SQL Scripts for generating STAR Schema and for extracting data from transactional databases to Fact and dimension tables

DIMENSION TABLES

```
CREATE TABLE dim_product
(
  product_id INT NOT NULL,
  product_version INT NOT NULL,
  external_product_code INT NOT NULL,
  short_name VARCHAR NOT NULL,
  long_name VARCHAR NOT NULL,
  description INT NOT NULL,
  thumbnail_location VARCHAR NOT NULL,
  image_location VARCHAR NOT NULL,
  weight INT NOT NULL,
  quantity_in_inventory INT NOT NULL,
  price INT NOT NULL,
  date_created DATE NOT NULL,
  date_last_updated DATE NOT NULL,
  PRIMARY KEY (product_id)
);
```

```
CREATE TABLE dim_order

  order_id VARCHAR NOT NULL,
  external_order_id VARCHAR NOT NULL,
  order_status VARCHAR NOT NULL,
  paypal_payment_transaction_id INT NOT NULL,
  order_total INT NOT NULL,
  date_created DATE NOT NULL,
  date_last_updated DATE NOT NULL,
  date_submitted DATE NOT NULL,
  date_shipped DATE NOT NULL,
  date_cancelled DATE NOT NULL,
  date_refunded DATE NOT NULL,
  PRIMARY KEY (order_id)

);
```

```
CREATE TABLE dim_online_line_item
(
  line_item_id VARCHAR NOT NULL,
  product_version INT NOT NULL,
  quantity INT NOT NULL,
  price INT NOT NULL,
  date_created DATE NOT NULL,
  date_last_updated DATE NOT NULL,
  PRIMARY KEY (line_item_id)
);
```

```
CREATE TABLE dim_cust_address
(
  address_id INT NOT NULL,
  street_address INT NOT NULL,
  city INT NOT NULL,
  state VARCHAR NOT NULL,
  posatl_code INT NOT NULL,
  date_created DATE NOT NULL,
  date_last_updated DATE NOT NULL,
  PRIMARY KEY (address_id)
);
```

```
CREATE TABLE dim_customer
(
  customer_id INT NOT NULL,
  first_name VARCHAR NOT NULL,
  last_name VARCHAR NOT NULL,
  date_created DATE NOT NULL,
  date_last_updated DATE NOT NULL,
  PRIMARY KEY (customer_id)
);
```

```
CREATE TABLE dim_order_fulfillment
(
  fulfillment_id INT NOT NULL,
  fulfillment_status VARCHAR NOT NULL,
  package_tracking_number VARCHAR NOT NULL,
  date_created DATE NOT NULL,
  date_last_updated DATE NOT NULL,
  date_received DATE NOT NULL,
  date_in_progress DATE NOT NULL,
  date_shipped DATE NOT NULL,
  PRIMARY KEY (fulfillment_id)
);
```

```
CREATE TABLE dim_employee
(
  employee_id INT NOT NULL,
  employee_badge_number INT NOT NULL,
  first_name VARCHAR NOT NULL,
  last_name VARCHAR NOT NULL,
  date_created DATE NOT NULL,
  date_last_updated DATE NOT NULL,
  PRIMARY KEY (employee_id)
);
```

```
CREATE TABLE dim_tracking
(
  generated_tracking_id INT NOT NULL,
  user_zip_code INT NOT NULL,
  date_created DATE NOT NULL,
  date_last_updated DATE NOT NULL,
  PRIMARY KEY (generated_tracking_id)
);
```

```
CREATE TABLE dim_shopping_log
(
  shopping_log_id INT NOT NULL,
  shopping_event INT NOT NULL,
  timestamp INT NOT NULL,
  PRIMARY KEY (shopping_log_id)
);
```

```
CREATE TABLE dim_online_log
(
  online_log_id INT NOT NULL,
  order_event INT NOT NULL,
  timestamp INT NOT NULL,
  PRIMARY KEY (online_log_id)
);
```

