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`
`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

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_p

```
-- top_five_products_each_category
select *
    from (select
    category,
    product_name,
    round(sum(cast(sales as numeric)), 2) as product_total_sales,
    round(sum(cast (profit as numeric)), 2) as product_total_profit,
    rank () over (partition by category order by sum(o.sales) desc) as product_rank
from products as p
inner join orders as o
on p.product_id = o.product_id
group by category, product_name
    ) as tmp
    where product_rank < 6;
```

	category	product_name	product_total_sales	product_total_profit	produ…
0	Furniture	Hon Executive Leather Armchair, Adjustable	58193.48	5997.25	
1	Furniture	Office Star Executive Leather Armchair, Adjusta…	51449.8	4925.8	
2	Furniture	Harbour Creations Executive Leather Armchair, …	50121.52	10427.33	
3	Furniture	SAFCO Executive Leather Armchair, Black	41923.53	7154.28	
4	Furniture	Novimex Executive Leather Armchair, Adjustable	40585.13	5562.35	
5	Office Supplies	Eldon File Cart, Single Width	39873.23	5571.26	
6	Office Supplies	Hoover Stove, White	32842.6	-2180.63	
7	Office Supplies	Hoover Stove, Red	32644.13	11651.68	
8	Office Supplies	Rogers File Cart, Single Width	29558.82	2368.82	
9	Office Supplies	Smead Lockers, Industrial	28991.66	3630.44	
10	Technology	Apple Smart Phone, Full Size	86935.78	5921.58	
11	Technology	Cisco Smart Phone, Full Size	76441.53	17238.52	
12	Technology	Motorola Smart Phone, Full Size	73156.3	17027.11	
13	Technology	Nokia Smart Phone, Full Size	71904.56	9938.2	
14	Technology	Canon imageCLASS 2200 Advanced Copier	61599.82	25199.93	

Rows: 15 Expand

Projects Data DataFrame as i

```
-- 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 as 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 •••	•••	•••	•••	•••	•••	calculated_quan... •••
0	FUR-ADV-10000571	0	EMEA	EMEA	438.96		4
1	FUR-ADV-10004395	0	EMEA	EMEA	84.12		2
2	FUR-BO-10001337	0.15	US	West	308.499		3
3	TEC-STA-10003330	0	Africa	Africa	506.64		2
4	TEC-STA-10004542	0	Africa	Africa	160.32		4

Rows: 5 Expand