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
`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!

Final Solution

Projects Data DataFrame as `top_five_products_each_category`

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

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

Rows: 15 Expand

Projects Data DataFrame as `d`

```
SELECT *
FROM orders
```

	order_id	order_date	ship_date	ship_mo...	cus...	customer_name	se...	city
0	957	MX-2014-105921	2014-05-28T00:00:00.000	2014-06-03T00:00:00.000	Standard Class	ZC-21910	Zuschuss Carroll	Consumer	San Salvador
1	24359	ID-2013-61442	2013-01-15T00:00:00.000	2013-01-21T00:00:00.000	Standard Class	JB-16000	Joy Bell-	Consumer	Manila
2	32298	CA-2012-124891	2012-07-31T00:00:00.000	2012-07-31T00:00:00.000	Same Day	RH-19495	Rick Hansen	Consumer	New York City
3	26341	IN-2013-77878	2013-02-05T00:00:00.000	2013-02-07T00:00:00.000	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong
4	25330	IN-2013-71249	2013-10-17T00:00:00.000	2013-10-18T00:00:00.000	First Class	CR-12730	Craig Reiter	Consumer	Brisbane
5	13524	ES-2013-1579342	2013-01-28T00:00:00.000	2013-01-30T00:00:00.000	First Class	KM-16375	Katherine Murray	Home Office	Berlin
6	47221	SG-2013-4320	2013-11-05T00:00:00.000	2013-11-06T00:00:00.000	Same Day	RH-9495	Rick Hansen	Consumer	Dakar
7	22732	IN-2013-42360	2013-06-28T00:00:00.000	2013-07-01T00:00:00.000	Second Class	JM-15655	Jim Mitchum	Corporate	Sydney
8	30570	IN-2011-81826	2011-11-07T00:00:00.000	2011-11-09T00:00:00.000	First Class	TS-21340	Toby Swindell	Consumer	Porirua
9	31192	IN-2012-86369	2012-04-14T00:00:00.000	2012-04-18T00:00:00.000	Standard Class	MB-18085	Mick Brown	Consumer	Hamilton
10	40155	CA-2014-135909	2014-10-14T00:00:00.000	2014-10-21T00:00:00.000	Standard Class	JW-15220	Jane Waco	Corporate	Sacramento
11	40936	CA-2012-116638	2012-01-28T00:00:00.000	2012-01-31T00:00:00.000	Second Class	JH-15985	Joseph Holt	Consumer	Concord
12	34577	CA-2011-102988	2011-04-05T00:00:00.000	2011-04-09T00:00:00.000	Second Class	GM-14695	Greg Maxwell	Corporate	Alexandria
13	28879	ID-2012-28402	2012-04-19T00:00:00.000	2012-04-22T00:00:00.000	First Class	AJ-10780	Anthony Jacobs	Corporate	Kabul
14	45794	SA-2011-1830	2011-12-27T00:00:00.000	2011-12-29T00:00:00.000	Second Class	MM-7260	Magdelene Morse	Consumer	Jizan

Rows: 4,761 ⚠ Truncated from 51,290 rows Expand

Projects Data DataFrame as `i`

```sql
-- impute_missing_values
WITH missing AS
(SELECT *
FROM orders
WHERE quantity IS NULL ),

unit_prices AS
(
SELECT product_id,
    SUM(sales) AS total_sales,
    SUM(quantity) AS total_quantity,
    CAST(SUM(sales) / SUM(quantity) AS NUMERIC) AS unit_price
FROM orders
WHERE quantity IS NOT NULL
    AND quantity != 0
GROUP BY product_id
)

SELECT DISTINCT m.product_id,
    m.discount,
    m.market,
    m.region,
    m.sales,
    m.quantity,
    ROUND(CAST((m.sales / u.unit_price) AS NUMERIC ),0) AS calculated_quantity
FROM missing AS m
LEFT JOIN unit_prices as u
ON m.product_id = u.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		4
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

Data Exploration

Notes

To Do List

1. Find top 5 products from each category with highest sales
    - Sorted by category in ascending order and
    - Sorted by sales in descending order
2. Impute missing values in quantity colum by determining the unit price of each product

1 hidden cell

Projects Data DataFrame as `d`

```
--- Test Area
WITH missing AS
(SELECT *
FROM orders
WHERE quantity IS NULL ),

unit_prices AS
(
SELECT product_id,
    SUM(sales) AS total_sales,
    SUM(quantity) AS total_quantity,
    CAST(SUM(sales) / SUM(quantity) AS NUMERIC) AS unit_price
FROM orders
WHERE quantity IS NOT NULL
    AND quantity != 0
GROUP BY product_id
)

