JUST EAT TAKEAWAY.COM

DWH - Case study

Kiran Khan
Kiran.khan19978@gmail.com

Problem Statement

Design the ETL pipeline and data Warehouse (DWH)

Solution

Solution includes:

- Ingestion Process (Consume datasets from S3 URL and ingest into raw tables)
- Dimensional modelling of data using DBT
- Data quality considerations and automated testing using DBT.
- Workflow development using Talend Open Studio, which can be scheduled.
- Proper exception handling and logging of all the steps.

Ingestion Process

Consists of a well written python code with proper exception handling which downloads and un compresses the products and reviews files. Then loads Json files into raw tables present in postgreSQL locally setup database.

Dimensional Modelling

Dimensional model consists of 5 dimensions and a fact table which are as follow:

1. **Calendar Dimension**
   All the time-related information like month, quarter, year, etc. It will have the following columns:

Date (PK)	Date type column contains dates in yyyy-mm-dd format
Year	The year in yyyy format
Quarter	The quarter number in the year
Month	The month number in the year

2. **Product Dimension**
   Contains product related and some additional attributes to check if products viewed and bought contains same product IDs.

SK_product_dim (PK)	Surrogate key of the product dimension
Asin	Unique identifier of each product
Title	Title of the product
Description	Description of the product
Image_url	Image url of the product
Also_viewed	List of product asin which were also viewed
Also_bought	List of product asin which were also bought
Bought_together	List of product asin which were bought together
Viewed_and_bought	Asins viewed and bought together
Viewed_and_bought_flag	Flag to indicate if there are any such products which were viewed and bought

3. **Product Category Dimension**
   Contains product category and sub-category related information extracted from the list of categories available in the data.

Sk_product_category_dim (PK)	Surrogate key of the product category dimension
Category	Represents the parent category
Sub_category	Represents the sub-category

4. Price Bucket Dimension

Price buckets created based on the product distribution in the different price buckets. One time manually populated.

Sk_price_bucket_dim (PK)	Surrogate key of the price bucket dimension
Bucket_name	Name of the bucket
Bucket_lower_limit	Lower price limit
Bucket_upper_limit	Upper price limit

5. Reviewer Dimension

To keep track of historical changes in the reviewer name, slowly changing dimension type-2 has been created.

