

# Capstone project: Brazilian E-Commerce sales analysis

## Problem & target:

- Targeting the sales historical information
  - Focusing on the reporting sales historical performance on a monthly basis to overview the big picture of the selling amount
- Overview of sales segmentation based on each feature
  - Focusing on the reporting sales historical performance every month to overview the big picture of the selling amount.
- Planning for future marketing
  - Applying a report to let end users plan their future marketing plan, which targets to maximize sales amount during each month in a year and can focus the right products in each season in different states around brazil.

## Questions:

After defining the problem and target, the main focus is on the selling report we would like to answer the following questions:

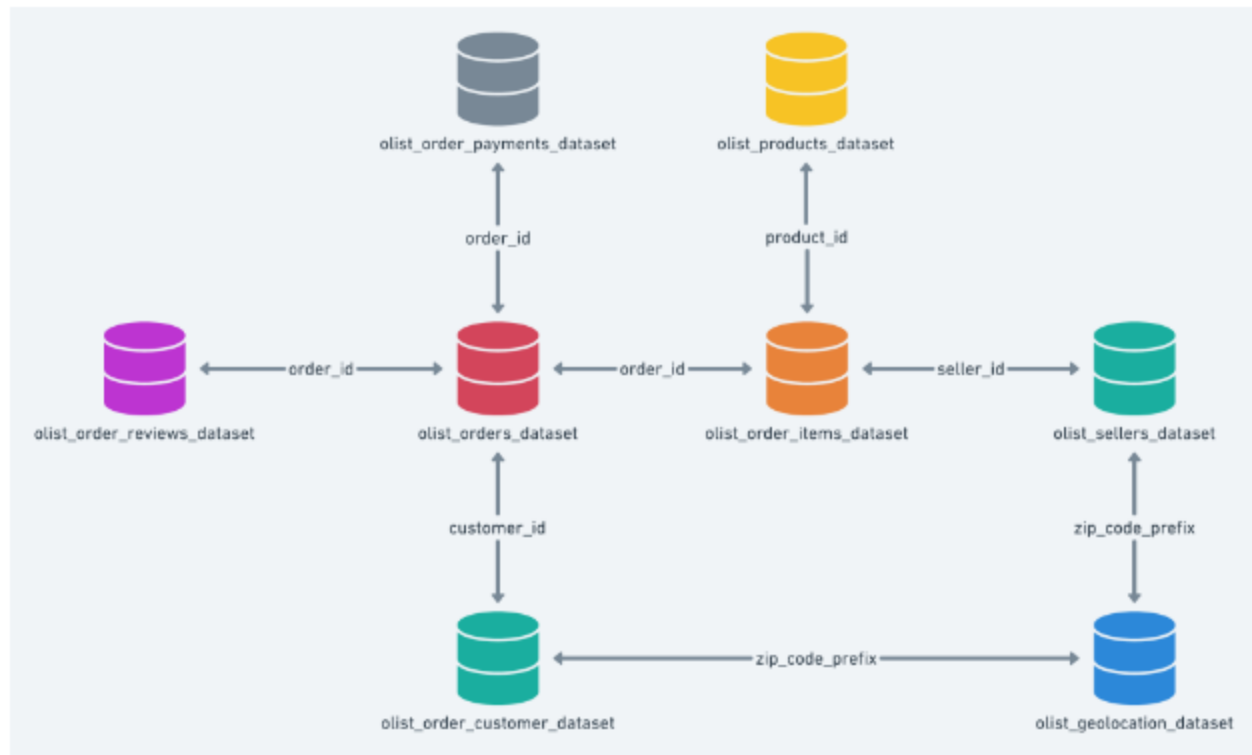
1. How is the total sales amount compare in term of product categories and each state
2. Average sales per order in each state
3. Each payment type sales proportion

## Data set:

Brazilian E-Commerce Public Dataset by Olist: [link](#)

Containing: 8 tables + 1 product English translation file (9 CSV files)

## Data Schema:



## Raw data overview:

### 1) Customer table

customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7fcb0	14409	franca	SP
18955e83d337fd6b2def6b18a428ac77	290c77bc529b7ac935b93aa66c333dc3	9790	sao bernardo do campo	SP
4e7b3e00288586ebd08712fdd0374a03	060e732b5b29e8181a18229c7b0b2b5e	1151	sao paulo	SP
b2b6027bc5c5109e529d4dc6358b12c3	259dac757896d24d7702b9acbbf3f3c	8775	mogi das cruzeiras	SP
4fd8ab171c80ec8364f7c12e35b23ad	345ecd01c38d18a9036ed96c73b8d066	13056	campinas	SP
879864dab9bc3047522c92c82e1212b8	4c93744516667ad3b8f11b645a3116a4	89254	jaragua do sul	SC
fd826e7cf63160e536e0908c76c3f441	addec96d2e059c80c30fe6871d30d177	4534	sao paulo	SP
5e274e7a0c3809e14aba7ad5aae0d407	57b2a98a409812fe9618067b6b8ebe4f	35182	timoteo	MG
5adf08c34b2e993982a47070956c5c65	1175e95fb47ddff9de6b2b06188f7e0d	81560	curitiba	PR
4b7139f34592b3a31687243a3021a75b	9afe194fb833f79e300e37e580171f22	30575	belo horizonte	MG
9fb35e1ed6da14a4977cd9aea4042bb	2a7715e1ed516b289ed9b29c7d0539a5	39400	montes claros	MG
5aa9e4fdd4fd20959cad2d772509598	2a46fb94aef5cbeeb850418118cee090	20231	rio de janeiro	RJ
b2d1536598b73a9abd18c0d75d92f0a3	918dc87cd72cd9f6cd4bd442cd785235	18682	lencois paulista	SP

## 2) Geolocation

geolocation_zip_code_prefix	geolocation_lat	geolocation_lng	geolocation_city	geolocation_state
1037	-23.54562128	-46.63929205	sao paulo	SP
1046	-23.54608113	-46.6448203	sao paulo	SP
1046	-23.54612897	-46.64295148	sao paulo	SP
1041	-23.54439216	-46.63949931	sao paulo	SP
1035	-23.54157796	-46.64160722	sao paulo	SP
1012	-23.5477623	-46.63536054	são paulo	SP
1047	-23.54627311	-46.64122517	sao paulo	SP
1013	-23.54692321	-46.6342637	sao paulo	SP
1029	-23.54376906	-46.63427784	sao paulo	SP
1011	-23.54763955	-46.63603162	sao paulo	SP
1013	-23.54732513	-46.63418379	sao paulo	SP
1032	-23.5384181	-46.63477838	sao paulo	SP
1014	-23.54643534	-46.63383023	sao paulo	SP
1012	-23.54894599	-46.63467113	sao paulo	SP
1037	-23.54518734	-46.63785524	são paulo	SP
1046	-23.54608113	-46.6448203	sao paulo	SP

