**Project: Database Design and Implementation for Coffee Shop Marketing**

New York based coffee shop chain that is looking to expand nationally by opening a number of franchise locations.

The objective is to design their relational database systems for improved operational efficiencies and to make it easier for their executives to make data driven decisions.

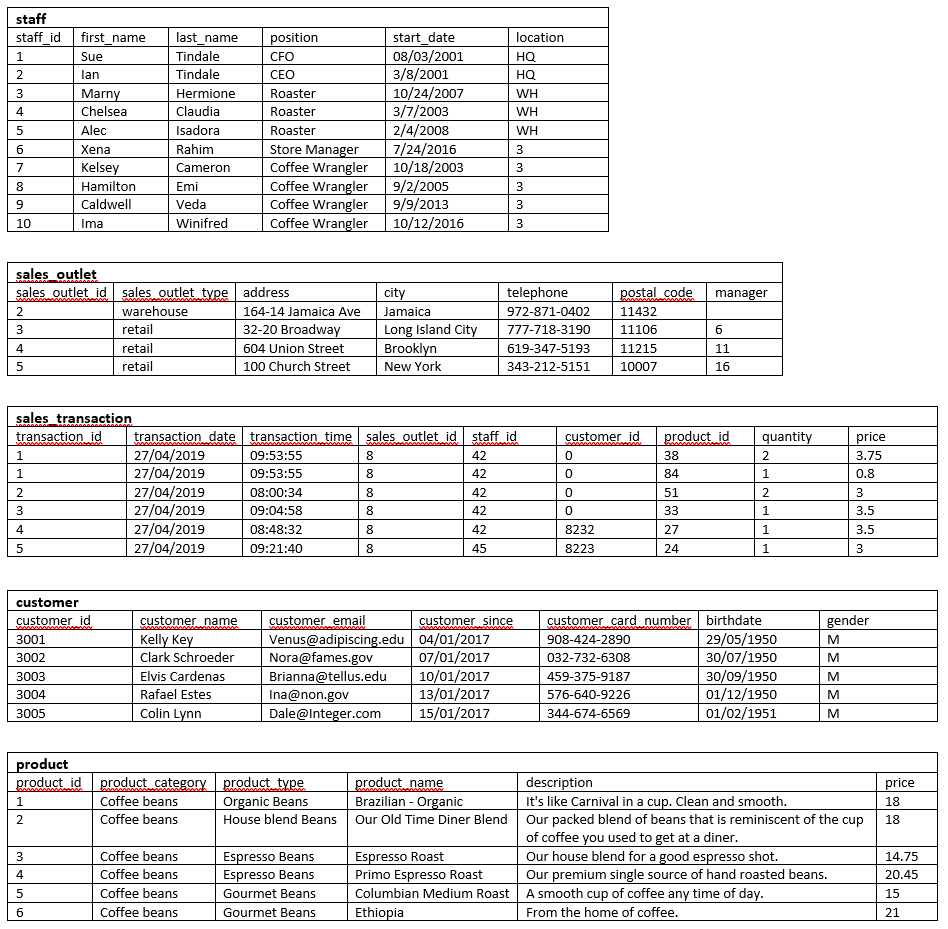
The data resides in several different systems: accounting software, suppliers’ databases, point of sales (POS) systems, and even spreadsheets.

Therefore, i review the data in all of these systems, design a central database to house all of the data and I create the database objects, load them with source data, and create subsets of data that your business partners require, export them, and then load them into staging databases that use different RDBMS.

In scenario, I worked with data from the following sources:

* Staff information held in a spreadsheet at HQ
* Sales outlet information held in a spreadsheet at HQ
* Sales data output as a CSV file from the POS system in the sales outlets
* Customer data output as a CSV file from a bespoke customer relationship management system
* Product information maintained in a spreadsheet exported from your supplier's database

The following image shows sample data from each of the data sources that you will be working with to design your new central database.



**Task 1: Identify entities**

**Task 2: Identify attributes**

**Task 3: Create an ERD**

* I used the PostgreSQL database ( pgAdmin tool),
* I create a new database named COFFEE and tables using entities shown in the image above.

**Task 4: Normalize tables**

* Determine which columns should be stored in a separate table to remove the repeating rows and to put this table into second normal form.
* Determine which columns should be stored in a separate table to reduce redundant data and to put this table into second normal form.
* I created 2 new tables named product type and sales detail for and delete the moved columns from the sales transaction and product table leaving a matching column in each of two tables to later create a relationship between them.

**Task 5: Define keys and relationships**

* After normalizing tables, i define their primary keys and define relationships between the tables in ERD.

**Task 6: Create database objects by generating and running the SQL script from the ERD Tool**

* I generated an SQL script from ERD, which you could use to create your database schema.

**Task 7: Create a view and export the data**

The external payroll company have requested a list of employees and the locations at which they work. This should not include the CEO or CFO who own the company.

In this task, i created a view in PostgreSQL database that returns this information and export the results to a CSV file using the query bellow and save as CSV file “staff\_locations\_view.csv”:

**SELECT STAFF.STAFF\_ID,**

**STAFF.FIRST\_NAME,**

**STAFF.LAST\_NAME,**

**STAFF.LOCATION**

**FROM STAFF**

**WHERE "POSITION" NOT IN ('CEO', 'CFO');**

**Task 8: Create a materialized view and export the data**

* A marketing consultant requires access the product data in their MySQL database for a marketing campaign. I created a materialized view in PostgreSQL database that returns this information using query bellow and export the results to a CSV file “product\_info\_m-view.csv”.

**SELECT PRODUCT.PRODUCT\_NAME, PRODUCT.DESCRIPTION, PRODUCT\_TYPE.PRODUCT\_CATEGORY**

**FROM PRODUCT**

**JOIN PRODUCT\_TYPE**

**ON PRODUCT.PRODUCT\_TYPE\_ID = PRODUCT\_TYPE.PRODUCT\_TYPE\_ID;**

**Task 9: Import data into a Db2 database**

* The external payroll company have asked to upload the staff location information to their Db2 database. So, I load a new table named STAFF\_LOCATIONS with the staff location information saved in the staff\_locations\_view.csv file that I exported from the view you created in Task 7.

## **Task 10: Import data into a MySQL database**

* The marketing consultant has asked to upload the product information to their MySQL database. So, i used phpMyAdmin, create a new database named coffee\_shop\_products, and then import the product information saved in the product\_info\_m-view.csv file from materialized view into a new table in the coffee\_shop\_products database.