Identify the purchase history of customer Bo Li in the last year. Display the customer name, products purchased and price.

$A \leftarrow \sigma_{customerFirst=\text{"}Bo\text{"}\ \wedge\ customerLast=\text{"}Li\text{"}}(Customer)$

$B \leftarrow \sigma_{A.customerId\, =\, Transaction.customerId\ \wedge\ Transaction.transDate\, \geq\, 3-9-21}(A \times Transaction)$

$C \leftarrow \sigma_{B.transId\, =\, transProduct.transId}\ (B \times transProduct)$

- This is because every transaction can have multiple products. So we find all the products in the transProduct relation corresponding to the same transId.

$D \leftarrow \sigma_{C.productId\, =\, product.productId}\ (C \times product)$

$(Answer) \leftarrow \pi_{customerFirst,\ customerLast,\ productName,\ productPrice}(D)$

Identify products available at the Flushing store. Display the product name, calories and price.

$A \leftarrow \sigma_{storeCity=\text{"}flushing\text{"}}(Store)$

$B \leftarrow \sigma_{A.storeId\, =\, storeStock.storeId}(A \times storeStock)$

- This step of joining the storeStock relation allows us to view the stock of products in this specific store by using the storeId.

$C \leftarrow \sigma_{B.productId\, =\, product.productId}(B \times product)$

$(Answer) \leftarrow \pi_{productName,\ productCalories,\ productPrice}(C)$

Identify the number of active customers. Display the number.

$(Answer) \leftarrow\ \Im_{\ count\ customerId}\ (Customer)$

Identify the number of customers by zipcode. Display 3 columns: zip code, number of customers and total dollar amount of purchases.

$(A) \leftarrow \sigma_{customer.customerId\, =\, Transaction.customerId}(Transaction \times Customer)$

$(B) \leftarrow \sigma_{transProduct.transId\, =\, A.transId}(transProduct \times A)$

$(C) \leftarrow \sigma_{product.productId\, =\, B.productId}(product \times B)$

$(Answer) \leftarrow \rho_{answer}(Zip\ code,\ Number\ of\ customers,\ total\ revenue)\ {}_{customerZip}\Im_{count\ customerId,\ sum\ productPrice}(C)$

Identify all customers who purchased milk today at the Flushing store. Display the customer name, product name and price.

$(A) \leftarrow \sigma_{customer.customerId\, =\, Transaction.customerId\ \wedge\ Transaction.transDate\, =\, \text{"}3-9-22\text{"}}(Transaction \times Customer)$

$(B) \leftarrow \sigma_{A.transId\, =\, transProduct.transId}\ (A \times transProduct)$

- This just gets all the products in a transaction. This isn't important for this answer because the projection will remove all duplicate entries but this doesn't change the final answer anyway.

$(C) \leftarrow \sigma_{product.productId\, =\, B.productId\ \wedge\ product.productName\, =\, \text{"}milk\text{"}}(product \times B)$

$(Answer) \leftarrow \pi_{customerFirst,\ customerLast,\ productName,\ productPrice}(B)$

Identify customers who have not made a purchase in the last year. Display the customer name and email address.

$(purchased) \leftarrow \sigma_{Transaction.transDate \geq 3-9-21} (Transaction)$

- This will filter all the transactions in all of Stop and shops databases who have purchased something in the last year.

$(notPurchased) \leftarrow \pi_{customerId}(customer) \; - \; \pi_{customerId} (purchased)$

- This will find the complement by subtracting the customerIds.

$(display) \leftarrow \sigma_{notPurchased.customerId = Customer.customerId} (notPurchased \times Customer)$

- This will add customer columns to appropriate Ids.

$(Answer) \leftarrow \pi_{customerFirst, \, customerLast, \, customerEmail} (display)$

Identify staff not assigned to stores. Display the staff first and last name.

$(Assigned) \leftarrow \sigma_{Staff.staffId = storeStaff.staffId}(Staff \times storeStaff)$

- This will find all the staff assigned to stores by joining the [storeStaff] relation which has a set of composite primary keys {storeId and staffId}. This is needed because 1 staff can work at many stores.

$(notAssigned) \leftarrow \pi_{staffId}(Staff) \; - \; \pi_{staffId} (Assigned)$

$(B) \leftarrow \sigma_{Staff.staffId = notAssigned.staffId}(Staff \times notAssigned)$

$(Answer) \leftarrow \pi_{staffFirst, \, staffLast}(B)$