DATA TYPE CONVERSION AND THE USE OF UNION ALL OPERATOR TO APPEND VERTICALLY TWO TABLES

```
use Commercial_Project

-- CHANGE DATA TYPES FROM VARCHAR(50) TO DATE FOR THE FOLLOWING COLUMNS

ALTER TABLE return_data
ALTER COLUMN Return_date date

ALTER TABLE customers
ALTER COLUMN birthdate date

ALTER TABLE customers
ALTER COLUMN acct_open_date date

ALTER TABLE stores
ALTER COLUMN first_opened_date date

ALTER TABLE stores
ALTER COLUMN last_remodel_date date

-- TO VIEW ALL THE COLUMNS DATA TYPES OF A TABLE

SELECT *
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'customers'


-- CHANGE DATA TYPE FROM VARCHAR TO INTEGER IN THE CUSTOMERS TABLE

select * from Customers

SELECT *
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'customers'

ALTER TABLE customers
ALTER COLUMN total_children integer

ALTER TABLE customers
ALTER COLUMN num_children_at_home integer

-- CHANGE DATA TYPE FROM FLOAT TO VARCHAR(50) IN THE PRODUCTS TABLE

select * from Products

SELECT *
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'products'

ALTER TABLE products
ALTER COLUMN product_sku varchar(50)

-- CHANGE DATA TYPE FROM FLOAT TO DECIMAL(10,2) IN THE PRODUCTS TABLE

ALTER TABLE products
ALTER COLUMN product_retail_price decimal(10,2)

ALTER TABLE products
ALTER COLUMN product_weight decimal(10,2)

-- CHANGE DATA TYPE FROM VARCHAR TO DECIMAL(10,2) IN THE PRODUCTS TABLE
```

```sql
ALTER TABLE products
ALTER COLUMN product_cost decimal(10,2)


-- CHANGE DATA TYPES IN THE RETURN_DATA TABLE

select * from Return_Data

SELECT *
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'return_data'

ALTER TABLE return_data
ALTER COLUMN quantity decimal(10,2)


-- CHANGE DATA TYPE FROM VARCHAR TO DATE IN THE TRANSACTIONS_1997 TABLE

select * from Transactions_1997

SELECT *
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'transactions_1997'

ALTER TABLE transactions_1997
ALTER COLUMN transaction_date date

ALTER TABLE transactions_1997
ALTER COLUMN stock_date date

-- CHANGE DATA TYPE FROM VARCHAR TO INTEGER IN THE TRANSACTIONS_1997 TABLE

ALTER TABLE transactions_1997
ALTER COLUMN quantity integer

-- CHANGE DATA TYPE FROM VARCHAR TO DATE IN THE TRANSACTIONS_1998 TABLE

select * from Transactions_1998

SELECT *
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'transactions_1998'

ALTER TABLE transactions_1998
ALTER COLUMN transaction_date date

ALTER TABLE transactions_1998
ALTER COLUMN stock_date date

-- CHANGE DATA TYPE FROM VARCHAR TO INTEGER IN THE TRANSACTIONS_1998 TABLE

ALTER TABLE transactions_1998
ALTER COLUMN quantity integer


-- UNION ALL operator which does not remove duplicates to append vertically Transactions_1997 and
Transactions_1998
-- in order to mimic the function Append Queries or Get Data > From Folder in Power BI

select transaction_date, stock_date, product_id, customer_id, store_id, quantity from Transactions_1997
UNION ALL
select transaction_date, stock_date, product_id, customer_id, store_id, quantity from Transactions_1998
order by transaction_date ASC
```

```
-- HOW TO MOVE UNION QUERY RESULTS TO A NEW TABLE (in this case I named it Transaction_Data)

SELECT * INTO Transaction_Data
FROM
(select transaction_date, stock_date, product_id, customer_id, store_id, quantity from
Transactions_1997
UNION ALL
select transaction_date, stock_date, product_id, customer_id, store_id, quantity from
Transactions_1998) A

select * from Transaction_Data
order by transaction_date ASC
```

ADD FOREIGN KEY TO CONNECT TWO DIMENSION (LOOKUP) TABLES AS A POWER BI SNOWFLAKE SCHEMA

```
-- ADD FOREIGN KEY to Region_id column of the Stores table
-- to reference the Region_id PK column of the Regions table as a Power BI 'snowflake' schema

ALTER TABLE Stores
ADD FOREIGN KEY (region_id)
REFERENCES Regions(region_id)
```

ADD FOREIGN KEYS TO FACT(DATA) TABLES IN ORDER TO REFERENCE THEM WITH DIMENSION (LOOKUP) TABLES AS A POWER BI STAR SCHEMA

```
ALTER TABLE Transaction_data
ADD FOREIGN KEY (store_id)
REFERENCES Stores(store_id)


ALTER TABLE Transaction_data
ADD FOREIGN KEY (customer_id)
REFERENCES Customers(customer_id)

ALTER TABLE Transaction_data
ADD FOREIGN KEY (product_id)
REFERENCES Products(product_id)


ALTER TABLE Return_data
ADD FOREIGN KEY (product_id)
REFERENCES Products(product_id)

ALTER TABLE Return_data
ADD FOREIGN KEY (store_id)
REFERENCES Stores(store_id)
```