FACT TABLE

```
CREATE TABLE fact_table
(
  product_id INT NOT NULL,
  order_id VARCHAR NOT NULL,
  line_item_id VARCHAR NOT NULL,
  customer_id INT NOT NULL,
  address_id INT NOT NULL,
  fulfillment_id INT NOT NULL,
  employee_id INT NOT NULL,
  online_log_id INT NOT NULL,
  shopping_log_id INT NOT NULL,
  generated_tracking_id INT NOT NULL,
  FOREIGN KEY (product_id) REFERENCES dim_product(product_id),
  FOREIGN KEY (order_id) REFERENCES dim_order(order_id),
  FOREIGN KEY (line_item_id) REFERENCES dim_online_line_item(line_item_id),
  FOREIGN KEY (customer_id) REFERENCES dim_customer(customer_id),
  FOREIGN KEY (address_id) REFERENCES dim_cust_address(address_id),
  FOREIGN KEY (fulfillment_id) REFERENCES dim_order_fulfillment(fulfillment_id),
  FOREIGN KEY (employee_id) REFERENCES dim_employee(employee_id),
  FOREIGN KEY (online_log_id) REFERENCES dim_online_log(online_log_id),
  FOREIGN KEY (shopping_log_id) REFERENCES dim_shopping_log(shopping_log_id),
  FOREIGN KEY (generated_tracking_id) REFERENCES dim_tracking(generated_tracking_id)
);
```

ETL queries from transactional database to STAR Schema

Inserting DATA into Dimension tables

- INSERT INTO **dim_product** (product_id, product_version, external_product_code , short_name, long_name,description, thumbnail_location, image_location, weight, quantity_in_inventory, price, date_created, date_last_updated)

```
SELECT product_id,
product_version,
external_product_code ,
short_name,
 long_name,description,
thumbnail_location,
image_location,
weight,
quantity_in_inventory,
 price,
date_created,
date_last_updated
FROM product;
```

- INSERT INTO **dim_order** (order, external_order_id, order_status, paypal_payment_transaction_id, order_total, date_created, date_last_updated, date_submitted, date_shipped, date_cancelled, date_refunded)
  SELECT order, external_order_id, order_status, paypal_payment_transaction_id, order_total, date_created, date_last_updated, date_submitted, date_shipped, date_cancelled, date_refunded FROM ORDER;

- INSERT INTO **dim_online_line_item** (line_item_id, product_version, quantity, price, date_created, date_last_updated)
  SELECT line_item_id, product_version, quantity, price, date_created, date_last_updated FROM online_line_item;

- INSERT INTO **dim_cust_address**(address_id, street_address, city, state, posatl_code, date_created, date_last_updated)
  SELECT address_id, street_address, city, state, posatl_code, date_created, date_last_updated FROM customer_address;

- INSERT INTO **dim_customer** (customer_id, first_name, last_name, date_created, date_last_updated)
  SELECT customer_id, first_name, last_name, date_created, date_last_updated FROM customer;

- INSERT INTO **dim_order_fulfillment** (fulfillment_id, fulfillment_status, package_tracking_number, date_created, date_last_updated, date_received, date_in_progress, date_shipped)
  SELECT (fulfillment_id, fulfillment_status, package_tracking_number, date_created, date_last_updated, date_received, date_in_progress, date_shipped FROM order_fulfillment;

- INSERT INTO **dim_employee** (employee_id, employee_badge_number, first_name, last_name, date_created, date_last_updated)
  SELECT employee_id, employee_badge_number, first_name, last_name, date_created, date_last_updated FROM employee;

- INSERT INTO **dim_tracking** (generated_tracking_id, user_zip_code, date_created, date_last_updated)
  SELECT generated_tracking_id, user_zip_code, date_created, date_last_updated FROM tracking;

Inserting DATA into Fact table

1) Create a temp table and generate schema like fact table

```
SELECT * INTO #fact_ship FROM dw.dbo.fact_table WHERE 1 = 0
```

2) Populate the temp table

INSERT INTO #fact (<coulumn_names>)
SELECT (all primary keys from each dimension table using LEFT JOIN with actual transactional tables and fact table
And also using ISNULL on each column to handle default value)

3) Update fact_table with existing records

Update ft
<update all the columns with temp table fact table columns>
FROM fact_table ft INNER JOIN #fact_ship fs;

4) Insert the new record.

