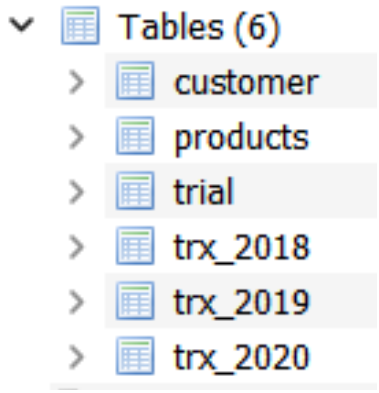
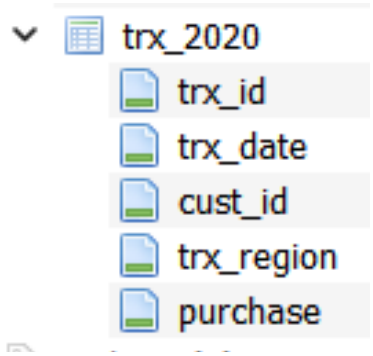
DATA CLEANING & PREPARATION PROCESS IN SQL

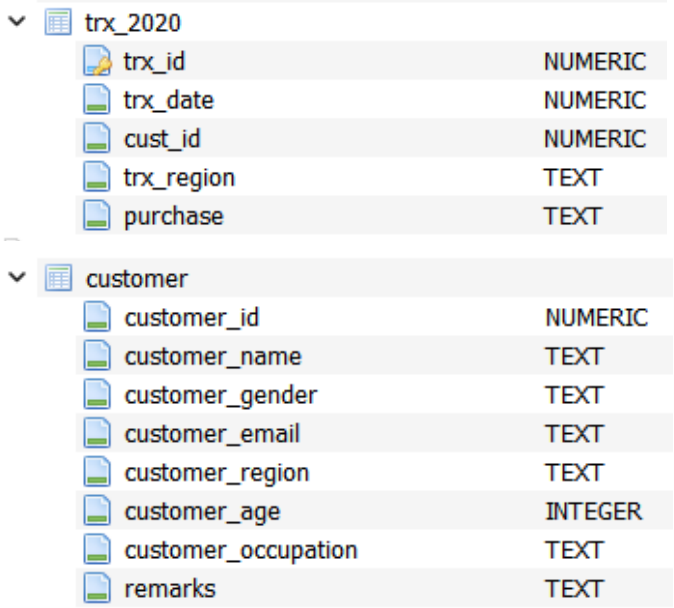
1. **Import tables into database**

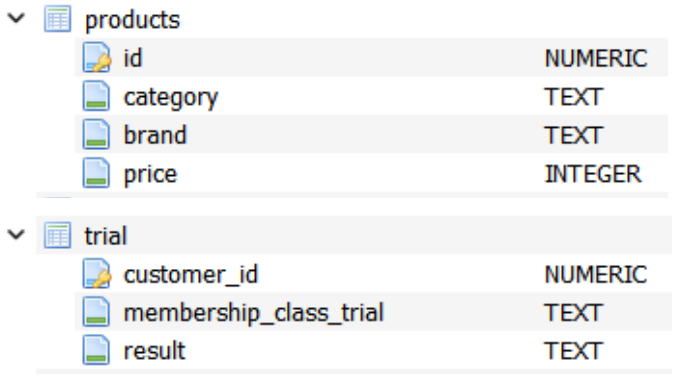


2. **Rename columns for transactions tables to have same column names**



3. **Modify data types for all tables**





4. **Update transaction date (trx_date) to year**

```
UPDATE trx_2018
SET trx_date = '2018';

UPDATE trx_2019
SET trx_date = '2019';
```

5. **Combine all transactions in 1 table (trx)**

```
CREATE TABLE trx AS
SELECT * FROM trx_2018 WHERE 0;

INSERT INTO trx
SELECT * FROM trx_2018
UNION ALL
SELECT * FROM trx_2019
UNION ALL
SELECT * FROM trx_2020;
```

6. **Delete duplicate transactions if available**

```
DELETE FROM trx
WHERE rowid NOT IN (
   SELECT MIN(rowid)
   FROM trx
   GROUP BY trx_id, trx_date, cust_id, trx_region, purchase
);
```

