Foundations of Data Management

Lab Exercise 02

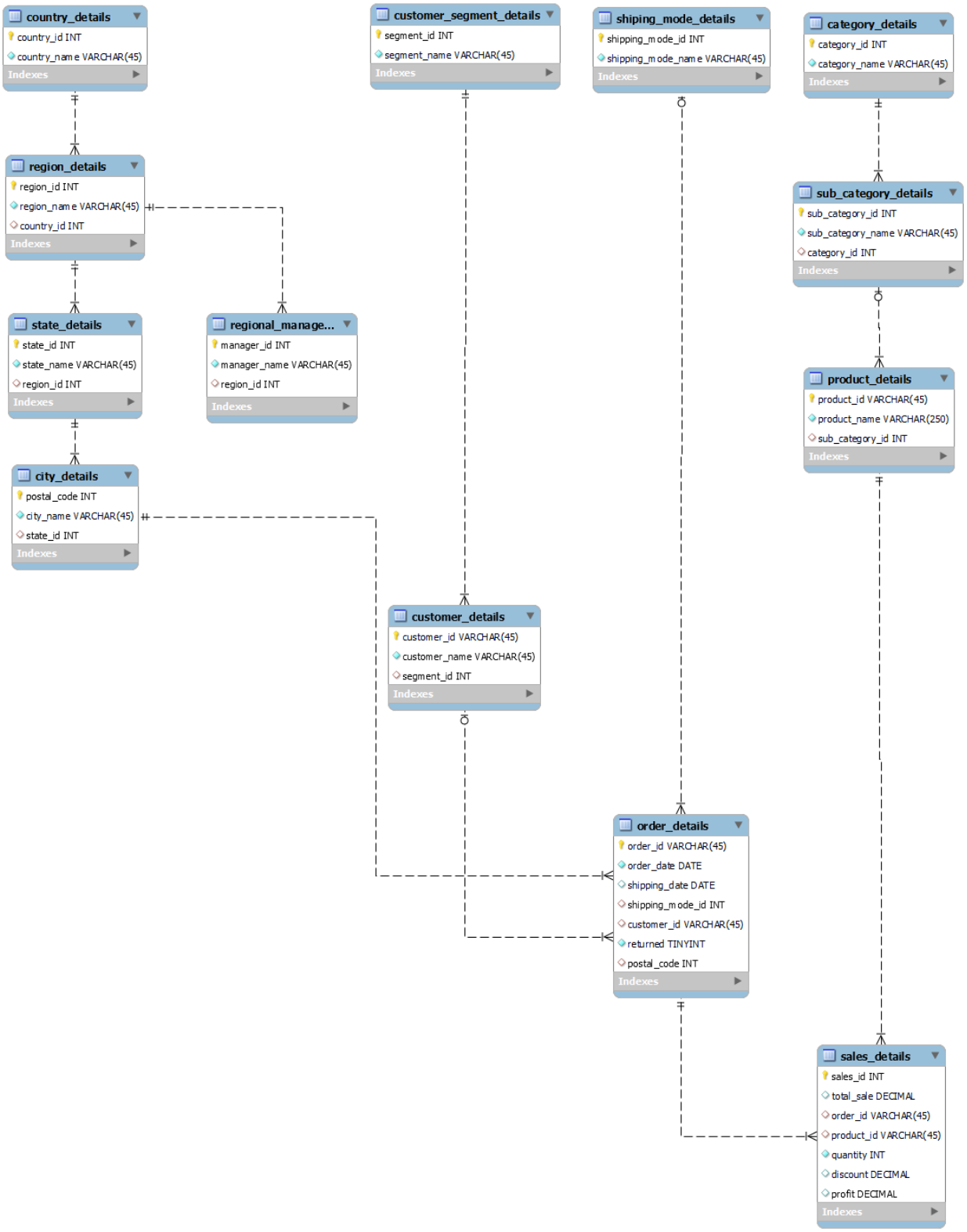
Prepared by:

