Here are simplified versions of three tables from our database:

**LEADS**

**lead\_id** (integer, e.g. 234574)

**lead\_name** (varchar, e.g. “Russ Turk)

**lead\_phone** (varchar, e.g. “206-555-5555”)

**lead\_source** (varchar, e.g. “Paid Google” or “Facebook”)

**lead\_creation\_datetime** (timestamp, e.g. “4/3/2021 05:03:10”)

**valid\_lead** (integer, 0 = not valid, 1 = valid)

**CLIENTS**

**client\_id** (integer, e.g. 54571)

**client\_creation\_datetime** (timestamp, e.g. “4/10/2021 11:04:15”)

**lead\_id** (integer, references the record from the Leads table that is associated with this client)

**ORDERS**

**order\_id** (integer, e.g. 3452)

**order\_amount** (numeric, order amount in cents, e.g. 102454 if order was $1,024.54)

**order\_creation\_datetime** (timestamp, date of order, e.g. “4/15/2021 17:15:45”)

**client\_id** (integer, references the record from the Clients table that is associated with this order)

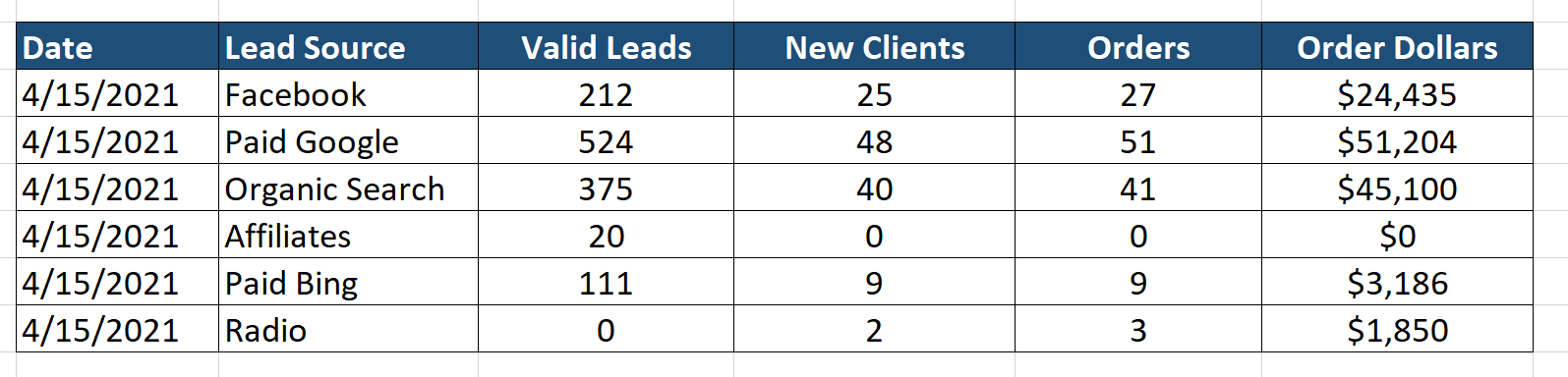
This is the basic pipeline:

* Visitors come to the website and some of those visitors end up filling out a lead form that provides their name, phone number, and their specific tutoring interest.  The submission of a lead form creates a record in the **Leads** table.
* Leads are marked as valid (valid\_lead = 1) or invalid (valid\_lead = 0) based on certain criteria.
* All valid leads flow into the sales pipeline and are called by sales representatives.  If the lead ends up making a purchase then a record in the **Clients** table is created and a record in the **Orders** table is created.  The purchase might happen the same day the lead is created or a subsequent day.  If the same client makes a second purchase, then a second record in the **Orders** table is created.
* Complicating the above, there are also cases where a visitor comes to the website and instead of filling out a lead form they register for a free course.  This creates both a record in the **Leads** table and a record in the **Clients** table but \*not\* a record in the **Orders** table because they have not purchased anything.

In terms of table relationships:

* Each record in the **clients** table is associated with one and only one record in the **leads** table.  A **lead** record is associated with 0 or 1 **client** records.
* Each record in the **orders** table is associated with one and only one record in the **clients** table.  A **client** record is associated with 0 or 1 or many **order** records.

We want a report that shows daily data for the last 30 days broken out by Lead Source.  The output should look something like this:



where...

* **Valid Leads** is the number of valid leads for that lead source on that day.  In the report, valid leads should be counted on the date that the lead record was created.
* **New Clients** is the number of new clients on that day who originally came from that lead source.  A new client is defined as someone making their first purchase and does **not** include free course registrants ***unless*** they subsequently make a purchase.  In the report, new clients should be counted on the date that the first purchase occurred.
* **Orders** is defined as the number of orders on that day from clients who originally came from that lead source.  It could be the clients first order or their tenth order.  In the report, orders should be counted on the date of the order.
* **Order Dollars** is defined as the dollar value of all orders on that day from clients who originally came from that lead source.  In the report, order dollars should be counted on the date of the order.

Note:  As the example of “Radio” in the table illustrates, it’s possible for a lead source to not have any leads on that particular day but to still have new clients and/or orders on that day (from leads that were created on prior days, e.g. the lead was created on 4/1 and the first purchase occurred on 4/7).

Please write SQL to produce the desired report.  Any SQL dialect is ok, but Postgresql is slightly preferred since it’s the primary dialect we use here and the one that I’m personally most familiar with. It’s ok (but not required) to create temporary views/tables (i.e. I’m going to use this SQL to create this particular view of the data and then I’m going to use this other SQL to query the view and generate the report).

Feel free to email me with any questions!