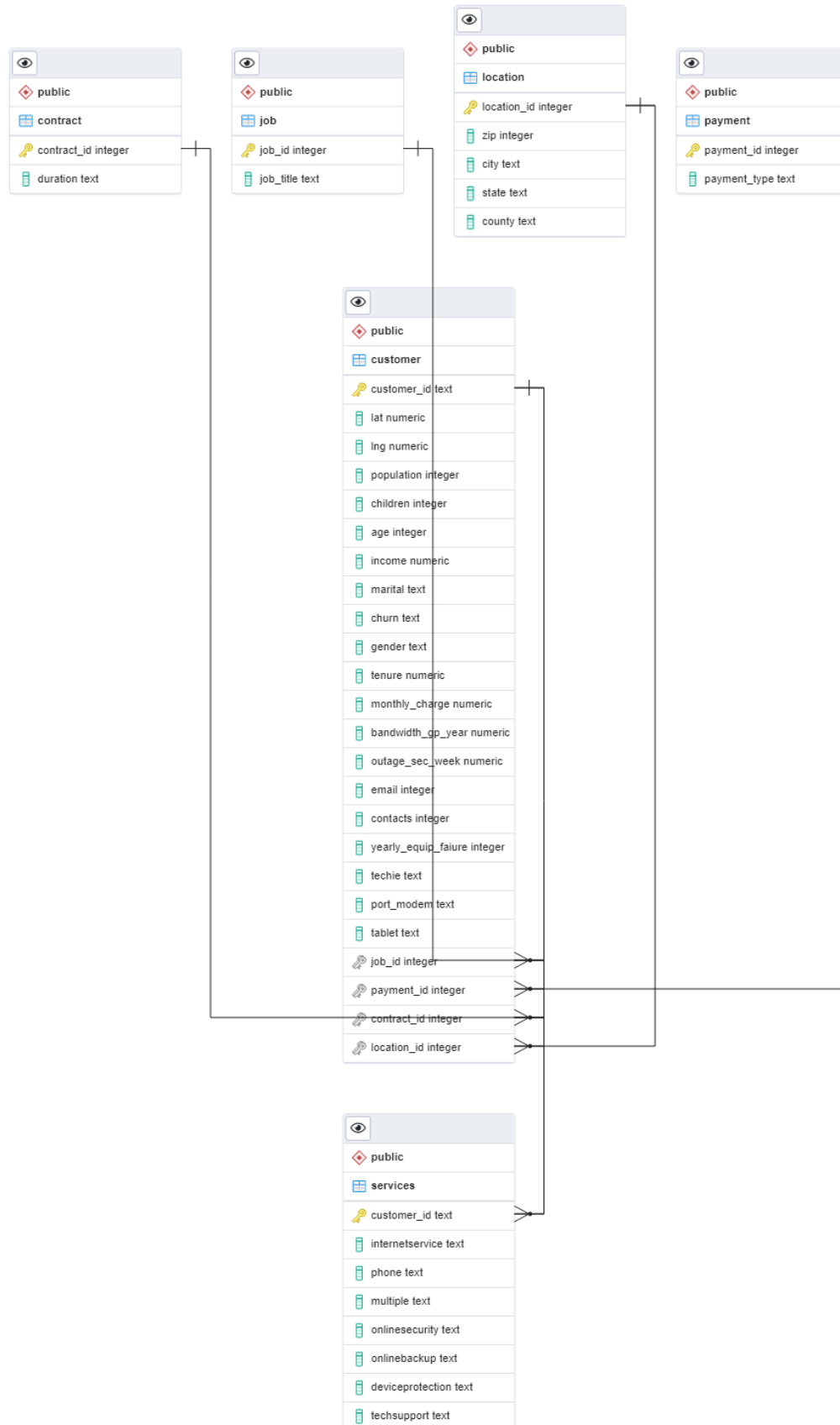


A. I have chosen my research question for this project: "What are the top three services chosen by customers with children?" Identifying trends among customer demographics is vital to many businesses. Marketing campaigns and promotional offers towards new or existing customers of an identified demographic can have increased success and generate additional revenue.

