Modify Postgres Tables to make Tableau Visualization Easier

Preliminary Planning:

- Rename csvs folder to reflect that it is the original csvs.

- Create a new folder for updated csvs

- create .sql files that create generated virtual columns:

    Alter ‘Ingredients’ table to include generated virtual column ‘price_per_serv’

    Alter ‘Items’ table to include generated virtual column ‘cost’

    Alter ‘Items’ table to include generated virtual column ‘unit_profit’

    Alter ‘Customer’ table to include generated virtual column ‘times_dined’

Virtual Generated columns are used instead of creating new columns because with a virtual generated column, the data is able to be updated automatically. If I had simply created a column using the following syntax:

ALTER TABLE inventory
ADD COLUMN price_per_serve FLOAT

Then if an ingredient’s carton cost changed, I would have to manually alter the table to account for the price change.

Troubleshooting:

- Alter ‘Customer’ table to include generated virtual column ‘num_transactions’

    What didn’t work:

```
-- Create column 'num_transactions' in Table customers

ALTER TABLE customer
ADD COLUMN num_transactions INTEGER
    GENERATED ALWAYS AS

    (
    SELECT COUNT(customer_id)
    FROM transaction
    WHERE transaction.customer_id = customer.customer_id
    )
    STORED;
```

    Why:

    Postgres Documentation 5.3 specifies that virtual columns cannot reference subqueries :(

- The generation expression can only use immutable functions and cannot use subqueries or reference anything other than the current row in any way.

What works:

A view. Virtual columns aren’t flexible enough to reference views either.

A view of the customer table including the number of transactions will be exported to csv for Tableau