## 3) Order-item

order_id	order_item_id	product_id	seller_id	shipping_limit_date	price	freight_value
00010242fe8c5a6d1ba2dd792cb16214	1	4244733e06e7ecb4970a6e2683c13e61	48436dade18ac8b2bce089ec2a041202	9/19/2017 9:45	58.9	13.29
00018f77f2f0320c557190d7a144bdd3	1	e5f2d52b802189ee58865ca93d83a8f	dd7ddc04e1b6c2c614352b383efe2d36	5/3/2017 11:05	239.9	19.93
000229ec398224ef6ca0657da4fc703e	1	c777355d18b72b67abbeef9df44fd0fd	5b51032eddd242adc84c38acab88f23d	1/18/2018 14:48	199	17.87
00024acbcdf0a6daa1e931b038114c75	1	7634da152a4610f1595efa32f14722fc	9d7a1d34a5052409006425275ba1c2b4	8/15/2018 10:10	12.99	12.79
00042b26cf59d7ce69dfabb4e55b4fd9	1	ac6c3623068f30de03045865e4e10089	df560393f3a51e74553ab94004ba5c87	2/13/2017 13:57	199.9	18.14
00048cc3ae777c65dbb7d2a0634bc1ea	1	ef92defde845ab8450f9d70c526ef70f	6426d21aca402a131fca5d0980a3c90	5/23/2017 3:55	21.9	12.69
00054e8431b9d7675808bcb819fb4a32	1	8d4f2bb7e93e6710a28f34fa83ee7d28	7040e82f899a04d1b434b795a43b4617	12/14/2017 12:10	19.9	11.85
000576fe39319847cbb9d288c5617fa6	1	557d850972a7d6f792fd18ae1400d9b6	5996cddab893a4652a15592fb58ab8db	7/10/2018 12:30	810	70.75
0005a1a1728c9d785b8e2b08b904576c	1	310ae3c140ff94b03219ad0adc3c778f	a416b6a846a11724393025641d4edd5e	3/26/2018 18:31	145.95	11.65
0005f50442cb953dc1d121e1fb923495	1	4535b0e1091c278df193e5a1d63b39f	ba143b05f0110f0dc71ad71b4466ce92	7/6/2018 14:10	53.99	11.4
00061f2a7bc09da83e415a52dc8a4af1	1	d63c1011f49d98b976c352955b1c4bea	cc419e0650a3c5ba77189a1882b7556a	3/29/2018 22:28	59.99	8.88
00063b381e2406b52ad429470734ebd5	1	f177554ea93259a5b282f24e33f65ab6	8602a61d680a10a82cceeda0d99ea3d	7/31/2018 17:30	45	12.98
0006ec9db01a64e59a68b2c340bf65a7	1	99a4788cb24856965c36a24e339b6058	4a3ca9315b744ce9f8e9374361493884	7/26/2018 17:24	74	23.32
0008288aa423d2a3f00fcb17cd7d8719	1	368c6c730842d78016ad823897a372db	1f50f920176fa81dab994f9023523100	2/21/2018 2:55	49.9	13.37
0008288aa423d2a3f00fcb17cd7d8719	2	368c6c730842d78016ad823897a372db	1f50f920176fa81dab994f9023523100	2/21/2018 2:55	49.9	13.37
0009792311464db532ff765bf7b182ae	1	8cab8abac59158715e0d70a36c807415	530ec6109d11eaf87999465c6afee01	8/17/2018 12:15	99.9	27.65
0009c9a17f916a706d71784483a5d643	1	3f27ac8e699df3d300ec4a5d8c5cf0b2	fc05ace8bcc92f75707dc0f01a27d269	5/2/2018 9:31	639	11.34
000aed2e25dbad2f9ddb70584c5a2ded	1	4fa33915031a8cde03dd0d3e8fb27f01	fe2032dab1a61af8794248c8196565c9	5/16/2018 20:57	144	8.77

## 4) Payment

order_id	payment_sequential	payment_type	payment_installments	payment_value
b81ef226f3fe1789b1e8b2acac839d17	1	credit_card	8	99.33
a9810da82917af2d9aefd1278f1dcfa0	1	credit_card	1	24.39
25e8ea4e93396b6fa0d3dd708e76c1bd	1	credit_card	1	65.71
ba78997921bbcd1373bb41e913ab953	1	credit_card	8	107.78
42fdf880ba16b47b59251dd489d4441a	1	credit_card	2	128.45
298fcdf1f73eb413e4d26d01b25bc1cd	1	credit_card	2	96.12
771ee386b001f06208a7419e4fc1bbd7	1	credit_card	1	81.16
3d7239c394a212faae122962df514ac7	1	credit_card	3	51.84
1f78449c87a54faf9e96e88ba1491fa9	1	credit_card	6	341.09
0573b5e23cbd798006520e1d5b4c6714	1	boleto	1	51.95
d88e0d5fa41661ce03cf6cf336527646	1	credit_card	8	188.73
2480f727e869fdeb397244a21b721b67	1	credit_card	1	141.9
616105c9352a9668c38303ad44e056cd	1	credit_card	1	75.78
cf95215a722f3ebf29e6bbab87a29e61	1	credit_card	5	102.66
769214176682788a92801d8907fa1b40	1	credit_card	4	105.28
12e5cfe0e4716b59afb0e0f4a3bd6570	1	credit_card	10	157.45
61059985a6fc0ad64e95d9944caacdad	1	credit_card	1	132.04
79da3f5fe31ad1e454f06f95dc032ad5	1	credit_card	1	98.94
8ac09207f415d55acff302df7d6a895c	1	credit_card	4	244.15