Group 07

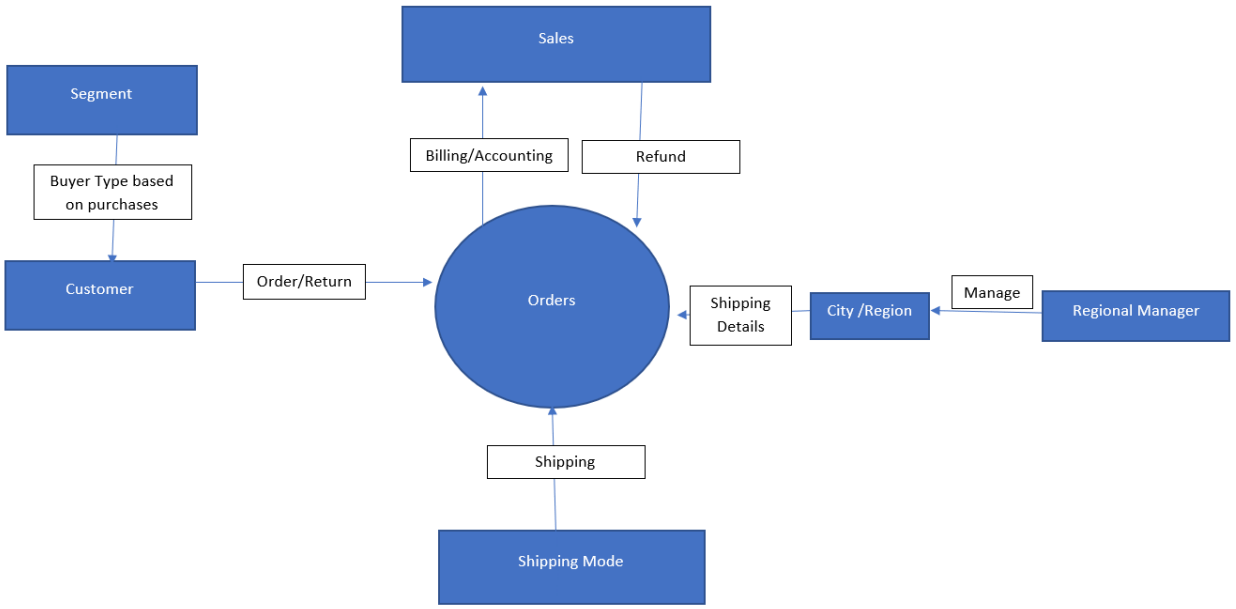
Student Names
Garv Balotia
Rajanbir Singh Sethi
Harpreet Kaur
Sukhsimar Singh
Sean Tran
Kautak Udavant
Sertan Avdan

October 14, 2022

Logical-level ERD

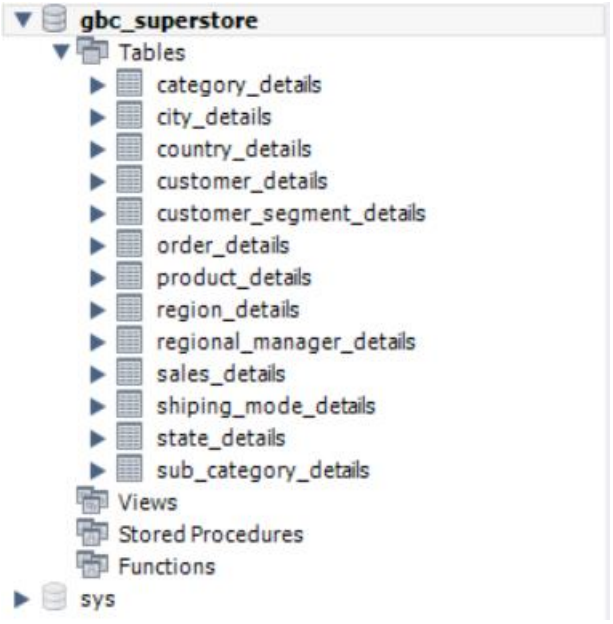


Data Flow


Sales
Segment
Billing/Accounting
Refund
Buyer Type based on purchases
Customer
Order/Return
Orders
Shipping Details
City /Region
Manage
Regional Manager
Shipping
Shipping Mode

Database Schema

Following is the screenshot of the tables created in SQL workbench and short description of all the tables provided in the below schema.