SELECT DISTINCT m.product_id,
    m.discount,
    m.market,
    m.region,
    m.sales,
    u.unit_price,
    ROUND(CAST((m.sales / u.unit_price) AS NUMERIC ),0) AS calculated_quantity
FROM missing AS m
LEFT JOIN unit_prices as u
ON m.product_id = u.product_id
```

	product_id					unit_price	calculated_quan...
0	FUR-ADV-10000571	0	EMEA	EMEA	438.96	109.74	4
1	FUR-ADV-10004395	0	EMEA	EMEA	84.12	19.628	4
2	FUR-BO-10001337	0.15	US	West	308.499	106.4624	3
3	TEC-STA-10003330	0	Africa	Africa	506.64	229.9366153846	2
4	TEC-STA-10004542	0	Africa	Africa	160.32	38.8143157895	4

Rows: 5 Expand

Projects Data DataFrame as

```
SELECT *
FROM orders
WHERE quantity IS NULL
```

		order_id	order_date	ship_date	ship_mo...	cus...	custo...	s.	city	state
0	50688	IV-2013-8930	2013-09-04T00:00:00.000	2013-09-04T00:00:00.000	Same Day	MJ-7740	Max Jones	Consumer	Abidjan	Lagun
1	43428	RS-2011-1760	2011-11-29T00:00:00.000	2011-12-04T00:00:00.000	Standard Class	TH-11235	Tiffany House	Corporate	Severodvinsk	Arkha
2	34093	CA-2011-154599	2011-04-12T00:00:00.000	2011-04-17T00:00:00.000	Standard Class	KN-16450	Kean Nguyen	Corporate	Redondo Beach	Califo
3	46824	AG-2014-9060	2014-12-27T00:00:00.000	2015-01-01T00:00:00.000	Standard Class	YS-11880	Yana Sorensen	Corporate	Algiers	Alger
4	43722	RS-2011-3610	2011-12-08T00:00:00.000	2011-12-10T00:00:00.000	First Class	ND-8460	Neil Ducich	Corporate	Usol'ye-Sibirskoye	Irkutsk

Rows: 5 Expand

Projects Data DataFrame as

```
SELECT product_id,
    SUM(sales) AS total_sales,
    SUM(quantity) AS total_quantity,
    CAST(SUM(sales) / SUM(quantity) AS NUMERIC) AS unit_price
FROM orders
WHERE quantity IS NOT NULL
    AND quantity != 0
GROUP BY product_id
```

	product_id	tot...	total_qu...	unit_price
0	OFF-SME-10000950	4.032	1	4.032
1	TEC-CIS-10001717	5691.888	11	517.4443636364
2	OFF-ST-10002905	427.2	24	17.8
3	OFF-GLO-10000201	38.28	5	7.656
4	OFF-PA-10003656	538.152	21	25.6262857143
5	OFF-SU-10002032	320.852	25	12.83408
6	OFF-AR-10003691	494.68	20	24.734
7	TEC-PH-10002085	752.286	14	53.7347142857
8	FUR-IKE-10002719	725.7	2	362.85
9	OFF-SU-10004279	565.11	15	37.674
10	TEC-AC-10004070	2972.52	12	247.71
11	TEC-PH-10004575	2571.12	4	642.78
12	OFF-EN-10002613	68.352	10	6.8352
13	TEC-AC-10003666	6893.88	31	222.3832258065
14	FUR-CH-10004011	3978	16	248.625
15	FUR-FU-10003394	1469.16	26	56.5061538462

Rows: 10,292 Expand

Projects Data DataFrame as

```
SELECT product_id,
    discount,
    market,
    region,
    sales,
    quantity,
    (sales / quantity) AS calculated_quantity
FROM orders
WHERE quantity IS NULL
```

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

Rows: 5 Expand

Projects Data DataFrame as

```
SELECT *
FROM
(SELECT *
FROM orders AS o
LEFT JOIN products AS p
ON o.product_id = p.product_id
) temp
WHERE temp.quantity IS NULL
```

		order_id	order_date	ship_date	ship_mo…	cus…	custo…	s.	city	state
0	50688	IV-2013-8930	2013-09-04T00:00:00.000	2013-09-04T00:00:00.000	Same Day	MJ-7740	Max Jones	Consumer	Abidjan	Lagun
1	43428	RS-2011-1760	2011-11-29T00:00:00.000	2011-12-04T00:00:00.000	Standard Class	TH-11235	Tiffany House	Corporate	Severodvinsk	Arkha
2	34093	CA-2011-154599	2011-04-12T00:00:00.000	2011-04-17T00:00:00.000	Standard Class	KN-16450	Kean Nguyen	Corporate	Redondo Beach	Califo
3	46824	AG-2014-9060	2014-12-27T00:00:00.000	2015-01-01T00:00:00.000	Standard Class	YS-11880	Yana Sorensen	Corporate	Algiers	Alger
4	43722	RS-2011-3610	2011-12-08T00:00:00.000	2011-12-10T00:00:00.000	First Class	ND-8460	Neil Ducich	Corporate	Usol'ye-Sibirskoye	Irkutsk

Rows: 5 Expand