## 5) Review

review_id	order_id	review_score	review_comment_title	review_comment_message	review_creation_date	review_answer_timestamp
7bc2406110b926393aa56f80a40eba40	73fc7af87114b39712e6da79b0e377eb	4			1/18/2018 0:00	1/18/2018 21:46
80e641a11e56f04c1a6409d5645fd0e	a548910a1c6147796098f8f73dbba33	5			3/10/2018 0:00	3/11/2018 3:05
228ce5500dc1d8e020d8d1322874b0f0	f9e4b658b201a9f2ecdccbb34bed034b	5			2/17/2018 0:00	2/18/2018 14:36
e94fb393e7b328340b789f88b30750e	658677c97b385a9be170737859d3511b	5		Recebi bem antes do prazo est	4/21/2017 0:00	4/21/2017 22:02
f7c4243c7fe1938f181bec41a392bdeb	8eebf881e283fa7e4f11123a3fb894f1	5		Parabéns lojas lanister ador	3/1/2018 0:00	3/2/2018 10:26
15197a66ff480650b5434f1b46cda19	b18dcdf73be6366873cd26c572d1dc	1			4/13/2018 0:00	4/16/2018 0:39
07f9bee5d1850860dafd761afa7ff16	a48aa0d2dce3a2e87348811bctdf22b	5			7/16/2017 0:00	7/18/2017 19:30
7c6400515c67679fbce952a7525281ef	c31a859e34e3adac22f376954e19b39d	5			8/14/2018 0:00	8/14/2018 21:36
a3f6f7f6f433de0aefbb97da197c554c	9c214ac970e84273583ab523dafd09b	5			5/17/2017 0:00	5/18/2017 12:05
8670d52e15e0043ae7de4c01cc2fe06	b9bf720beb4ab3728760088589c62129	4	recomendo	aparelho eficiente, no site a m	5/22/2018 0:00	5/23/2018 16:45
c9cfd2d5ab5911836ababae136c3a10c	cdf9ae68e72324eeb25c7de974696ee2	5			12/23/2017 0:00	12/26/2017 14:36
99052551d87e5f62e6c9f6974ec392e9	3d374c9e46530b50ed4a7648915306a9	5			12/19/2017 0:00	12/20/2017 10:25
4b49719c8a200003f700d3d980ea1a19	9d6f15f95d01e79bd1349cc208361f09	4		Mas um pouco	2/16/2018 0:00	2/20/2018 10:52
23f75a37effc35d9a913b4e1a9483793	2eaf8e099d871cd5c22b831b5ea8f9e0e	4			3/28/2018 0:00	3/30/2018 15:10
9a0abb668baf95a6d2b05db43284c4	d7bd0e4efd94846eb73642b4e3e75c3	3			4/30/2017 0:00	5/3/2017 0:02
3948b09f7c818e2d86c9a546758b2335	e51478e7e277a83743b6f9991dbfa3fb	5	Super recomendo	Vendedor confiável, produto i	5/23/2018 0:00	5/24/2018 3:00
93148f9799f5bfa510cc7bcd488c01	0dacf04c5ad59fd5a0cc1faa07c34e39	2		GOSTARIA DE SABER O QUE H	1/18/2018 0:00	1/20/2018 21:25
8a15a274d95600fa14f8be64e37a0e67	ff1581a08b3011021e7c7de592ddc81e	5			3/24/2018 0:00	3/26/2018 15:58
fdbdb2629a7cde0f66657acc92084e7f	70a752414a13d09cc1f2b4376914b28e	3			9/29/2017 0:00	10/2/2017 1:12
373cbeecae8286a2b66c97b1b157ec46	583174f6e37d3d5f0d6661b63aad1786	1	NÃO chegou meu produ	PÁDssimo	8/15/2018 0:00	8/15/2018 4:10

## 6) Order

order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date
e481f51cdbc54678b7cc49136f2d6af7	9ef432be6251297304e76186b10a928d	delivered	10/2/2017 10:56	10/2/2017 11:07	10/4/2017 19:55	10/10/2017 21:25	10/18/2017 0:00
53cddb2fc8bc7dce0b6741e2150273451	b0830fb7474a6c6d20dea0b8c802d7ef	delivered	7/24/2018 20:41	7/26/2018 3:24	7/26/2018 14:31	8/7/2018 15:27	8/13/2018 0:00
47770eb9100c2d0c4946d9cf07ec65d	41ce2a54c0b03bf3443cd931a367089	delivered	8/8/2018 8:38	8/8/2018 8:55	8/8/2018 13:50	8/17/2018 18:06	9/4/2018 0:00
949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbe7375364d82	delivered	11/18/2017 19:28	11/18/2017 19:45	11/22/2017 13:39	12/2/2017 0:28	12/15/2017 0:00
ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbcb4fb7aad2c	delivered	2/13/2018 21:18	2/13/2018 22:20	2/14/2018 19:46	2/16/2018 18:17	2/26/2018 0:00
a4591c265e18cb1dcee52889e2d8acc3	503740e9ca751ccdda7ba28e9ab8f608	delivered	7/9/2017 21:57	7/9/2017 22:10	7/11/2017 14:58	7/26/2017 10:57	8/1/2017 0:00
136cce7faa42fdb2cef53fcd79a6098	ed0271e0b7da060a393796590e7b737a	invoiced	4/11/2017 12:22	4/13/2017 13:25			5/9/2017 0:00
6514b8ad8028c9f2c2374ded245783f	9bdf08b4b3b52b5526ff42d37d47f222	delivered	5/16/2017 13:10	5/16/2017 13:22	5/22/2017 10:07	5/26/2017 12:55	6/7/2017 0:00
76c6e866289321a7c93b82b54852d333	f54a9f0e6b351c431402b8461ea51999	delivered	1/23/2017 18:29	1/25/2017 2:50	1/26/2017 14:16	2/2/2017 14:08	3/6/2017 0:00
e69bf5beb88e0ed6a785585b27e16dbf	31ad1d1b63eb9962463f764d4e6e0c9d	delivered	7/29/2017 11:55	7/29/2017 12:05	8/10/2017 19:45	8/16/2017 17:14	8/23/2017 0:00
e6ce16cb79ecd1d90b1da9085a6118aeb	494ded5b201313c64ed7f100595b59c	delivered	5/16/2017 19:41	5/16/2017 19:50	5/18/2017 11:40	5/29/2017 11:18	6/7/2017 0:00
34513ce0c4fab462a55830c0989c7edb	7711cf624183d4843aafe81855097bc37	delivered	7/13/2017 19:58	7/13/2017 20:10	7/14/2017 18:43	7/19/2017 14:04	8/8/2017 0:00
82566a660a982b15fb86e904cd832918	d3e3b74c766bc6214e0c830b17ee2341	delivered	6/7/2018 10:06	6/9/2018 3:13	6/11/2018 13:29	6/19/2018 12:05	7/18/2018 0:00
5f96c15d0b717ac6ad1fd77225a350	19402a48fe860416adf93348aba37740	delivered	7/25/2018 17:44	7/25/2018 17:55	7/26/2018 13:16	7/30/2018 15:52	8/8/2018 0:00
432aaaf21d851677c2b80e948c4e42cc	3df704f53d3f1d4818840b34ec672a9f	delivered	3/1/2018 14:14	3/1/2018 15:10	3/2/2018 21:09	3/12/2018 23:36	3/21/2018 0:00
dc3b6b511fca05097c45c05de84dc3	3c6828a50ffe546942b7a473d70ac0fc	delivered	6/7/2018 19:03	6/12/2018 23:31	6/11/2018 14:54	6/21/2018 15:34	7/4/2018 0:00
403b978360bc04a622354cf531062e5f	738b086814c6fcc74b8cc583f8516ee3	delivered	1/2/2018 19:00	1/2/2018 19:09	1/3/2018 18:19	1/20/2018 1:38	2/6/2018 0:00
116f0b09343b49556bba5d735bee0cdf	3187789ec990987628d7a9beb4dd6ac	delivered	12/26/2017 23:41	12/26/2017 23:50	12/28/2017 18:33	1/8/2018 22:36	1/29/2018 0:00
85ce859fd6dc634de8d2f1e290444043	059f7fc5719c7da6cbafe370971a8d70	delivered	11/21/2017 0:03	11/21/2017 0:14	11/23/2017 21:32	11/27/2017 18:28	12/11/2017 0:00

