Lab 2: Reporting DB

To complete the task, you will need to know about

1. Denormalization of database with attribute/column redundancy. Evolutionary Database Design Practices.
2. Design a Reporting database.

Previous Task

You have completed the Evolutionary Database design for the Kids shop application. Now, you have the following design.
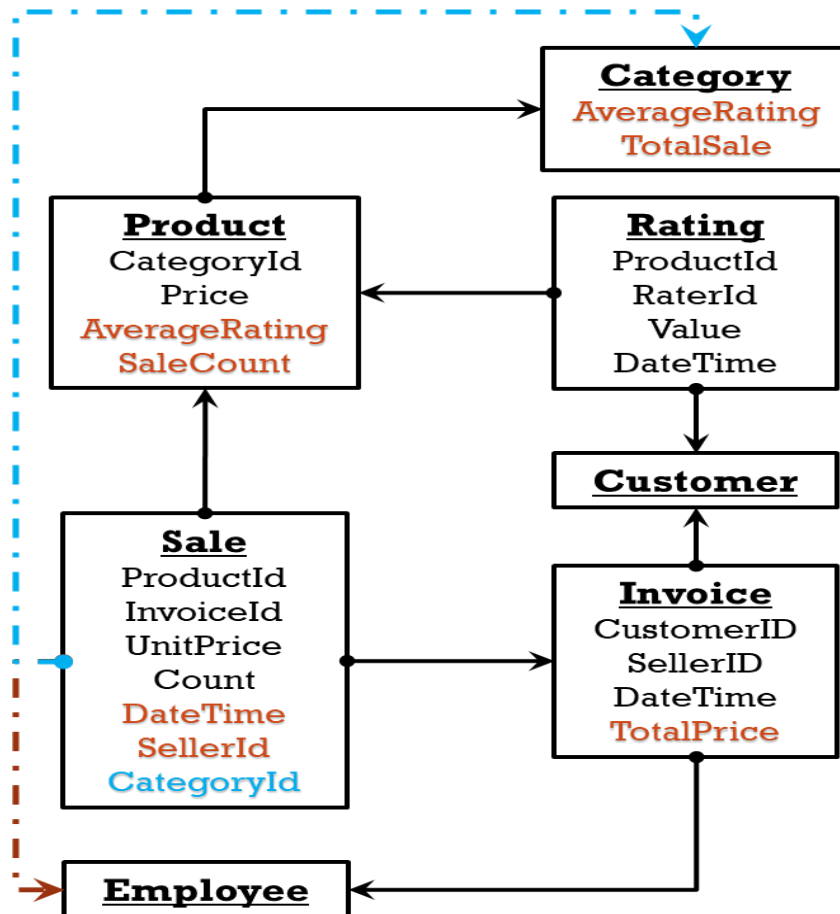

Figure: Denormalized Database design.

If you did not maintain `change_log` for evolutionary practice, you should complete the following:

- Create a change_log schema with automatic `id` along with `applied_at`, `created_by`, `script_name`, and `script_details`.
- Script name should have a prefix for serial number—for example, 000_init_schema.sql. The change_log file should be the first script.
- For every previous task, you have a separate SQL file. An insertion to the change_log table should be added at the end of the script.

Reporting database

- Task: Create the star schema that covers the below queries. When creating the star schema, use prefixes to distinguish between the fact table and dimension tables. For example, you may use `fact_sale` name as a table name. You have to identify the fact and dimension tables for your star schema.
- Create a script to copy data from the operational DB to the reporting DB. As you introduced denormalization in the operational database, select only the relevant columns that would be needed to fulfill the queries.
- For each query, write a stored procedure with the mentioned name.
  a. `get_top_3_products` - find the top 3 products
  b. `get_top_2_categories` - find the top 2 categories
  c. `get_top_product_by_duration(start_date: DATE, end_date: DATE)` - find the product with the top rating in the given date range
  d. `get_top_product_by_category(category_id)` - find the product with the top rating in the given category
  e. `get_top_employee_by_duration(start_date: DATE, end_date: DATE)` - find the top employee with the highest sales in the given date range.