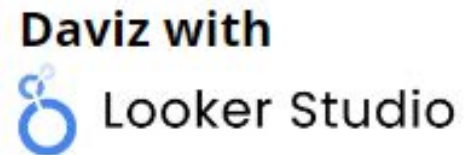


FULLSTACK INTENSIVE BOOTCAMP

MySkill

Data Analyst



Final Project

Kelompok 16

Our Team



Muhammad Tamam Setia



Devina Ellysia Alfiany



Muhammad Syauqirrahman Kancana



Taufik Yasir Sukarda

Our Mentor



Kak Rifki

Table of Content

1. Final Project SQL
 - a. Overview
 - b. Business Questions
 - c. Results
2. Final Project Python
 - a. Overview
 - b. Results
3. Final Project Data Visualization
 - a. Overview
 - b. Business Questions
 - c. Results

Final Project SQL

Overview

The data used is data from Kaggle: [Pakistan's Largest E-Commerce Dataset](#) with some changes to make it easier to practice using sql. The price listed has been converted 1 Rupee equals IDR 58.

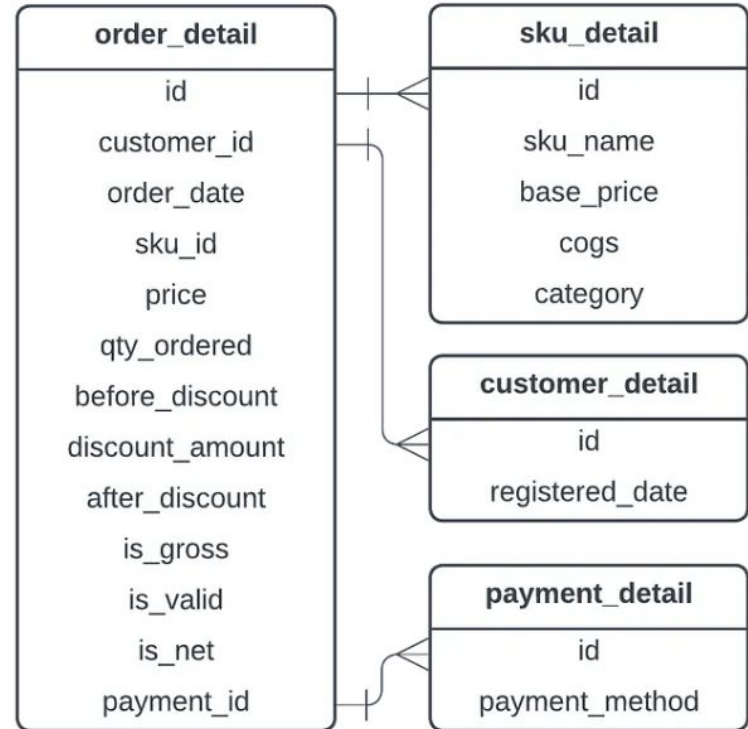


Image source: [Medium Ferry Andhika](#)

Business Questions

1. Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (after_discount) paling besar? Gunakan is_valid = 1 untuk memfilter data transaksi.
2. Selama transaksi yang terjadi selama 2021, pada bulan apa total jumlah pelanggan (unique), total order (unique) dan total jumlah kuantitas produk paling banyak? Gunakan is_valid = 1 untuk memfilter data transaksi.
3. Selama transaksi yang terjadi selama 2022, kategori apa yang menghasilkan nilai transaksi paling besar? Gunakan is_valid = 1 untuk memfilter data transaksi.
4. Bandingkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022. Gunakan is_valid = 1 untuk memfilter data transaksi.
5. Tampilkan Top 10 sku_name (beserta kategorinya) berdasarkan nilai transaksi yang terjadi selama tahun 2022. Tampilkan juga total jumlah pelanggan (unique), total order (unique) dan total jumlah kuantitas. Gunakan is_valid = 1 untuk memfilter data transaksi.
6. Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022 (berdasarkan total unique order). Gunakan is_valid = 1 untuk memfilter data transaksi.
7. Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya. Gunakan is_valid = 1 untuk memfilter data transaksi.
 - a. Samsung
 - b. Apple
 - c. Sony
 - d. Huawei
 - e. Lenovo
8. Seperti pertanyaan no. 3, buatlah perbandingan dari nilai profit tahun 2021 dan 2022 pada tiap kategori. Kemudian buatlah selisih % perbedaan profit antara 2021 dengan 2022 ($\text{profit} = \text{after_discount} - (\text{cogs} * \text{qty_ordered})$). Gunakan is_valid = 1 untuk memfilter data transaksi.
9. Tampilkan top 5 SKU dengan kontribusi profit paling tinggi di tahun 2022 berdasarkan kategori paling besar pertumbuhan profit dari 2021 ke 2022 (berdasarkan hasil no 8). Gunakan is_valid = 1 untuk memfilter data transaksi.
10. Tampilkan jumlah unique order yang menggunakan top 5 metode pembayaran (soal no 6) berdasarkan kategori produk selama tahun 2022. Gunakan is_valid = 1 untuk memfilter data transaksi.