## 7) Product

product_id	product_category_name	product_name_lenght	product_description_lenght	product_photos_qty	product_weight_g	product_length_cm	product_height_cm	product_width_cm
1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	40	287	1	225	16	10	14
3aa071139cb16b67ca9e5dea641aa2 artes		44	276	1	1000	30	18	20
96bd76ec8810374ed1b65e291975717	esporte_lazer	46	250	1	154	18	9	15
cef67b7cfe19066a932b7673e239be23c	bebes	27	261	1	371	26	4	26
9dc1a7de27444849c219cf195d0b71	utilidades_domesticas	37	402	4	625	20	17	13
41d3672d4792049fa1779bb35283ed1	instrumentos_musicais	60	745	1	200	38	5	11
732bd381ad09e530fe0a5f457d81becc	cool_stuff	56	1272	4	18350	70	24	44
2548af3e6e77a690c3eb6368e9ab61e	moveis_decoracao	56	184	2	900	40	8	40
37cc742be07708b53a98702e77a21a0	eletrodomesticos	57	163	1	400	27	13	17
8c92109888e8cdf9d66dc7e46302557	brinquedos	36	1156	1	600	17	10	12
14aa47b7fe5c25522b47b4b29c98dbcf	cama_mesa_banho	54	630	1	1100	16	10	16
03b63c5fc16691530586ae020c34551	bebes	49	728	4	7150	50	19	45
cf55509ea8edaac1d28fdb16e48fc22	instrumentos_musicais	43	1827	3	250	17	7	17
7bb6f29c2be5716194f96496660c7c2	moveis_decoracao	51	2083	2	600	68	11	13
eb31436580a610f202c859463d8c741	construcao_ferramentas	59	1602	4	200	17	7	17
3bb7f144022e6732727d8d838a7b13b	esporte_lazer	22	3021	1	800	16	2	11
6a2fb4dd53d2cddb88e0432f1284a004	perfumaria	39	346	2	400	27	5	20
a1b71017a84f92fd8da4aeefba108a24	informatica_acessorios	59	636	1	900	40	15	20
a0736b92e52f6cead290e30b578413b	moveis_decoracao	56	296	2	1700	100	7	15

## 8) Seller

seller_id	seller_zip_code_prefix	seller_city	seller_state
3442f8959a84dea7ee197c632cb2df15	13023	campinas	SP
d1b65fc7debc3361ea86b5f14c68d2e2	13844	mogi guacu	SP
ce3ad9de960102d0677a81f5d0bb7b2d	20031	rio de janeiro	RJ
c0f3eea2e14555b6faeea3dd58c1b1c3	4195	sao paulo	SP
51a04a8a6bdbcb23deccc82b0b80742cf	12914	braganca paulista	SP
c240c4061717ac1806ae6ee72be3533b	20920	rio de janeiro	RJ
e49c26c3edfa46d227d5121a6b6e4d37	55325	brejao	PE
1b938a7ec6ac5061a66a3766e0e75f90	16304	penapolis	SP
768a86e36ad6aae3d03ee3c6433d61df	1529	sao paulo	SP
ccc4bbb5f32a6ab2b7066a4130f114e3	80310	curitiba	PR
8cb7c5ddf41f4d506eba76e9a4702a25	75110	anapolis	GO
a7a9b880c49781da66651ccf4ba9ac38	13530	itirapina	SP
8bd0f31cf0a614c658f6763bd02dea69	1222	sao paulo	SP
05a48cc8859962767935ab9087417fbb	5372	sao paulo	SP
7b8e8ec35bad4b0ef7e3963650b0a87b	88705	tubarao	SC
1444c08e64d55fb3c25f0f09c07ffc2	42738	lauro de freitas	BA
166e8f1381e09651983c38b1f6f91c11	88780	imituba	SC
e38db885400cd35c71dfdd162f2c1dbcf	70740	brasilia	DF
d2e753bb80b7d4faa77483ed00edc8ca	45810	porto seguro	BA
f9ec7093df3a7b346b7bcf7864069ca3	5138	sao paulo	SP



## 9) Product category translation

product_category_name	product_category_name_english
beleza_saude	health_beauty
informatica_acessorios	computers_accessories
automotivo	auto
cama_mesa_banho	bed_bath_table
moveis_decoracao	furniture_decor
esporte_lazer	sports_leisure
perfumaria	perfumery
utilidades_domesticas	housewares
telefonica	telephony
relogios_presentes	watches_gifts
alimentos_bebidas	food_drink
bebes	baby
papelaria	stationery
tablets_impressao_imagem	tablets_printing_image
brinquedos	toys
telefonica_fixa	fixed_telephony
ferramentas_jardim	garden_tools
fashion_bolsas_e_acessorios	fashion_bags_accessories
eletroportateis	small_appliances
consoles_games	consoles_games
audio	audio
fashion_calcados	fashion_shoes
cool_stuff	cool_stuff
malas_acessorios	luggage_accessories