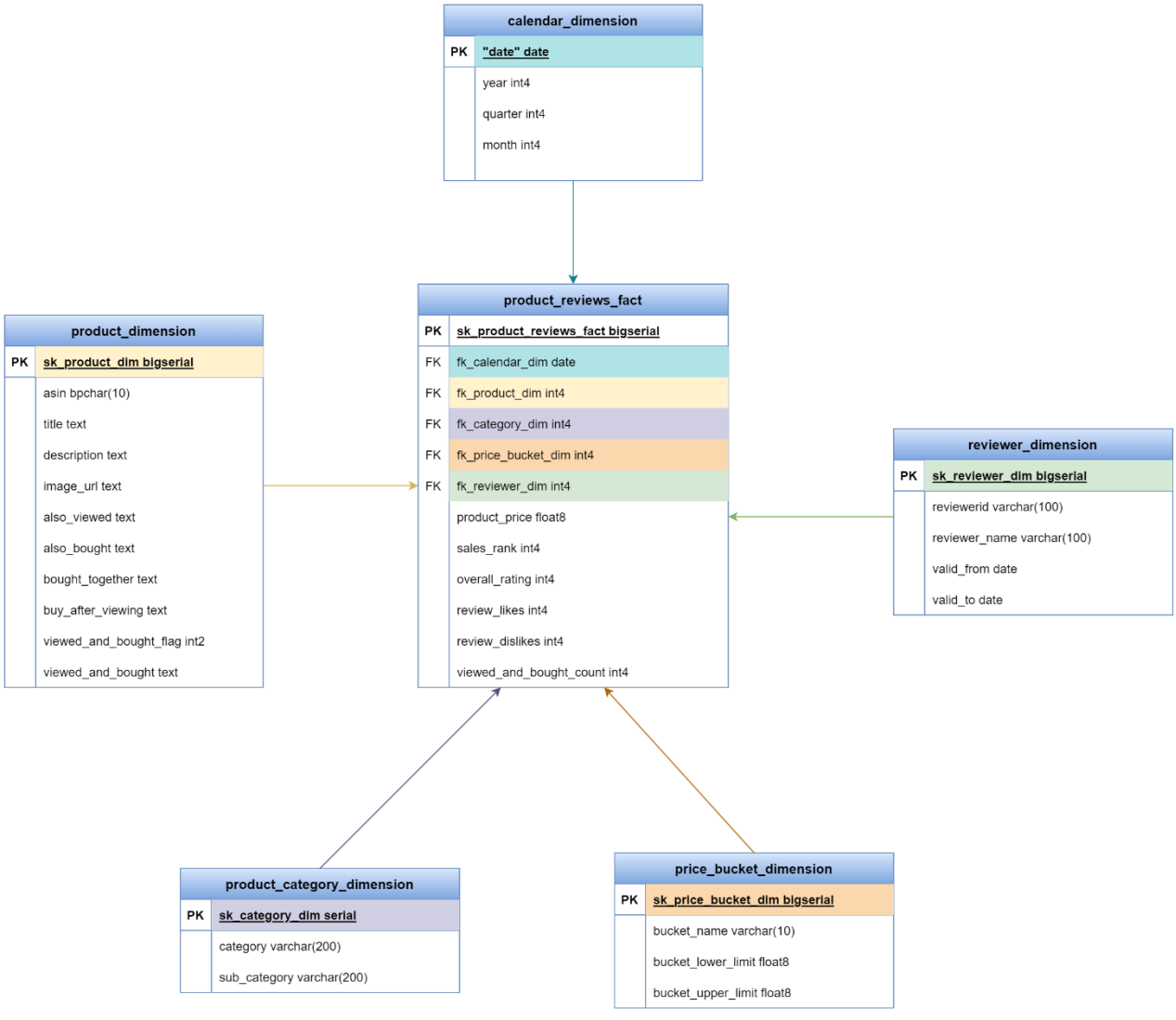
Sk_reviewer_dim (PK)	Surrogate key of the reviewer dimension
Reviewer_id	Identifier assigned to each reviewer
Reviewer_name	Name of the reviewer
Valid_from	Effective date
Valid_to	Expiry date

6. Product Reviews Fact Table

Products joined with all the available dimension tables and reviews data, to create a consolidated fact table consisting of foreign keys of all the dimension tables and multiple quantitative measures which allows us to answer questions such as:

What is the average rating of a particular product?
What is the distribution of ratings for a particular product category?
What is the correlation between the price of a product and its overall rating?
What is the average price of products that received a certain rating?
And many more….

Sk_product_reviews_fact (PK)	Surrogate key of the product reviews dimension
Fk_calendar_dim	Foreign key of calendar dimension
Fk_product_dim	Foreign key of product dimension
Fk_product_category_dim	Foreign key of product category dimension
Fk_price_bucket_dim	Foreign key of price bucket dimension
Fk_reviewer_dim	Foreign key of reviewer dimension
Product_price	Price of the product
Sales_rank	Sales rank of the product
Overall_rating	Overall rating of the product out of 5
Review_likes	Likes given on the review
Review_dislikes	Dislikes given on the review
Viewed_and_bought_count	Counts of products where were viewed and bought



Entity Relation Diagram

Automated Testing and logging

DBT provides a framework for automating the testing of data transformations, which helps ensure that your data is accurate, consistent, and reliable. Testing cases such as not null, unique, primary constraints are automatically tested when defined in model configs. Also, maintains logs for each step.

Data Flow Process Development

Data flow job developed using Talend Open Studio by defining triggering mechanism for each step of the flow. Helps on monitoring the job by capturing logs and stats about the execution and cause of failure of the each component of the job when enabled with the ability to schedule job multiple times a day.