Results

1. Total revenue by month in 2021

	month_2021 text	total_revenue double precision
1	November	4783379762
2	September	1617193583
3	October	1440524283
4	August	1281443859
5	December	1280043500
6	July	1071236711
7	January	387831189
8	June	292988288
9	March	277609852
10	May	274319573
11	February	269133289
12	April	248972318

Total rows: 12 of 12

Query complete 00:00:01.605

-- Query No.1

SELECT

TO_CHAR(order_date, 'Month') AS month_2021,
ROUND(SUM(after_discount)) AS total_revenue

FROM

order_detail

WHERE

is_valid =1 AND

order_date BETWEEN '2021-01-01' AND '2021-12-31'

GROUP BY

month_2021

ORDER BY

total_revenue DESC;

Results

2. Total of Customers, Orders, and Quantity, by month in 2021

	month_2021 text	total_customer bigint	total_order bigint	total_qty bigint
1	November	13885	22414	47385
2	September	3876	5282	10148
3	December	3332	4842	8339
4	October	2860	4428	8522
5	August	2752	4019	7725
6	July	2056	3132	7968
7	January	1097	1331	3150
8	March	933	1101	2481
9	February	918	1119	2340
10	April	895	1080	2169
11	May	830	992	2098
12	June	814	974	1859
Total rows: 12 of 12		Query complete 00:00:04.403		

-- Query No.2

```
SELECT
    TO_CHAR(order_date, 'Month') AS month_2021,
    COUNT(DISTINCT customer_id) AS total_customer,
    COUNT(DISTINCT id) AS total_order,
    SUM(qty_ordered) AS total_qty
FROM
    order_detail
WHERE
    is_valid =1 AND
    order_date BETWEEN '2021-01-01' AND '2021-12-31'
GROUP BY
    1
ORDER BY
    2 DESC;
```

Results

3. Category with the Biggest Income every Month in 2022

	category text	total_revenue double precision
1	Mobiles & Tablets	8556323757
2	Appliances	2265213375
3	Entertainment	1688656686
4	Women Fashion	1676060313
5	Men Fashion	990800833
6	Computing	547579974
7	Superstore	435047343
8	Beauty & Grooming	394193739
9	Home & Living	313040278
10	Health & Sports	224672234
11	Soghaat	153669298
12	Kids & Baby	142530948
13	Others	74523562
14	School & Education	32458049
15	Books	11255822

Total rows: 15 of 15

Query complete 00:00:03.012

-- Query No.3

```
SELECT
    sku.category,
    ROUND(SUM(ord.after_discount)) AS total_revenue
FROM
    order_detail AS ord
LEFT JOIN
    sku_detail AS sku
    ON ord.sku_id = sku.id
WHERE
    order_date BETWEEN '2021-01-01' AND '2021-12-31' AND
    is_valid = 1
GROUP BY
    1
ORDER BY
    2 DESC;
```