\* Raw data will be uploaded to AWS S3 on a monthly basis

**Amazon S3** ×

Amazon S3 > Buckets > brazilian-bucket-final

**brazilian-bucket-final** info

Objects Properties Permissions Metrics Management Access Points

**Objects (10)**

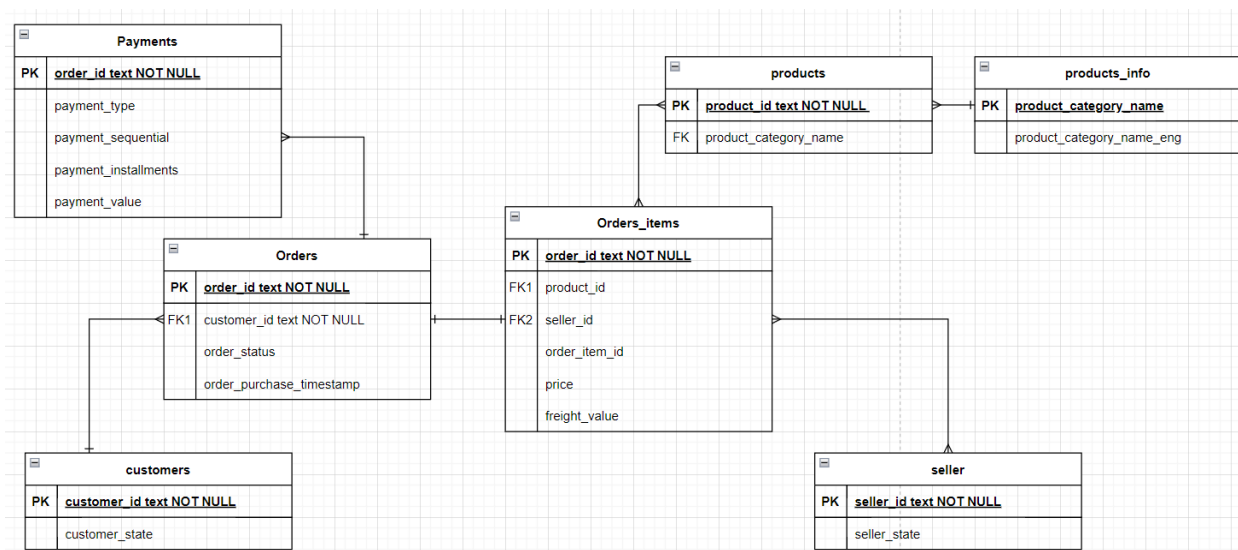
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
final_table/	Folder	-	-	-
olist_geolocation_dataset.csv	csv	December 17, 2022, 11:36:34 (UTC+07:00)	58.4 MB	Standard
olist_orders_dataset.csv	csv	December 17, 2022, 11:36:34 (UTC+07:00)	16.8 MB	Standard
olist_order_items_dataset.csv	csv	December 17, 2022, 11:36:41 (UTC+07:00)	14.7 MB	Standard
olist_order_payments_dataset.csv	csv	December 17, 2022, 11:36:55 (UTC+07:00)	5.5 MB	Standard
olist_order_reviews_dataset.csv	csv	December 17, 2022, 11:37:06 (UTC+07:00)	13.7 MB	Standard
olist_products_dataset.csv	csv	December 17, 2022, 11:37:14 (UTC+07:00)	2.5 MB	Standard
olist_sellers_dataset.csv	csv	December 17, 2022, 11:37:15 (UTC+07:00)	170.6 KB	Standard
product_category_name_translation.csv	csv	December 17, 2022, 11:37:16 (UTC+07:00)	2.6 KB	Standard
olist_customers_dataset.csv	csv	December 17, 2022, 11:37:27 (UTC+07:00)	8.6 MB	Standard

## Data model:



## ETL process (using Pyspark) in Data lake:

- Some of the tables are not used due to unnecessary information and not relate to sales which are:
  - Geolocation
  - Review
- Transform data by using the following step to clean in each table before merging into 1 table in the final step
  - Table customer

```
df_customers = spark.read.csv(file1, header=True)
```

```
df_customers.show(5)
```

```

+-----+-----+-----+-----+-----+
| customer_id | customer_unique_id | customer_zip_code_prefix | customer_city | customer_state |
+-----+-----+-----+-----+-----+
| 06b8999e2fba1a1fb... | 861eff4711a542e4b... | 14409 | franca | SP |
| 18955e83d337fd6b2... | 290c77bc529b7ac93... | 09790 | sao bernardo do c... | SP |
| 4e7b3e00288586ebd... | 060e732b5b29e8181... | 01151 | sao paulo | SP |
| b2b6027bc5c5109e5... | 259dac757896d24d7... | 08775 | mogi das cruze... | SP |
| 4f2d8ab171c80ec83... | 345ecd01c38d18a90... | 13056 | campinas | SP |
+-----+-----+-----+-----+-----+
only showing top 5 rows
  
```

## ○ Table order\_item

```
df_order_item= spark.read.csv(file2, header=True)
```

```
df_order_item=df_order_item.withColumn("price", col("price")*1.0)
df_order_item=df_order_item.withColumn("freight_value", col("freight_value")*1.0)
```

```
df_order_item.show(3)
```

order_id	order_item_id	product_id	seller_id	shipping_limit_date	price	freight_value
00010242fe8c5a6d1...	1	4244733e06e7ecb49...	48436dade18ac8b2b...	2017-09-19 09:45:35	58.9	13.29
00018f77f2f0320c5...	1	e5f2d52b802189ee6...	dd7ddc04e1b6c2c61...	2017-05-03 11:05:13	239.9	19.93
000229ec398224ef6...	1	c777355d18b72b67a...	5b51032eddd242adc...	2018-01-18 14:48:30	199.0	17.87

only showing top 3 rows

```
df_order_item.printSchema()
```

```
root
|-- order_id: string (nullable = true)
|-- order_item_id: string (nullable = true)
|-- product_id: string (nullable = true)
|-- seller_id: string (nullable = true)
|-- shipping_limit_date: string (nullable = true)
|-- price: double (nullable = true)
|-- freight_value: double (nullable = true)
```

\* Noted: The reason for calculation column by multiplying by 1.0 is to convert the string to float type (After trying, float keyword or even FloatType() with withColumn() method is not working which we can not aggregate price or freight\_value)

```
df_order_item=df_order_item.groupBy('order_id','product_id','seller_id') \
    .agg(sum("freight_value").alias("freight_value"), \
        sum("price").alias("price"), \
        )
```

```
df_order_item.show(5)
```

order_id	product_id	seller_id	freight_value	price
005e5166e99d1e4d0...	03bb24d19ea7449ce...	e9bc59e7b60fc3063...	16.6	120.1
0070092bb6004faaf...	31a2f42a87890f87d...	4c03b9dd4c11ee2cb...	14.06	149.0
014e78d1239f19fdc...	394b16cd32e9f8833...	8b321bb669392f516...	14.1	14.4
01786fff06f2f7e8a...	382132cf96c88a277...	b2ba3715d723d2451...	9.34	59.9
01c4766d43d50d892...	639cacc61d50ee820...	3361277dc30b7cccd...	18.23	35.89

only showing top 5 rows



\* Group by is used to aggregate all numerical in each order in a different product and seller ID

