Milestone 3

Create the database schema
Develop the star-based schema of the database, ensuring that the columns are of the correct data types.

Task 1: Cast the columns of the orders_table to the correct data types.

```
ALTER TABLE orders_table
  ALTER COLUMN card_number TYPE VARCHAR(19),
  ALTER COLUMN store_code TYPE VARCHAR(12),
  ALTER COLUMN product_code TYPE VARCHAR(11),
  ALTER COLUMN date_uuid TYPE UUID USING CAST(date_uuid AS UUID),
  ALTER COLUMN user_uuid TYPE UUID USING CAST(user_uuid AS UUID),
  ALTER COLUMN product_quantity TYPE SMALLINT;
```

Task 2: Cast the columns of the dim users_table to the correct data types.

```
ALTER TABLE dim_user_table
    ALTER COLUMN country_code TYPE VARCHAR(2),
    ALTER COLUMN first_name TYPE VARCHAR(255),
    ALTER COLUMN last_name TYPE VARCHAR(255),
    ALTER COLUMN date_of_birth TYPE DATE USING date_of_birth::DATE,
    ALTER COLUMN user_uuid TYPE UUID USING user_uuid::UUID,
    ALTER COLUMN join_date TYPE DATE USING join_date::DATE;
```

Task 3: Update the dim_store _details table.

```
UPDATE dim_store_details
SET locality = COALESCE(locality, 'N/A');
```

Task 4: Make changes to the dim_products table for the delivery team.

```
CREATE TABLE dim_products (
    product_id SERIAL PRIMARY KEY,
    product_name VARCHAR(255),
    product_price DECIMAL(10, 2),
    weight DECIMAL(10, 2),
    weight_class VARCHAR(255)
);
```

```
-- Remove £ character from product_price
UPDATE dim_products
SET product_price = REPLACE(CAST(product_price AS VARCHAR), '£', '')::DECIMAL(10, 2);

-- Update weight_class based on weight range
UPDATE dim_products
SET weight_class =
    CASE
        WHEN weight < 2.0 THEN 'Light'
        WHEN weight >= 2 AND weight < 40 THEN 'Mid_Sized'
        WHEN weight >= 40 AND weight < 140 THEN 'Heavy'
        WHEN weight >= 140 THEN 'Truck_Required'
    END;
```

Task 5: Update the dim_products table with the required data types.

```
-- Check if the still_available_temp column exists
DO $$
BEGIN
  IF EXISTS (SELECT 1 FROM information_schema.columns WHERE table_name =
'dim_products' AND column_name = 'still_available_temp') THEN
    -- Rename the still_available column
    ALTER TABLE dim_products
    RENAME COLUMN still_available TO still_available_temp;

    -- Update data types of columns
    ALTER TABLE dim_products
        ALTER COLUMN product_price TYPE FLOAT,
        ALTER COLUMN weight TYPE FLOAT,
        ALTER COLUMN EAN TYPE VARCHAR(255),
        ALTER COLUMN product_code TYPE VARCHAR(255),
        ALTER COLUMN date_added TYPE DATE,
        ALTER COLUMN uuid TYPE UUID,
        ALTER COLUMN still_available_temp TYPE BOOL,
        ALTER COLUMN weight_class TYPE VARCHAR(255);

    -- Rename the still_available_temp column back to still_available
    ALTER TABLE dim_products
    RENAME COLUMN still_available_temp TO still_available;
  END IF;
END $$;
```

Task 6: Update the dim_date_times table.

```
CREATE TABLE dim_date_times (
    month VARCHAR(255),
    year VARCHAR(255),
    day VARCHAR(255),
    time_period VARCHAR(255),
    date_uuid UUID
);
-- Update data types of columns in dim_date_times
ALTER TABLE dim_date_times
    ALTER COLUMN month TYPE VARCHAR(255),
    ALTER COLUMN year TYPE VARCHAR(255),
    ALTER COLUMN day TYPE VARCHAR(255),
    ALTER COLUMN time_period TYPE VARCHAR(255),
    ALTER COLUMN date_uuid TYPE UUID;
```

Task 7: Updating the dim_ card_details table.

```
-- Create dim_card_details table
CREATE TABLE dim_card_details (
    card_number VARCHAR(255),
    expiry_date VARCHAR(255),
    date_payment_confirmed DATE
);

-- Update data types of columns in dim_card_details
ALTER TABLE dim_card_details
    ALTER COLUMN card_number TYPE VARCHAR(255),
    ALTER COLUMN expiry_date TYPE VARCHAR(255),
    ALTER COLUMN date_payment_confirmed TYPE DATE;
```

Task 8: Create the primary keys in the dimension tables.

```
-- Adds primary keys in dim_tables
ALTER TABLE dim_card_details
        ADD CONSTRAINT pk_card_nuber PRIMARY KEY (card_number);

ALTER TABLE dim_date_times
        ADD PRIMARY KEY (date_uuid);

ALTER TABLE dim_products
        ADD PRIMARY KEY (product_code);
```

```
ALTER TABLE dim_store_details
        ADD PRIMARY KEY (store_code);

ALTER TABLE dim_users
        ADD PRIMARY KEY (user_uuid);
```

Task 9: Finalising the star-based schema & adding the foreign keys to the orders table.

```
ALTER TABLE orders_table
        ADD FOREIGN KEY (card_number)
        REFERENCES dim_card_details(card_number);

ALTER TABLE orders_table
        ADD FOREIGN KEY (date_uuid)
        REFERENCES dim_date_times(date_uuid);

ALTER TABLE orders_table
        ADD FOREIGN KEY (product_code)
        REFERENCES dim_products(product_code);

ALTER TABLE orders_table
        ADD FOREIGN KEY (store_code)
        REFERENCES dim_store_details(store_code);

ALTER TABLE orders_table
        ADD FOREIGN KEY (user_uuid)
        REFERENCES dim_users(user_uuid);
```