Results

4. Comparison of Monthly Income in 2021 and 2022, for Each Product Category

	category text	year_2021 double precision	year_2022 double precision	different_revenue double precision
1	Others	171613712	74523562	-97090150
2	Soghaat	225961733	153669298	-72292435
3	Books	21991531	11255822	-10735709
4	Men Fashion	991653117	990800833	-852284
5	School & Education	28808675	32458049	3649374
6	Home & Living	306034894	313040278	7005384
7	Health & Sports	187743236	224672234	36928998
8	Beauty & Grooming	353969385	394193739	40224354
9	Kids & Baby	101165297	142530948	41365651
10	Computing	487602682	547579974	59977292
11	Superstore	220967052	435047343	214080291
12	Appliances	1971541230	2265213375	293672145
13	Entertainment	1224956504	1688656686	463700182
14	Women Fashion	621113392	1676060313	1054946921
15	Mobiles & Tablets	6309553768	8556323757	2246769989
Total rows: 15 of 15		Query complete 00:00:01.262		

```
-- Query No.4
WITH tab_revenue AS (
SELECT
    EXTRACT('year' FROM order_date) AS year_order,
    sku.category,
    ROUND(SUM(ord.after_discount)) AS total_revenue
FROM
    order_detail AS ord
LEFT JOIN
    sku_detail AS sku
ON ord.sku_id = sku.id
WHERE
    EXTRACT('year' FROM order_date) IN (2021, 2022) AND
    is_valid = 1
GROUP BY
    sku.category,
    EXTRACT('year' FROM order_date)
ORDER BY
    total_revenue DESC
)
SELECT
    *,
    (year_2022 - year_2021) AS different_revenue
FROM
    (
    SELECT
        category,
        SUM(CASE
            WHEN (year_order = 2021) THEN total_revenue
            ELSE NULL
            END) AS year_2021,
        SUM(CASE
            WHEN (year_order = 2022) THEN total_revenue
            ELSE NULL
            END) AS year_2022
    FROM
        tab_revenue
    GROUP BY
        category) AS pivot
ORDER BY
    different_revenue DESC;
```

Results

5. Top 10 sku_name Based on Revenue Value in 2022

	sku_name text	category text	total_revenue double precision	total_customer bigint	total_order bigint	total_qty bigint
1	IDROID_BALRX7-Gold	Mobiles & Tablets	865224544	286	490	1632
2	IDROID_BALRX7-Jet black	Mobiles & Tablets	525626251	14	15	1014
3	infinix_Zero 4-Grey	Mobiles & Tablets	443381130	158	233	334
4	IDROID_BALRX7-Jet black	Mobiles & Tablets	298041332	306	427	542
5	iphone-7-32gb-wof-Matt Black	Mobiles & Tablets	289575962	33	48	60
6	Infinix Hot 4-Black	Mobiles & Tablets	278470817	212	266	396
7	Infinix Hot 4-Gold	Mobiles & Tablets	272945143	173	244	391
8	AYS_32B8500-32-Inches	Entertainment	192536626	170	188	199
9	closecomfort_PC8	Appliances	169312440	81	103	105
10	Haier_HSU-18-HNF	Appliances	149577215	44	46	49

Total rows: 10 of 10 Query complete 00:00:04.403

-- Query No.5

SELECT

sku.sku_name,

sku.category,

ROUND(SUM(ord.after_discount)) AS total_revenue,

COUNT(DISTINCT ord.customer_id) AS total_customer,

COUNT(DISTINCT ord.id) AS total_order,

SUM(ord.qty_ordered) AS total_qty

FROM

order_detail AS ord

LEFT JOIN

(

SELECT

id,

sku_name,

category

FROM

sku_detail

) AS sku

ON ord.sku_id = sku.id

WHERE

is_valid = 1 AND

date_part('Year', order_date) = 2022

GROUP BY

sku.category,

sku.sku_name

ORDER BY

total_revenue DESC

LIMIT 10;

Results

6. Top 5 Payment Methods in 2022

	payment_method text	total_order bigint
1	cod	42609
2	Payaxis	5341
3	Easypay	1443
4	customercredit	1378
5	jazzwallet	1368



Total rows: 5 of 5

Query complete 00:00:06.358

```
-- Query No.6
SELECT
    pay.payment_method,
    COUNT(DISTINCT ord.id) AS total_order
FROM
    order_detail AS ord
LEFT JOIN
    payment_detail AS pay
    ON ord.payment_id = pay.id
WHERE
    order_date BETWEEN '2022-01-01' AND '2022-12-31' AND
    is_valid = 1
GROUP BY
    pay.payment_method
ORDER BY
    total_order DESC
LIMIT 5;
```

Results

7. Top 5 Products based on revenue value

	brand text 	total_revenue double precision 
1	Samsung	3757776955
2	Apple	1670041035
3	Huawei	1012497392
4	Lenovo	393545797
5	Sony	190611121
Total rows: 5 of 5		Query complete 00:00:01.605

```
-- Query No.7
WITH tab_brand AS (
SELECT
  id,
  (CASE
    WHEN LOWER(sku_name) LIKE '%samsung%' THEN 'Samsung'
    WHEN LOWER(sku_name) LIKE '%apple%' OR LOWER(sku_name) LIKE '%iphone%' THEN 'Apple'
    WHEN LOWER(sku_name) LIKE '%sony%' THEN 'Sony'
    WHEN LOWER(sku_name) LIKE '%huawei%' THEN 'Huawei'
    WHEN LOWER(sku_name) LIKE '%lenovo%' THEN 'Lenovo'
  END) AS brand
FROM
  sku_detail
)
SELECT
  sku.brand,
  ROUND(SUM(ord.after_discount)) AS total_revenue
FROM
  order_detail AS ord
LEFT JOIN
  tab_brand AS sku
ON ord.sku_id = sku.id
WHERE
  is_valid = 1 AND
  brand IS NOT NULL
GROUP BY
  sku.brand
ORDER BY
  total_revenue DESC;
```