### ○ Table payments

```
df_payments = spark.read.csv(file3, header=True)
```

```
df_payments.show(5)
```

```
+-----+-----+-----+-----+-----+
|          order_id|payment_sequential|payment_type|payment_installments|payment_value|
+-----+-----+-----+-----+-----+
|b81ef226f3fe1789b...|          1|credit_card|          8|          99.33|
|a9810da82917af2d9...|          1|credit_card|          1|          24.39|
|25e8ea4e93396b6fa...|          1|credit_card|          1|          65.71|
|ba78997921bbcdc13...|          1|credit_card|          8|         107.78|
|42fdf880ba16b47b5...|          1|credit_card|          2|         128.45|
+-----+-----+-----+-----+-----+
```

only showing top 5 rows

```
df_payments = df_payments.withColumn("payment_value", col("payment_value")*1.0)
```

```
df_payments.printSchema()
```

```
root
 |-- order_id: string (nullable = true)
 |-- payment_sequential: string (nullable = true)
 |-- payment_type: string (nullable = true)
 |-- payment_installments: string (nullable = true)
 |-- payment_value: double (nullable = true)
```

```
df_payments = df_payments.groupBy('order_id', 'payment_type') \
    .agg(sum("payment_value").alias("payment_value"), \
    )
```

```
df_payments.show(5)
```

```
+-----+-----+-----+
|          order_id|payment_type|payment_value|
+-----+-----+-----+
|298fcdf1f73eb413e...|credit_card|          96.12|
|d9b53f70b57a028c3...|credit_card|          24.29|
|cf014dc8804713618...|   voucher|107.49000000000001|
|c85ea30e9a24abecb...|credit_card|          72.98|
|873d039c319333bd4...|credit_card|          98.25|
+-----+-----+-----+
```

only showing top 5 rows

## ○ Table orders

```
df_orders = spark.read.csv(file4, header=True)
```

```
df_orders.show(5)
```

```
+-----+-----+-----+-----+-----+-----+-----+
|      order_id|      customer_id|order_status|order_purchase_timestamp|  order_approved_at|order_delivered_carrier_date|order_delivered_customer_date|order_estimated_delivery_date|
+-----+-----+-----+-----+-----+-----+-----+
|e481f51cbdc54678b...|9ef432eb625129730...| delivered| 2017-10-02 10:56:33|2017-10-02 11:07:15|      2017-10-04 19:55:00|
|2017-10-10 21:25:13|      2017-10-18 00:00:00|
|53cdb2fc8bc7dce0b...|b0830fb4747a6c6d2...| delivered| 2018-07-24 20:41:37|2018-07-26 03:24:27|      2018-07-26 14:31:00|
|2018-08-07 15:27:45|      2018-08-13 00:00:00|
|47770eb9100c2d0c4...|41ce2a54c0b03bf34...| delivered| 2018-08-08 08:38:49|2018-08-08 08:55:23|      2018-08-08 13:50:00|
|2018-08-17 18:06:29|      2018-09-04 00:00:00|
|949d5b44dbf5de918...|f88197465ea7920ad...| delivered| 2017-11-18 19:28:06|2017-11-18 19:45:59|      2017-11-22 13:39:59|
|2017-12-02 00:28:42|      2017-12-15 00:00:00|
|ad21c59c0840e6cb8...|8ab97904e6daea886...| delivered| 2018-02-13 21:18:39|2018-02-13 22:20:29|      2018-02-14 19:46:34|
|2018-02-16 18:17:02|      2018-02-26 00:00:00|
+-----+-----+-----+-----+-----+-----+-----+
only showing top 5 rows
```

## ○ Table products

```
df_products = spark.read.csv(file5, header=True)
```

```
df_products.show(5)
```

```
+-----+-----+-----+-----+-----+-----+-----+
|      product_id|product_category_name|product_name_lenght|product_description_lenght|product_photos_qty|product_weight_g|product_length_cm|product_height_cm|product_width_cm|
+-----+-----+-----+-----+-----+-----+-----+
|1e9e8ef84dbcff454...|      perfumaria|      40|      287|      1|      225|      16|      10|
|14|
|3aa071139cb16b67c...|      artes|      44|      276|      1|      1000|      30|      18|
|20|
|96bd76ec8810374ed...|      esporte_lazer|      46|      250|      1|      154|      18|      9|
|15|
|cef67bcfe19066a93...|      bebes|      27|      261|      1|      371|      26|      4|
|26|
|9dc1a7de274444849...|      utilidades_domest...|      37|      402|      4|      625|      20|      17|
|13|
+-----+-----+-----+-----+-----+-----+-----+
only showing top 5 rows
```

```
df_products.printSchema()
```

```
root
|-- product_id: string (nullable = true)
|-- product_category_name: string (nullable = true)
|-- product_name_lenght: string (nullable = true)
|-- product_description_lenght: string (nullable = true)
|-- product_photos_qty: string (nullable = true)
|-- product_weight_g: string (nullable = true)
|-- product_length_cm: string (nullable = true)
|-- product_height_cm: string (nullable = true)
|-- product_width_cm: string (nullable = true)
```

## ○ Table sellers

```
df_sellers= spark.read.csv(file6, header=True)
```

```
df_sellers.show(5)
```

```
+-----+-----+-----+-----+
| seller_id|seller_zip_code_prefix| seller_city|seller_state|
+-----+-----+-----+-----+
|3442f8959a84dea7e...|13023|campinas|SP|
|d1b65fc7debc3361e...|13844|mogi guacu|SP|
|ce3ad9de960102d06...|20031|rio de janeiro|RJ|
|c0f3eea2e14555b6f...|04195|sao paulo|SP|
|51a04a8a6bdc23de...|12914|braganca paulista|SP|
+-----+-----+-----+-----+
only showing top 5 rows
```

```
df_sellers.printSchema()
```

```
root
|-- seller_id: string (nullable = true)
|-- seller_zip_code_prefix: string (nullable = true)
|-- seller_city: string (nullable = true)
|-- seller_state: string (nullable = true)
```

## ○ Table translate

```
df_translate= spark.read.csv(file7, header=True)
```

```
df_translate.show(5)
```

```
+-----+-----+
|product_category_name|product_category_name_english|
+-----+-----+
|beleza_saude|health_beauty|
|informatica_acess...|computers_accesso...|
|automotivo|auto|
|cama_mesa_banho|bed_bath_table|
|moveis_decoracao|furniture_decor|
+-----+-----+
only showing top 5 rows
```

```
df_translate.printSchema()
```

```
root
|-- product_category_name: string (nullable = true)
|-- product_category_name_english: string (nullable = true)
```

- Using “createOrReplaceTempView( )” in each table and then merging all table into a final table by selecting only some features that relate to our purpose in the data model which is listed in below table

```

final_table = spark.sql("""
select
    o.order_id
    , o.customer_id
    , i.seller_id
    , i.product_id
    , o.order_status
    , month(o.order_date) as month
    , year(o.order_date) as year
    , p.payment_type
    , c.customer_state
    , s.seller_state
    , t.product_category_name_english
    , p.payment_value
    , i.price
    , i.freight_value
from table_orders o
inner join table_order_item i
    on o.order_id = i.order_id
inner join table_customers c
    on o.customer_id = c.customer_id
inner join table_payments p
    on o.order_id = p.order_id
inner join table_sellers s
    on i.seller_id = s.seller_id
inner join table_products pr
    on i.product_id = pr.product_id
left join table_translate_cat t
    on pr.product_category_name = t.product_category_name

""")