7. **Add gender from customer table**

```
ALTER TABLE trx ADD COLUMN cust_gender TEXT;
```

```
UPDATE trx
SET cust_gender = (
   SELECT customer_gender
   FROM customer
   WHERE customer.customer_id = trx.cust_id
);
```

8. **Format the purchase details to readable json format and separate purchase details to new columns (item_id) and (quantity)**

```
-- Step 1
UPDATE trx
SET purchase = REPLACE(purchase, """", "");

-- Step 2
ALTER TABLE trx
ADD COLUMN item_id INTEGER;

ALTER TABLE trx
ADD COLUMN quantity INTEGER;

-- Step 3
UPDATE trx
SET
   item_id = (
      SELECT GROUP_CONCAT(json_extract(value, '$.id'))
      FROM json_each(trx.purchase)
   ),
   quantity = (
      SELECT GROUP_CONCAT(json_extract(value, '$.qty'))
      FROM json_each(trx.purchase)
   );

-- Step 4
CREATE TABLE trx_final AS
SELECT
   trx_id,
   trx_date,
   cust_id,
   cust_gender,
   trx_region,
   json_extract(value, '$.id') AS item_id,
   json_extract(value, '$.qty') AS quantity
FROM
```

```
    trx,
    json_each(purchase);

DROP TABLE trx;
```

9. **Extract details of category and brand based on (item_id) purchased**

```
-- Step 1
ALTER TABLE trx_final
ADD COLUMN category TEXT;

ALTER TABLE trx_final
ADD COLUMN brand TEXT;

-- Step 2
UPDATE trx_final
SET
    category = products.category,
    brand = products.brand
FROM products
WHERE trx_final.item_id = products.id;
```

10. **Calculate total spent (spent) based on item and quantity purchased**

```
-- Step 1
ALTER TABLE trx_final
ADD COLUMN spent INTEGER;

-- Step 2
UPDATE trx_final
SET spent = products.price * trx_final.quantity
FROM products
WHERE trx_final.item_id = products.id;
```

11. **Create new table containing unique transactions (trx_unique) for data visualization**

```
CREATE TABLE trx_unique AS
SELECT
    trx_id,
    trx_date,
    cust_id,
        cust_gender,
        trx_region,
```

```
    SUM(spent) AS total_spent
FROM
    trx_final
GROUP BY
    trx_id, trx_date, cust_id;
```

12. Create new table containing transactions per customer (trx_cust) for modeling

```
CREATE TABLE trx_cust AS
SELECT
    cust_id,
    cust_gender,
    SUM(total_spent) AS total_spent,
    COUNT(*) AS count_trx
FROM trx_unique
GROUP BY cust_id;
```

13. Calculate average spent per transaction (avg_spent_per_trx) and average transactions in a year (avg_annual_trx)

```
-- Step 1
ALTER TABLE trx_cust
ADD COLUMN avg_spent_per_trx INTEGER;

ALTER TABLE trx_cust
ADD COLUMN avg_annual_trx INTEGER;

-- Step 2
UPDATE trx_cust
SET avg_spent_per_trx = ROUND(total_spent * 1.0 / count_trx, 2);

-- Step 3
UPDATE trx_cust
SET avg_annual_trx = ROUND(count_trx * 1.0 / 3, 0);
```

14. Add customer and transaction details (from trx_cust) to trial table

```
-- Step 1
ALTER TABLE trial
ADD COLUMN total_spent INTEGER;

ALTER TABLE trial
ADD COLUMN count_trx INTEGER;
```

```
ALTER TABLE trial
ADD COLUMN avg_spent_per_trx INTEGER;

ALTER TABLE trial
ADD COLUMN avg_annual_trx INTEGER;

-- Step 2
UPDATE trial
SET total_spent = trx_cust.total_spent,
    count_trx = trx_cust.count_trx,
    avg_spent_per_trx = trx_cust.avg_spent_per_trx,
    avg_annual_trx = trx_cust.avg_annual_trx
FROM trx_cust
WHERE trial.customer_id = trx_cust.cust_id;
```