Cleaning a PostgreSQL Database

In this project, you will work with data from a hypothetical Super Store to challenge and enhance your SQL skills in data cleaning. This project will engage you in identifying top categories based on the highest profit margins and detecting missing values, utilizing your comprehensive knowledge of SQL concepts.

Data Dictionary:

`orders`:

Column	Definition	Data type	Comments
`row_id`	Unique Record ID	`INTEGER`	
`order_id`	Identifier for each order in table	`TEXT`	Connects to `order_id` in `returned_orders` table
`order_date`	Date when order was placed	`TEXT`	
`market`	Market order_id belongs to	`TEXT`	
`region`	Region Customer belongs to	`TEXT`	Connects to `region` in `people` table
`product_id`	Identifier of Product bought	`TEXT`	Connects to `product_id` in `products` table
`sales`	Total Sales Amount for the Line Item	`DOUBLE PRECISION`	
`quantity`	Total Quantity for the Line Item	`DOUBLE PRECISION`	
`discount`	Discount applied for the Line Item	`DOUBLE PRECISION`	
`profit`	Total Profit earned on the Line Item	`DOUBLE PRECISION`	

`returned_orders`:

Column	Definition	Data type
`returned`	Yes values for Order / Line Item Returned	`TEXT`
`order_id`	Identifier for each order in table	`TEXT`

Column	Definition	Data type
`market`	Market order_id belongs to	`TEXT`

`people`:

Column	Definition	Data type
`person`	Name of Salesperson credited with Order	`TEXT`
`region`	Region Salesperson in operating in	`TEXT`

`products`:

Column	Definition	Data type
`product_id`	Unique Identifier for the Product	`TEXT`
`category`	Category Product belongs to	`TEXT`
`sub_category`	Sub Category Product belongs to	`TEXT`
`product_name`	Detailed Name of the Product	`TEXT`

As you can see in the Data Dictionary above, date fields have been written to the `orders` table as `TEXT` and numeric fields like sales, profit, etc. have been written to the `orders` table as `Double Precision`. You will need to take care of these types in some of the queries. This project is an excellent opportunity to apply your SQL skills in a practical setting and gain valuable experience in data cleaning and analysis. Good luck, and happy querying!

Projects Data DataFrame as `top_five_products_each_category`

```sql
-- top_five_products_each_category
SELECT * FROM (
SELECT products.category,
    products.product_name,
    ROUND(SUM(CAST(ord.sales AS NUMERIC)), 2) AS product_total_sales,
    ROUND(SUM(CAST(ord.profit AS NUMERIC)), 2) AS product_total_profit,
    RANK() OVER(PARTITION BY products.category ORDER BY SUM(ord.sales) DESC) AS product_rank
    FROM orders AS ord
    INNER JOIN products
        ON ord.product_id = products.product_id
    GROUP BY products.category, products.product_name
) AS tmp
WHERE product_rank < 6;
```

	category	product_name	product_total_sales
0	Furniture	Hon Executive Leather Armchair, Adjustable	58
1	Furniture	Office Star Executive Leather Armchair, Adjustable	5
2	Furniture	Harbour Creations Executive Leather Armchair, Adjustable	50
3	Furniture	SAFCO Executive Leather Armchair, Black	41
4	Furniture	Novimex Executive Leather Armchair, Adjustable	40
5	Office Supplies	Eldon File Cart, Single Width	39
6	Office Supplies	Hoover Stove, White	3
7	Office Supplies	Hoover Stove, Red	32
8	Office Supplies	Rogers File Cart, Single Width	29
9	Office Supplies	Smead Lockers, Industrial	28
10	Technology	Apple Smart Phone, Full Size	86
11	Technology	Cisco Smart Phone, Full Size	76
12	Technology	Motorola Smart Phone, Full Size	7
13	Technology	Nokia Smart Phone, Full Size	71
14	Technology	Canon imageCLASS 2200 Advanced Copier	61

15 rows

Projects Data DataFrame as i

```sql
-- impute_missing_values
WITH missing AS (
    SELECT product_id,
        discount,
        market,
        region,
        sales,
        quantity
    FROM orders
    WHERE quantity IS NULL
),

unit_prices AS (SELECT o.product_id,
    CAST(o.sales / o.quantity AS NUMERIC) AS unit_price
FROM orders o
RIGHT JOIN missing AS m
    ON o.product_id = m.product_id
    AND o.discount = m.discount
WHERE o.quantity IS NOT NULL
)

SELECT DISTINCT m.*,
    ROUND(CAST(m.sales AS NUMERIC) / up.unit_price,0) AS calculated_quantity
FROM missing AS m
INNER JOIN unit_prices AS up
```

```
    ON m.product_id = up.product_id;
```

	product_id	discount	market	region	sales