

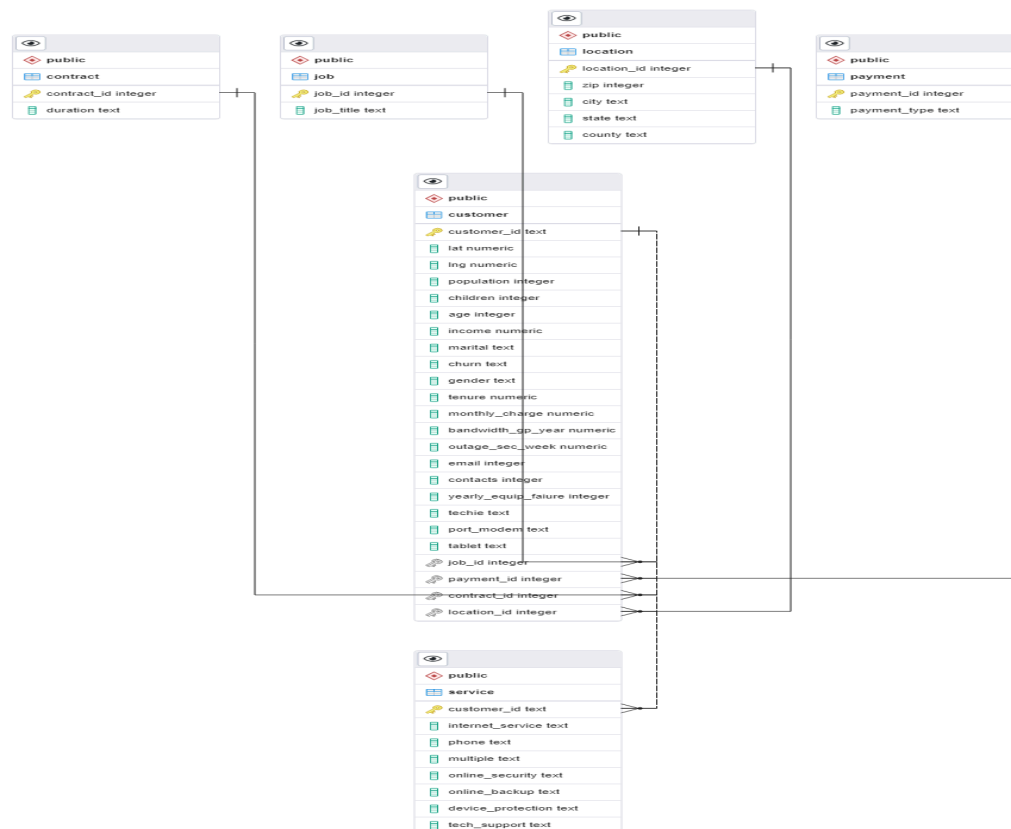
## A1. Research Question

The question I chose for this research assignment is, “What service provider charges its customers more on average?” This question is highly relevant to the business. Suppose we identify the provider that charges its customers more on average. In that case, other service providers can target those customers with promotional offers to help lower their service bill, thus increasing business and revenue for the provider that accomplishes switching over those potential clients.

## A2. Identifying Data

I will use the tables customer and the add-on CSV file called Services.csv. From the customer, the columns used are customer\_id and monthly charges. The columns used from the add-on CSV (Services.csv) are customer\_id and internet\_service. Customer\_id will be used to join the two tables, group the service providers, get the average payment for each service provider, and acquire the information needed to answer the research question.

### B-B1. Entity Relationship Diagram(ERD)



The add-on CSV file I will use is located in the lab-files folder and is called Services.csv. I will load this file onto the table created called service. The file contains eight columns, but only two columns will be used: customer\_id and internet\_service. They will all be defined the same, with “text” as the data type and having “NOT NULL”

values. The rest of the columns are out of the scope of our research question but will be loaded onto the table under the same circumstances.

Customer\_id will be the primary key to join the services with any other table. Customer\_id will also be a foreign key for the customer table to maintain referential integrity between the customer(parent table) and service(child table) tables. In this relationship, customer\_id establishes a one-to-many connection from the customer table (one side) to the service table (many side). Each record in the customer table can be linked to multiple records in the service table, indicating that a single customer may use various services such as internet, phone, online, and security. Conversely, each entry in the service table is associated with one and only one customer.

