

TDM	729.89	915.51	185.62	▲25.43%	FLR	660.27	745.28	85.01	▲12.88%
HUM	749.73	924.29	174.56	▲23.28%	UVD	155.59	181.57	25.98	▲16.70%
DMW	833.72	1004.01	170.29	▲20.43%	QUV	440.55	540.21	99.66	▲22.62%
YZJ	903.49	1127.46	223.97	▲24.79%	HZT	285.51	344.98	59.47	▲20.83%
GLY	982.07	1219.39	237.32	▲24.17%	PCW	811.44	1029.66	218.22	▲26.89%
VDA	113.74	143.41	29.67	▲26.09%	AIK	361.77	451.39	89.62	▲24.77%
UVV	468.08	535.41	67.33	▲14.38%	ZJJ	858.36	994.57	136.21	▲15.87%
HJS	545.49	659.05	113.56	▲20.82%	RHJ	894.79	1046.68	151.89	▲16.97%
EQC	566.96	664.69	97.73	▲17.24%	VGV	425.08	509.95	84.87	▲19.97%

PPJ	912.63	1038.36	125.73	▲13.78%	ZBK	391.59	491.48	99.89	▲25.51%
UAQ	1309.55	1655.62	346.07	▲26.43%	BNY	969.21	1130.65	161.44	▲16.66%
DAQ	1295.17	1641.66	346.49	▲26.75%	SDM	735.44	913.39	177.95	▲24.20%
PNR	654.33	775.84	121.51	▲18.57%	TQQ	1323.91	1646.42	322.51	▲24.36%
ZTM	161.99	192.22	30.23	▲18.65%	OIS	543.42	667.24	123.82	▲22.76%
					TTB	1495.17	1823.91	328.74	▲21.94%