TABLE_NAME	COLUMN_NAME	DATA_TYPE	COLUMN_TYPE	ORDINAL_POSITION
category_details	category_id	int	int	1
category_details	category_name	varchar	varchar(45)	2
city_details	postal_code	int	int	1
city_details	city_name	varchar	varchar(45)	2
city_details	state_id	int	int	3
country_details	country_id	int	int	1
country_details	country_name	varchar	varchar(45)	2
customer_details	customer_id	varchar	varchar(45)	1
customer_details	customer_name	varchar	varchar(45)	2
customer_details	segment_id	int	int	3
customer_segment_details	segment_id	int	int	1
customer_segment_details	segment_name	varchar	varchar(45)	2
order_details	order_id	varchar	varchar(45)	1
order_details	order_date	date	date	2
order_details	shipping_date	date	date	3
order_details	shipping_mode_id	int	int	4
order_details	customer_id	varchar	varchar(45)	5
order_details	returned	tinyint	tinyint	6

order_details	postal_code	int	int	7
product_details	product_id	varchar	varchar(45)	1
product_details	product_name	varchar	varchar(250)	2
product_details	sub_category_id	int	int	3
region_details	region_id	int	int	1
region_details	region_name	varchar	varchar(45)	2
region_details	country_id	int	int	3
regional_manager_details	manager_id	int	int	1
regional_manager_details	manager_name	varchar	varchar(45)	2
regional_manager_details	region_id	int	int	3
sales_details	sales_id	int	int	1
sales_details	total_sale	decimal	decimal(10,0)	2
sales_details	order_id	varchar	varchar(45)	3
sales_details	product_id	varchar	varchar(45)	4
sales_details	quantity	int	int	5
sales_details	discount	decimal	decimal(10,0)	6
sales_details	profit	decimal	decimal(10,0)	7
shiping_mode_details	shipping_mode_id	int	int	1
shiping_mode_details	shipping_mode_name	varchar	varchar(45)	2
state_details	state_id	int	int	1
state_details	state_name	varchar	varchar(45)	2
state_details	region_id	int	int	3
sub_category_details	sub_category_id	int	int	1
sub_category_details	sub_category_name	varchar	varchar(45)	2
sub_category_details	category_id	int	int	3

Note: For more descriptive schema details, refer to “Database Schema.xml”.

Data Source Changes

Below steps were performed to the excel sheet (Sample – Superstore):

- Make Product Name consistent everywhere in the excel workbook for Product Ids:
    1. FUR-CH-10001146
    2. OFF-PA-10001970
    3. OFF-ST-10001228
    4. TEC-AC-10003832
    5. OFF-PA-10002377
    6. TEC-PH-10001530
    7. OFF-AP-10000576
    8. OFF-BI-10004632
    9. FUR-FU-10004091
    10. TEC-AC-10002049
    11. OFF-PA-10000357
    12. FUR-FU-10001473
    13. OFF-PA-10000477
    14. OFF-PA-10000659
    15. FUR-FU-10004848
    16. FUR-BO-10002213
    17. TEC-PH-10002310
    18. FUR-FU-10004270
    19. TEC-PH-10004531
    20. OFF-BI-10004654
    21. TEC-PH-10002200
    22. OFF-ST-10004950
    23. TEC-MA-10001148
    24. TEC-AC-10002550
    25. OFF-PA-10002195
    26. FUR-FU-10004017
    27. OFF-PA-10001166
    28. FUR-FU-10004864
    29. OFF-PA-10003022
    30. TEC-PH-10001795
    31. OFF-BI-10002026
- Make City Name consistent for Postal Code 92024.
- Filling Postal Code for Burlington, Vermont, United States with 5401 everywhere (referring postal codes from internet).

ETL Process

Extraction:

- Reading the excel file using pandas library.
- Splitting each sheet into separate DataFrames.
- Preprocessing:
  - In the returns DataFrame, kept only those Order Ids which were returned by the customers.
- Extraction process leaves us with 3 DataFrames of each worksheet from the workbook “Sample – Superstore”.