```

```
final_table.show(3)
```

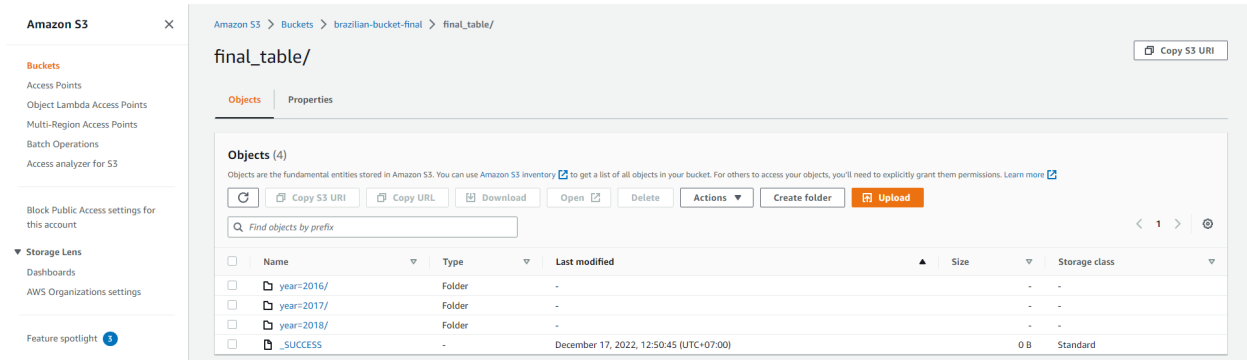
order_id	customer_id	seller_id	product_id	order_status	month	year	payment_type	customer_state	seller_state	product_category_english	payment_value	price	freight_value
[005e5166e99d1e4d0...]	[0f2a7baf176f693c2...]	[e9bc59e7b60fc3063...]	[03bb24d19ea7449ce...]	delivered	2	2018	boleto	SC	PR	computers_accessories	136.7	120.1	16.6
[0070092bb6004faaf...]	[399d1e628c48b7c3b...]	[4c03b9dd4c11ee2cb...]	[31a2f42a87890f87d...]	delivered	7	2017	credit_card	SP	SP	toys	163.06	149.0	14.06
[014e78d1239f19fdc...]	[bd5dee4ca2945a0ec...]	[8b321bb669392f516...]	[394b16cd32e9f8833...]	delivered	1	2018	credit_card	RS	SP	electronics	28.5	14.4	14.1

only showing top 3 rows

- Upload the final table back to AWS S3 by using this command

```
final_table.write.partitionBy("year").mode("overwrite").option("header",True).csv("s3a://brazilian-bucket-final/final_table")
```

**\*\* Final table is stored in the data lake in CSV format which is partitioning by “year” like blow figure:**

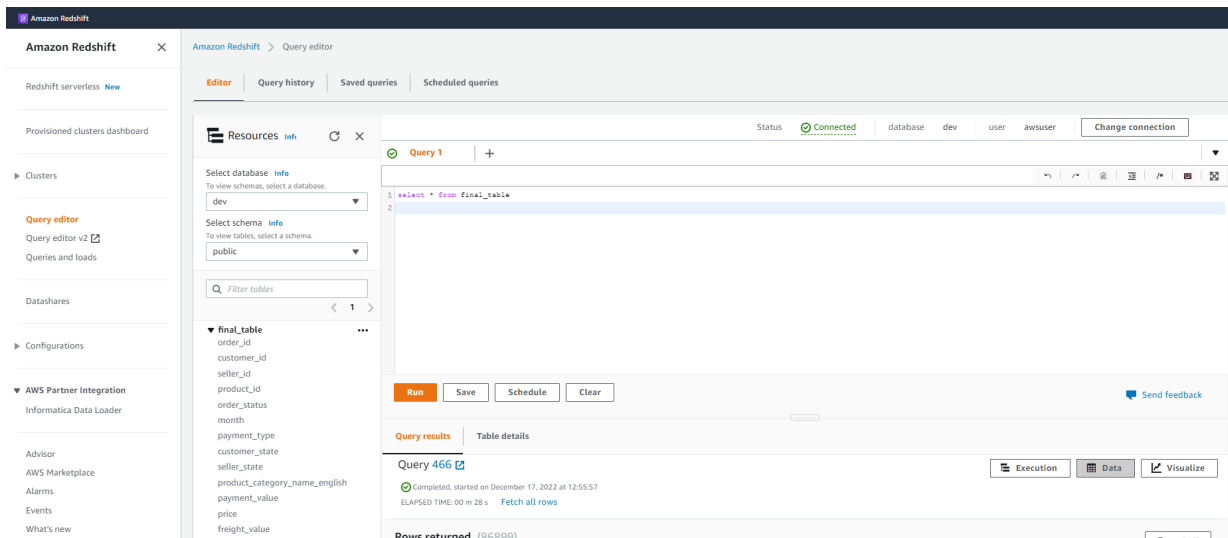


Amazon S3 console view of the `final_table/` bucket. The bucket contains four objects:

Name	Type	Last modified	Size	Storage class
<code>year=2016/</code>	Folder	-	-	-
<code>year=2017/</code>	Folder	-	-	-
<code>year=2018/</code>	Folder	-	-	-
<code>_SUCCESS</code>	-	December 17, 2022, 12:50:45 (UTC+07:00)	0 B	Standard

order_id	customer_id	seller_id	product_id	order_status	month	payment_type	customer_state	seller_state	product_category	payment_value	price	freight_value
1ff217aa6b3a9b720c4b1eaadf733430c5c1				delivered	10	credit_card	SP	SP	consoles_games	44.23	32.49	11.74
35b8e54d71aaa5eaa5897060da82c41d13dc				invoiced	10	boleto	MG	SP	computers	218.87	38	16.72
24523db67a60703155f0057b67636f5be2c				delivered	10	credit_card	RJ	SP	audio	147.08	132	15.08
063b573b7285195a51ecccfa2bb3d37da85f				shipped	10	credit_card	SP	PR	health_beauty	87.53	69.9	17.63
35d8c6d5e38fa25096dd2bdf855f436a5130				delivered	10	credit_card	SP	MG	furniture	192.56	169.9	22.66
02a0eb7c743d9a8df883e197e95f8c74f4a				invoiced	10	credit_card	PE	RJ	books_games	144.54	119.5	25.04
14b01335f10d74e1a0336182e7ace91f88d				delivered	10	credit_card	SP	SP	furniture	162.39	14.9	4.26
2e7a8482f08c5351af1554a6855c1488892f				shipped	9	credit_card	RR	MG	furniture	136.23	39.99	31.67
021d08e47470a1dd1ffff564a4fa819714dc				delivered	10	credit_card	MG	SP	auto	66.03	51.5	14.53
5c973d2b78a863458c620c87c17e9eebb8e				delivered	10	credit_card	RJ	RJ	perfumery	363.51	349.9	13.61
71303d7eb106b360f25e6ffe97d2998d7ce				canceled	10	credit_card	SP	SP	baby	109.34	100	9.34
4551b4884dfa7df4afecccfa2bb1e571e87				canceled	10	credit_card	RJ	PR	health_beauty	104.6	87.99	16.61
5cc475c7c75e8f990b897060da83410cbd7c				delivered	10	boleto	SP	SP	computers	82.69	71	11.69
6b3ee769721a6abdf0ce27a3cc34c7d4a2ef				delivered	10	credit_card	SP	SP	health_beauty	65.77	57	8.77
6b5e619dc9ff8efe6cd101c6da59805b37a2				delivered	10	credit_card	SP	SP	industry_c	207.22	179.8	27.42