If referential integrity is not enforced, it could lead to discrepancies, such as service records without corresponding customer entries, which could complicate data analysis and querying. Additionally, referential integrity constraints handle updates and deletions in a way that maintains consistency across related tables, such as cascading changes or restricting modifications to ensure data accuracy. The ERD above illustrates many relationships within the churn database, but the one I focused on was the interaction between the customer and service tables. There were no issues in the ERD.

## **B2. Table creation for CSV file**

```
CREATE TABLE service(  
    customer_id text NOT NULL PRIMARY KEY,  
    internet_service text NOT NULL,  
    phone text NOT NULL,  
    multiple text NOT NULL,  
    online_security text NOT NULL,  
    online_backup text NOT NULL,  
    device_protection text NOT NULL,  
    tech_support text NOT NULL,  
    FOREIGN KEY (customer_id) REFERENCES  
        costumer(customer_id)  
);
```

The SQL statement above was used to create the table for the add-on CSV file. I just typed out the script instead of using the create table tool. The create table tool was a little more complex and had more room for error, and if there were an error in defining data types, then the add-on CSV would not load correctly.

## **B3. Loading CSV file to table created**

```
copy service
(
    customer_id,
    internet_service,
    phone, multiple,
    online_security,
    online_backup,
    device_protection,
    tech_support
)
FROM 'C:/LabFiles/Services.csv'
DELIMITER ','
CSV HEADER;
```

The script above loads the add-on CSV file to the table created. Another way of loading the CSV file to the table is by right-clicking the table and using the importing tool, which I used first. The top script is generated by the importing tool.

### C. Query for research question

```
SELECT s.internet_service,
       ROUND(AVG(c.monthly_charge),2) AS avg_payment,
       ROUND(MIN(c.monthly_charge),2) AS min_payment,
       ROUND(MAX(c.monthly_charge),2) AS max_payment
FROM customer AS c
INNER JOIN service AS s
ON c.customer_id = s.customer_id
WHERE s.internet_service IN ('Fiber Optic','DSL')
GROUP BY s.internet_service
ORDER BY avg_payment DESC;
```

The output will be a list of service providers ("Fiber Optic" and "DSL") along with three statistics for each, avg payment: The average monthly charge, rounded to two decimal places; min\_payment: The minimum monthly charge, max\_payment: The maximum monthly charge. The results will be sorted by the average payment, from highest to lowest. This script helps understand the range and distribution of monthly charges for different service providers, providing insights into the cost variability and averages for "Fiber Optic" and "DSL" customers. Fiber Optic costs more, with an average payment of 186.49 per month, a minimum payment of 112.46, and a maximum

payment of 290.16. DSL costs less, on average, 167.06 per month, with a minimum payment of 92.46 and a maximum payment of 270.16.

#### D-D1. Refresh/Update add-on CSV file

```
--CLEAR ROWS
DELETE FROM service;
--INSERT DATA INTO ROWS
copy service
(
    customer_id,
    internet_service,
    phone, multiple,
    online_security,
    online_backup,
    device_protection,
    tech_support)
FROM 'C:/LabFiles/Services.csv'
DELIMITER ','
CSV HEADER;
```

The above script is how to delete the rows from a table and refresh the rows with current data from Services.csv. The primary purpose of the add-on CSV file is to create promotional offers for customers who might be paying more for their service provider to transition to a different provider with a lower service bill. Many are under contract, and contracts end at the end of the month. Thus, refreshing the table once a month with the add-on CSV file will give optimal data to create insights for promotional offers and which customers to target.

#### F. WEB SOURCES

W3Schools. (n.d.). *SQL foreign key*. W3Schools. Retrieved August 11, 2024, from [https://www.w3schools.com/sql/sql\\_foreignkey.asp](https://www.w3schools.com/sql/sql_foreignkey.asp)

CodeAcademy. (2021, February 3). *How to use SQL joins* [Video]. YouTube. <https://www.youtube.com/watch?v=Dd2ej-QKrWY>