Transformation:

- Splitting the 3 DataFrames generated from Extraction into smaller DataFrames that are required to load our tables.
- Country Details Dataframe contains unique values of country/region from Orders worksheet.
- Customer Segment Details Dataframe contains unique values of segment from Orders worksheet.
- Shipping Mode Details Dataframe contains unique values of Ship Mode from Orders worksheet.
- Category Details Dataframe contains unique values of category from Orders worksheet.
- Region Details DataFrame contains distinct pairs of Region and Country/Region from Orders worksheet.
- Sub-category Details DataFrame contains distinct pairs of sub-category and category from Orders worksheet.
- State Details DataFrame contains distinct pairs of State and Region from Orders worksheet.
- Regional Manager Details DataFrame contains distinct pairs of Manager Name and Region Name from People worksheet.
- Product Details DataFrame contains distinct triples of Product Id, Product Name and Sub-category from Orders worksheet.
- Product Details DataFrame contains distinct triples of Postal Code, City and State from Orders worksheet.
- Customer Details DataFrame contains distinct triples of Customer Id, Customer Name and Segment from Orders worksheet.
- Order Details DataFrame contains distinct sets of Order Id, Order Date, Shipping Date, Shipping Mode, Customer ID and Postal Code from Orders worksheet and a returned column is set to 0/1 based on if the Order Id is present in Returns worksheet with a value “Yes”.
- Sales Details DataFrame contains distinct sets of Sales, Order Id, Product Id, Quantity, Discount and Profit from Orders worksheet.
- Transformation returns us 13 DataFrames corresponding to each of the tables that have to be loaded into the schema “gbc_Superstore”.

Loading

- We will follow the below flow and queries to load our Data Frames to the "gbc_Superstore" schema.
    - Country details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `country_details`
        (`country_name`)
        VALUES
        (%s);'''
```

- Customer details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `customer_segment_details`
        (`segment_name`)
        VALUES
        (%s);'''
```

- Shipping mode details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `shiping_mode_details`
        (`shipping_mode_name`)
        VALUES
        (%s);'''
```

- Category details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `category_details`
          (`category_name`)
           VALUES
           (%s);'''
```

- Region details Data Frame is loaded into the Database using the below query:

```
   '''INSERT INTO `region_details`
           (`region_name`,
           `country_id`)
           VALUES
           (%s,
           (SELECT `country_id` from `country_details` where `country_name` = %s));
           '''
```

- Sub-category details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `sub_category_details`
        (`sub_category_name`,
        `category_id`)
        VALUES
        ( %s,
        (SELECT `category_id` FROM `category_details` where `category_name` = %s ));
      '''
```

- State details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `state_details`
        (`state_name`,
        `region_id`)
        VALUES
        ( %s,
        (SELECT `region_id` from `region_details` where `region_name` = %s));
        '''
```

- Regional manager details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `regional_manager_details`
        (`manager_name`,
        `region_id`)
        VALUES
        ( %s,
        (SELECT `region_id` FROM `region_details` where `region_name` = %s));
        '''
```

- Product details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `product_details`
        (`product_id`,
        `product_name`,
        `sub_category_id`)
        VALUES
        ( %s,
        %s,
        (SELECT `sub_category_id` FROM `sub_category_details` WHERE
      `sub_category_name` = %s));
      '''
```

- City details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `city_details`
        (`postal_code`,
        `city_name`,
        `state_id`)
        VALUES
        ( %s,
        %s,
        (SELECT `state_id` FROM `state_details` WHERE `state_name` = %s ));
      '''
```

- Customer details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `customer_details`
        (`customer_id`,
        `customer_name`,
        `segment_id`)
        VALUES
        ( %s,
        %s,
       (SELECT `segment_id` FROM `customer_segment_details` WHERE `segment_name` =
%s));'''
```

- Order details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `order_details`
        (`order_id`,
        `order_date`,
        `shipping_date`,
        `shipping_mode_id`,
        `customer_id`,
        `postal_code`,
        `returned`)
        VALUES
        (%s,
        %s,
        %s,
        (SELECT `shipping_mode_id` FROM `shiping_mode_details` WHERE
`shipping_mode_name` = %s),
        %s,
        %s,
        %s);'''
```

- Sales details Data Frame is loaded into the Database using the below query:

```
'''INSERT INTO `sales_details`
        (
        `total_sale`,
        `order_id`,
        `product_id`,
        `quantity`,
        `discount`,
        `profit`)
        VALUES
        (
        %s,
        %s,
        %s,
        %s,
        %s,
        %s);
        '''
```

- The loading process results in all the 13 tables loaded with the data as per the schema.

This concludes the ETL process as the data has been extracted, got transformed and loaded to the Database.