## Data warehouse:



Amazon Redshift console view of the `final_table` in the `dev` database. The table contains 466 rows. The query editor shows the SQL statement: `select * from final_table`.

Query results: 466 rows returned. Execution time: 00 m 28 s. Fetch all rows.

Query results | Table details

Query 466 [🔗](#) Execution Data Visualize

🟢 Completed, started on December 17, 2022 at 12:55:57  
ELAPSED TIME: 00 m 28 s [Fetch all rows](#)

Rows returned (86899) Export ▼

< 1 2 3 4 5 6 7 ... 8690 > ⚙️

order_id	customer_id	seller_id	product_id	order_status	
02358a9949662b292e80635305dea78b	e69e27216b375e2c7e1be6298b051bf2	99eaacc9e6046db1c82b163c5f84869f	0cf573090c66bb30ac5e53c82bdb0403	delivered	5
04f2e46bdb0ab9fb67a4717100fb3ca71	ce8e936f636f55678640360ddaa0adea	6c7d50c24b3ccd2fd83b44d8bb34e073	55b04296824029ae20bbeb3b200e69d0	delivered	1
056bfadd41b8600ad5ecfef2ac132188	91faeb032f47277f654a8202eaa9fed6	daeb5653dd96c1b11860f72209795012	2bb67f2cd3ff60400e0bad33aee72d2a	delivered	1
05e18a6cef41f63dc86b7b254954dffc	6aa98365fd711a9e996d0d41a80c3be0	4a3ca9315b744ce9f8e9374361493884	8963eba2629ee030c4de83703c0ae020	delivered	1
0641d69ef04d0d7b004cb9ec0106c3ea	f7f3861057dff6df914b58282f051b01	37be5a7c751166fbc5f8caba4119e043	f1c7f353075ce59d8a6f3cf58f419c9c	delivered	9
0707e2f91c707859b79e711d852150c8	09d1bc53f084c9c2a597d1a7d6329ecb	0db783cfd3b73998abc6e10e59a102f	888bc229e8f4a4bb412bdea51c650a49	delivered	3
070b4a621a6d06cdf99122e5c04b353a	9bf7217158566072ba308c3d057bbc44	53243585a1d6dc2643021fd1853d8905	6ae3f85f46a37844372252496c77b4d8	delivered	9
0a76ae785560c5882874b2738c60d958	7c2b2f906d186327dc6e0fe22ed570d0	1caf283236cd69af44cbc09a0a1e7d32	45ee7eeb2da75b49edaa30e7b01e04d2	delivered	8
0af1da637acdb5fdd0d77bd209530157	d3e29a43c924fd6064e10fe50ebc8925	1e47defeeadeca0e9a18fa5a9311e735	27ceea27f2ac01a3b74b92e5b0a72a29	delivered	6
0c139f0f63f0a404368637f931e98d01	80348ce139d0380cc6cdb14c2ac5cf3b	4e922959ae960d389249c378d1c939f5	4c1e109ecdff58453de365d217cefa64c	delivered	4

## Project workflow:

- The AWS Cloud environments setup
  - S3 to store the “raw data” (Data lake)
  - Redshift (Datawarehouse)
  - AWS Credentials for application to access AWS
- The source data (raw CSV file) will be stored in the AWS S3
  - The source data will be loaded into S3 by manually every month
- The Data lake process will load the “raw data” and produce the “cleaned data” in AWS S3
  - Datatransformation & cleansing by PySpark
  - Produce 1 table of final table
  - The output table is in CSV format
  - Table (CSV file) is partitioned by “year” to be easy for executed
  - Example S3 repository: “s3://brazilian-bucket-final/final\_table/”
- The Datawarehouse process will load the final table to AWS Redshift
  - Data transformation & load by Python



- b. Monthly schedule on the 1st of each month starts on 1-Feb,2017
  - c. Load final\_table from S3 into Redshift
  - d. Load and filtering “year” = 2017 (Team will change each year based on modifying each year)
  - e. transform “cleaned data” to produce “OLAP data”
  - f. The data warehouse table is partitioned by “year”
5. The dashboard for data visualization using Tableau Desktop
- a. Connect Tableau Desktop to Redshift with information in step 1.b
  - b. Select the database and data which collect the necessary information
  - c. Create the dashboard to answer the problem or questions
  - d. Publish the dashboard to Tableau Public for online access
6. The workflow orchestration using Airflow
- a. Steps 1 - 4 should be fully automated and controlled by Airflow
  - b. However, we use the Airflow to only automate and control for step 4 Datawarehouse process due to the limitation of sources and manpower.

### Project implementation instruction

For the full implementation instruction (step-by-step), please find the information here: [link](#)

## Final Dashboard

Please find the dashboard online here: [link](#)



This dashboard represents the analysis of sales information from the Brazilian E-commerce data set which is to answer all the questions and targets for understanding the trend & overall picture.

Which contains with:

1. Total sales amount compare in each category: show segmentation of each category sales proportion
2. Total sales amount in each state: To represent the contribution margin of each state
3. Payment type contribution: Show the margin size of each payment type
4. Average basket size per product category in each state: Show an average basket size of each product category based on each customer state (Need to be sorting a category)
5. Sales movement per state during each month: Represent a trend of sales revenue in each month

## Summary

Starting from raw data, the first step of the whole process is to collect inquiries from the end users and tried to understand an overview of data. Then, ETL process will be used to transform & utilize the data by following the data model.

After that, we create a pipeline of data flowing thru the data lake (S3) and data warehouse (Redshift) to build a final table containing all necessary features that we can process in the visualize to answer the question and even solve end users' problem that we got at the beginning.