BUSINESS QUESTIONS

What is the customer conversion rate from when a customer starts shopping to submitting an order? The information will be used to create a sales funnel visualization tool. For example:

1. **New traffic = 100%**
2. **Added Items to Cart = 80%**
3. **Submitted Order = 20%**
4. **Received Order = 15%**

SQL QUERY : Creating funnel visualization data on today's date

```
SELECT
        concat(cast(Temp.new_traffic as varchar),' %') as New_Traffic,
        concat(cast((Temp.added_items/Temp.new_traffic )as varchar),' %') as Added_Items-to_Cart,
        concat(cast((Temp.submitted_order/Temp.new_traffic )as varchar),' %') as Submitted_Order,
        concat(cast((Temp.received_order/Temp.new_traffic )as varchar),' %') as Received_Order

FROM
(

        SELECT count(*)   as new_traffic
        FROM dim_tracking
        WHERE date_created = cast(getdate() as date)

        UNION

        SELECT count(*) as added_items
        FROM fact _table as f
        INNER JOIN dim_tracking as t ON (f.tracking_id = t.tracking_id)
        INNER JOIN dim_shopping_log as s ON (f.shopping_log_id = s.shopping_log_id)
        WHERE s.shopping_event = 'added item to cart' AND cast(s.timestamp as date) = cast(getdate() as date)

        UNION

        SELECT count(*) as submitted_order
        FROM fact_table as f
        INNER JOIN dim_tracking as t ON (f.tracking_id = t.tracking_id)
        INNER JOIN dim_shopping_log as s ON (f.shopping_log_id = s.shopping_log_id)
        INNER JOIN dim_online_log as ol ON (ol.online_log_id = f.online_log_id)
        INNER JOIN dim_order as o ON (o.order_id = f.order_id)
        WHERE ol.order_event = 'submitted order' AND cast(ol.timestamp as date) = cast(getdate() as date)

        UNION

        SELECT count(*) as recieved_order
        FROM fact_table as f
        INNER JOIN dim_tracking as t ON (f.tracking_id = t.tracking_id)
        INNER JOIN dim_shopping_log as s ON (f.shopping_log_id = s.shopping_log_id)
        INNER JOIN dim_online_log as ol ON (ol.online_log_id = f.online_log_id)
        INNER JOIN dim_order as o ON (o.order_id = f.order_id)
        INNER JOIN dim_order_fulfillment olf (olf.fulfillment_id = f.fulfillment_id)
        WHERE olf.fulfillment_status = 'RECIEVED' AND cast(olf.date_last_updated as date) = cast(getdate() as date)
) As Temp;
```

What are the ZIP codes of users who browsed for products but did not submit an order? What products did they browse?

SQL QUERY

Here finding all those users details in CTE who just viewed the item and in the final query we put customer id as NULL/0. As mentioned in assignment customer ID will only generate when any usr try to submit the order and they must create an account. But if they only viewed then they have a tracking Id but not the customer ID.

```
with temp (track_id, zip_code, prod_id,cust_id)
AS
(

        SELECT
        f.generated_tracking_id as track_id,
        t.user_zip_code as zip_code
        f.product_id as prod_id,
        f.customer_id as cust_id
        FROM fact_table as f
        INNER JOIN dim_tracking as t (f.generated_tracking_id = t.generated_tracking_id)
        INNER JOIN dim_shopping_log as s (f.shopping_log_id = s.shopping_log_id)
        WHERE s.shopping_event = “viewed item”
)
    SELECT distinct
    t.zip_code as user_code,
    p.external_product_code as product_code,
    p.short_name as prod_name
    FROM temp t
    INNER JOIN dim_product as p (p.product_id = t.prod_id)
    WHERE ISNULL(cust_id,0) = 0
```