Data Cleaning & Sub-table Creations

skuinfo: packsize

To avoid type error when copying the skuinfo.csv to database, the field `packsize` was set as text type. By running the SQL query below, there is some rows with `packsize` that cannot be converted by integers automatically.

```
SELECT *
FROM skuinfo
WHERE packsize !~ E'^\\d+$';
```

So we are going to replace those values by the mode of `packsize` : 1.

```
SELECT mode() WITHIN GROUP (ORDER BY packsize::integer)
FROM skuinfo
WHERE packsize ~ E'^\\d+$';

Output:
1

UPDATE skuinfo
SET packsize = '1'
WHERE sku IN (
    SELECT sku
    FROM skuinfo
    WHERE packsize !~ E'^\\d+$'
);
```

Then we can convert text to integer.

```
ALTER TABLE skuinfo
ALTER COLUMN packsize TYPE integer USING packsize::integer;
```

trnsact: salesdate

Change the data type of `salesdate` from text to date.

```
ALTER TABLE trnsact
ALTER COLUMN saledate
TYPE DATE USING saledate::date;
```

Build index on `salesdate` for faster CRUD operations.

```
CREATE INDEX saledate_idx ON trnsact(saledate);
```

trnsact: quantity

Build index on `quantity` for faster CRUD operations.

```
CREATE INDEX quantity_idx ON trnsact(quantity);
```

Change the data type of quantity from text to integer.

```
ALTER TABLE trnsact
ALTER COLUMN quantity TYPE integer USING quantity::integer;
```

Subset table creation

We would like to investigate the sells on the Black Friday back on 26 Nov, 2004. Therefore, we created a separate table which contain this subset of transactions and later joined with `skuinfo` and `skstinfo` to get retrieve more information about the products.

```
# Create subset transaction table on date 2004-11-26 (The Black Friday in 2004)
CREATE TABLE black_friday_trnsact AS (
  SELECT *
  FROM trnsact
  WHERE saledate = '2004-11-26');


# Create indexes to make join operations more efficient
CREATE INDEX sku_idx ON black_friday_trnsact (sku);
CREATE INDEX store_idx ON black_friday_trnsact (store);
CREATE INDEX skst_sku_idx ON skstinfo (sku);
CREATE INDEX skst_store_idx ON skstinfo (store);


# Get retail cost price columns
# and all sku information about the products joined with transaction history
CREATE TABLE joined_trnsact AS (
  WITH retail_trnsact AS (
    SELECT s.sku, s.cost, s.retail,
      b.stype, b.quantity, b.orgprice, b.amt
    FROM skstinfo s INNER JOIN black_friday_trnsact b ON
      (s.sku = b.sku AND s.store = b.store)
  )
  SELECT r.*, s.style, s.color, s.size, s.packsize, s.vendor, s.brand
  FROM skuinfo s INNER JOIN retail_trnsact r ON s.sku = r.sku
);
```