1. The research question will be answered by featuring two critical parts in the SQL statement. The statement will determine the top three services utilized by customers. Most importantly, the statement will have its scope limited to customers with children.
2. The customer_id of both tables is required to connect the two tables. The children column from the customer table and most of the columns from services are required to answer the research question. The breakdown of the required data is as follows:

Source	Table	Column	Data Type
Original Dataset	customer	customer_id	text
Original Dataset	customer	children	int
Add-on CSV	services	customer_id	text
Add-on CSV	services	internetservice	text
Add-on CSV	services	phone	text
Add-on CSV	services	onlinesecurity	text
Add-on CSV	services	onlinebackup	text
Add-on CSV	services	deviceprotection	text
Add-on CSV	services	techsupport	text

B.



The relationship between the customer and services tables is one-to-one. The relationship is forced because the primary key in both tables is customer_id, which automatically applies a unique constraint.

1. The relationship was previously stated as one-to-one, but the ERD shows one-to-many. This is a limitation of pgAdmin4, which only supports one-to-many and many-to-many for table relationships (pgAdmin, n.d.). Despite this limitation, the relationship effectively remains one-to-one due to the primary key constraints in the database.

```
create table services (  
  customer_id text primary key references customer(customer_id),  
  internetservice text,  
  phone text,  
  multiple text,  
  onlinesecurity text,  
  onlinebackup text,  
  deviceprotection text,  
  techsupport text  
);
```

2. This code was used to create the services table in preparation for the CSV data.

The customer_id is set to the primary key for this table and the foreign key references the customer_id in the customer table. All columns were chosen as the text data type for two reasons. The existing tables in the database use the text data type. Furthermore, the text data type has no performance difference compared to the character-varying type (PostgreSQL Global Development Group, n.d.).

copy services from 'C:/LabFiles/Services.csv' delimiter ',' csv header;

3. This line of code successfully imports the data from Services.csv to the services table. Initially, I utilized the import tool in pgAdmin4 to load the data. However, I found that directly using the copy command in PostgreSQL was more straightforward.

```
select
  service,
  count(*) as service_count
from (
  select
    customer.customer_id,
    unnest(array[
      'internetservice',
      'phone',
      'onlinesecurity',
      'onlinebackup',
      'deviceprotection',
      'techsupport'
    ]) as service,
    unnest(array[
      internetservice,
      phone,
      onlinesecurity,
      onlinebackup,
      deviceprotection,
      techsupport
    ]) as service_status
  from
    customer
  join
    services on customer.customer_id = services.customer_id
  where
    customer.children > 0
) as service_selection
where
  service_status = 'Yes'
group by
  service
order by
  service_count desc
limit 3;
```

C. This query identifies the top three services chosen by customers with children. This is accomplished by using unnested arrays to create rows for each service type and their respective status. The query is limited to the customer's active services and customers with children. The result is then counted and grouped by service type, and the top three results are displayed.

1. The output of the query (queryresults.csv) was submitted alongside this report.

D. The services file should be updated daily to remain relevant to normal business activities. However, it should arguably be integrated into the database directly for a variety of reasons beyond the scope of this report. A monthly refresh interval of this data would satisfy our query and research question.

1. Given that the services file contains customers' active subscriptions, the previously recommended daily or real-time updates are vital to normal business activities. For example, service cancellations, refunds, bundle discounts, etc., should be available immediately for reporting purposes, and departments that rely on this information, such as Sales, Finance, and Customer Service. For our research question, a monthly refresh of this data would suffice. Identifying new or existing customers who are more likely to purchase a service is unlikely to shift drastically in less than a month. This, in turn, helps reduce the strain on the database, especially when the question is expanded to many demographics. Additionally, this query could then be connected to other processes to automatically send offers without spamming the customers.

References

pgAdmin. (n.d.). ERD tool. Retrieved from

https://www.pgadmin.org/docs/pgadmin4/latest/erd_tool.html

PostgreSQL Global Development Group. (n.d.). Character types. In *PostgreSQL 13*

Documentation. Retrieved from <https://www.postgresql.org/docs/13/datatype-character.html>