Sales Performance Data Pipeline

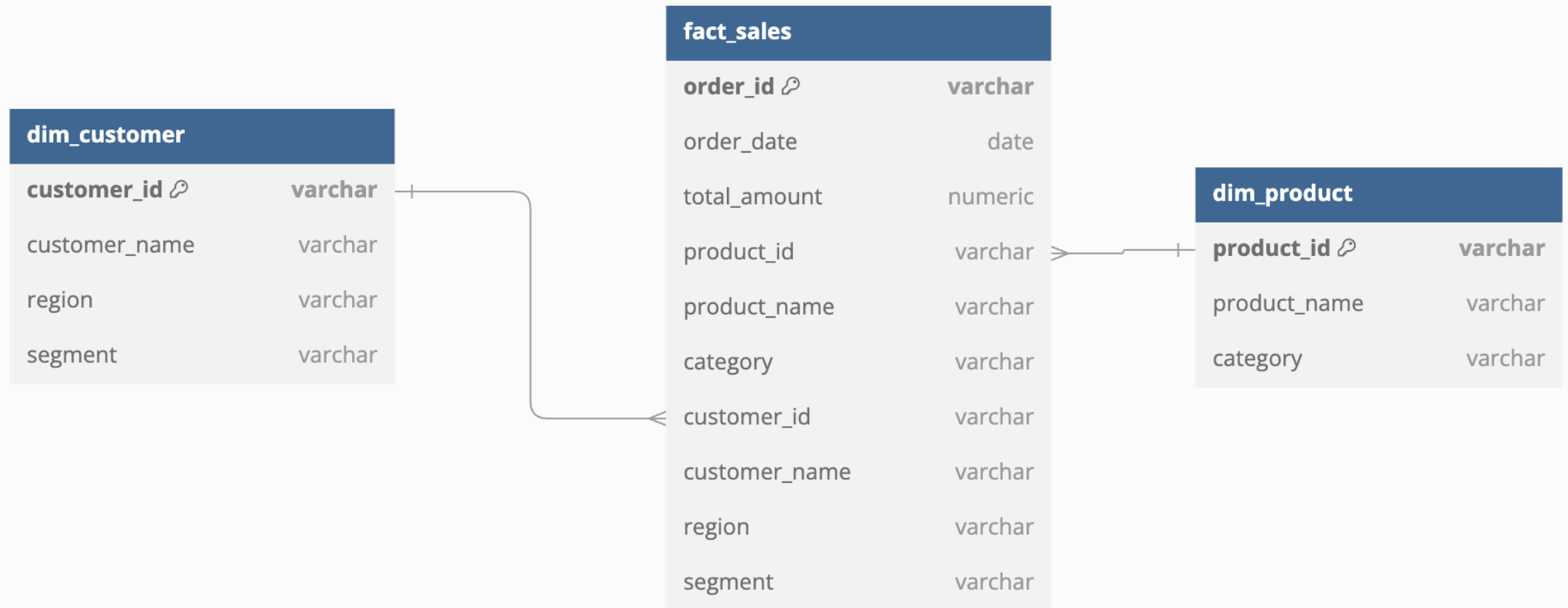
By: Deana Moghaddas



Data Pipeline Overview

1. Establish connection between RStudio and PostgreSQL database.
2. Load CSVs into database and store into preliminary data tables: **customers**, **products**, and **orders**.
3. Create intermediate datasets: **dim_customer** and **dim_product**
 - a) Use Create Table statements to define variables, types and constraints
 - Set maximum lengths dynamically using preliminary data tables; set primary keys to ensure uniqueness
 - b) Insert data into tables and take the following processing steps, using **customers** and **products** respectively:
 - Remove all whitespaces from ID variables, and remove trailing/leading spaces for remaining VARCHAR variables
 - Convert VARCHAR variables to uppercase for consistency. If VARCHAR variables are empty strings, set NULL.
 - Remove rows with primary keys containing NULL or NA since they cannot be used to join
 - Set all invalid dates to NA
4. Create intermediate dataset: **fact_sales**
 - a) Left join **dim_customer** and **dim_product** on foreign keys to **orders**. Repeat above processing steps for remaining variables.
 - b) Set numeric variables to 0 when NULL
5. Create final dataset: **sales_performance**
 - a) Using **fact_sales**, extract month from order_date variable. Aggregate revenue and group by month, customer and product.
 - b) Result: each row shows revenue generated from an individual customer on a certain product, within the month

Database Schema



Final Table and Use Cases

What does final table show?

- Each row shows revenue generated from an individual customer on a certain product, within the month

Use Cases within Tableau Dashboard (Graph, Chart, Table, etc.):

- Total revenue, revenue by product, revenue by month, revenue by region
- Top-performing products and customers, generating most revenue
- Top-performing products per segment
- Characteristics of customers most likely to make a purchase, and frequency of those purchases

customer_name <chr>	region <chr>	segment <chr>	product_name <chr>	category <chr>	month <dbl>	total_revenue <dbl>
ASHLEY MARTIN	NORTH	RETAIL	ONTO	CLOTHING	11	36734.40
PHILIP BOOTH	WEST	CORPORATE	PRESSURE	FURNITURE	11	23139.90
VICKI WILSON	EAST	CORPORATE	DETERMINE	CLOTHING	11	20832.30
LISA TRAN	SOUTH	SMALL BUSINESS	THROW	ACCESSORIES	11	14210.10
CYNTHIA LEE	NORTH	RETAIL	DETERMINE	CLOTHING	11	6489.00
ASHLEY MARTIN	NORTH	RETAIL	AGENCY	FURNITURE	10	39572.10
CYNTHIA LEE	NORTH	RETAIL	LETTER	ELECTRONICS	10	33572.70
DANIEL EDWARDS	WEST	SMALL BUSINESS	WHO	CLOTHING	10	27567.90
MICHAEL ERICKSON	WEST	RETAIL	LANGUAGE	ELECTRONICS	10	25135.20
RICHARD REED	WEST	RETAIL	DETERMINE	CLOTHING	10	19170.00

Example: 10 rows from **sales_performance**