basics are very similar. **Background information:**

### The database given includes only 3 tables, lead, event and attribute.

### Lead

### This table contains a row for every potential customer that Bright has seen in its lifetime. It only has 2 columns, lead\_id and created\_at. Which corresponds to the unique identifier of the lead and the moment in time it was uploaded to Bright’s system.

### Event

### This table contains a row for many kinds of events that can happen to a Bright lead.

### Attribute

### This table contains additional attributes about a lead, following the [EAV](https://en.wikipedia.org/wiki/Entity%E2%80%93attribute%E2%80%93value_model) data modeling pattern.

### **Conversion Rates:** Much of Bright’s sales analysis measures conversion rate. For example, of all the leads uploaded, what percentage of them *convert* to becoming approved customer? That particular conversion rate can be broken down into 2 parts: 1)leads who receive a sales visit, and 2)leads with a sales visit who become approved (since every lead needs at least 1 sales visit to be approved anyway). If we abbreviate “lead uploaded” as “Lead”, Sales Visit as “SV” and “approved customer” as “APP”. We can come up with these fake numbers. 5% of leads become approved, because 10% of them get Sales visits, and half of our sales visits become approved customers.

### Lead -> APP 5%

### Lead -> SV 10%

### SV -> APP 50%

### Doing this sort of analysis, we can focus on different parts of the sales cycle - to optimize them individually. Other parts of the process should have 100% conversion rate. For example, 100% of approved projects should be installed, and interconnected to CFE. Measuring conversion rate isn’t as useful for these parts of the process, so we usually measure [SLAs](https://en.wikipedia.org/wiki/Service-level_agreement) - that is, “what percentage of Approved projects were Interconnected within a 90 day window”. Or we will sometimes measure percentiles for example (95% of our approved customers were interconnected in X days), or our median interconnection time “50% of our customers are interconnected within X days”. In the SQLite database, the following events correspond to the given business events.

| **SQL event name** | **Business event description** |
| --- | --- |
| lead.created | Lead has been uploaded |
| doc.salesVisitReport.uploaded | Lead has a successful sales visit |
| doc.signerCredit.approved | Lead has passed Bright’s credit check |
| doc.subscriptionContract.uploaded | Lead has signed the Bright contract |
| doc.subscriptionContract.approved | Lead has been approved |
| doc.installationComplete.uploaded | Lead has been installed |
| doc.interconnection.approved | Lead has been successfully interconnected |

*Extra Info*

The “attribute” table contains values of past experiments we have run, as well as the “EC” (sales rep) and “CAA” (Bright employee who makes first contact with the customer). ecUserId, and caaUserId. Using this data, we can analyze our best sales rep, best CAA, or evaluate the effectiveness of an experiment. The event table also has several other events we collect during the sales / installation / interconnection process that you are welcome to explore as well.

**Instructions:**

**Visualize/analyze one of our conversion rates (listed above) in whatever way you see fit.**

**OR**

**Visualize/analyze our installation or interconnection speed in whatever way you see fit.**

**OR**

**Find something interesting to analyze that showcases your unique skills!**

NOTE: We will evaluate 2 aspects:

1. The visible part of the report. Be sure the presentation is functional for the final user. What is the best way to present the result? a table? a chart? in which order does the data make more sense? what columns should we include or not?
2. Dexterity and understanding of the data. The joins and filters are made to avoid any discrepancy or possible crack that can damage the veracity of the data?

### **Problem 2**

### We want to add a “Panel Placing Tool” (PPT) to the existing [site selection tool](https://ops.thinkbright.mx/html/pre_design/index.html?leadId=2645298125378044785). This PPT’s function would be to maximize the number of vertically placed panels that you can install on any roof shape. The panels are always the same size, shape, and vertically oriented.

### **Instructions: Explain step by step how you would fit and calculate the maximum amount of panels that can be placed on the selected area. What assumptions you would make and why. What other information would you need from the final user.**

### NOTE: we’re looking for a path to solve the problem, not the actual solution. There’s not one solution but many possibilities, and we’d be looking more at your approach on solving the challenge than the actual solution, so please be descriptive on how you got there

### Example: The input you have is a figure selected like the ones below



## **Section 2. PostgreSQL knowledge**

* **Instructions:** Based on the tables specified in [this document](https://docs.google.com/spreadsheets/d/12sBpAb1N4bHA-MVGWr7ncdzySkqXyRDDiUTT7ul0QAY/edit#gid=0). Write the query to get the answer to the questions.  
  The exercises are based on PostgreSQL and include certain specific data types and functions that you may not be familiarized with. Feel free to let the question empty, it won’t disqualify you. This section is to understand your background and define the future training needed in case you are accepted. The main evaluation criteria are the 2 exercises above.
  + **Problem 1**: Get how many classes are per department
  + **Problem 2**: Get the number of months each student has been on college until today.
  + **Problem 3**: Get the list of students (id and name). That is enrolled on “Matematicas II”
  + **Problem 4**: Get the list of all classes that have surpassed the maximum capacity of the class
  + **Problem 5**: Get the list of professors that haven’t submitted one or more grades.
  + **Problem 6**:Explain the order in which prostgresql will run each section of the query and show what will be the final result.