Results

8. Comparison of Profit Value in 2021 and 2022 for Each Category in Percentage Form

	category text	year_2021 double precision	year_2022 double precision	growth_profit double precision
1	Books	6102373	2651116	-57
2	Soghaat	44221633	31300608	-29
3	Others	17792956	13834450	-22
4	Men Fashion	226619893	182533481	-19
5	Kids & Baby	23428535	19504770	-17
6	Beauty & Grooming	70582139	75062659	6
7	Home & Living	55279202	59669888	8
8	School & Education	7210287	7934837	10
9	Health & Sports	40445160	47521558	17
10	Mobiles & Tablets	1396877440	1652332907	18
11	Appliances	355859880	431194679	21
12	Entertainment	185811916	265278570	43
13	Computing	81856374	117411142	43
14	Superstore	28866179	71091137	146
15	Women Fashion	129540656	345456561	167

Total rows: 15 of 15

Query complete 00:00:01.425

```
-- Query No.8
WITH tab_profit AS (
SELECT
    EXTRACT('Year' FROM order_date) AS year_order,
    sku.category,
    SUM(ord.after_discount - (sku.cogs * ord.qty_ordered)) AS total_profit
FROM
    order_detail AS ord
LEFT JOIN
    sku_detail AS sku
    ON ord.sku_id = sku.id
WHERE
    order_date BETWEEN '2021-01-01' AND '2022-12-31' AND
    is_valid = 1
GROUP BY
    sku.category, year_order
ORDER BY
    total_profit DESC
),
year_profit AS (
SELECT
    category,
    ROUND(SUM(CASE
        WHEN (year_order = 2021) THEN total_profit
        ELSE NULL
        END)) AS year_2021,
    ROUND(SUM(CASE
        WHEN (year_order = 2022) THEN total_profit
        ELSE NULL
        END)) AS year_2022
FROM
    tab_profit
GROUP BY
    category
)
SELECT
    *,
    ROUND(((year_2022 / year_2021) - 1) * 100) AS growth_profit
FROM
    year_profit
ORDER BY
    growth_profit;
```

Results

9. Top 5 SKUs with the Highest Profit Contribution in 2022

	sku_name text	total_profit double precision
1	sanasafinaz_SS-3A	3498908
2	sanasafinaz_SS-9B	3153692
3	sanasafinaz_SS-13A	3013622
4	sanasafinaz_SS-5B	2902320
5	sanasafinaz_SS-10A	2871696

Total rows: 5 of 5 Query complete 00:00:02.877

```
-- Query No.9
WITH tab_profit AS (
SELECT
    ord.id,
    sku.sku_name,
    ord.after_discount - (sku.cogs * ord.qty_ordered) AS profit
FROM
    order_detail AS ord
LEFT JOIN
    sku_detail AS sku
    ON sku.id = ord.sku_id
WHERE
    is_valid = 1 AND
    order_date BETWEEN '2022-01-01' AND '2022-12-31' AND
    sku.category = 'Women Fashion'
)
SELECT
    sku_name,
    SUM(profit) AS total_profit
FROM
    tab_profit
GROUP BY
    sku_name
ORDER BY
    total_profit DESC
LIMIT 5;
```


Results

10. Total of Unique Order Top 5 Payment Methods in Each Product Category, 2022

	category text	cod bigint	easypay bigint	payaxis bigint	customercredit bigint	jazzwallet bigint
1	Men Fashion	10510	186	928	300	195
2	Mobiles & Tablets	9853	618	1804	369	208
3	Women Fashion	7599	201	843	176	89
4	Beauty & Grooming	4125	85	301	108	89
5	Home & Living	3314	74	295	97	60
6	Soghaat	3065	63	236	111	109
7	Appliances	2808	175	893	131	58
8	Kids & Baby	2635	81	306	66	85
9	Health & Sports	2440	41	268	67	40
10	Superstore	2138	81	531	77	700
11	Computing	1287	82	344	44	29
12	Entertainment	1152	129	547	45	35
13	Others	1066	35	103	34	41
14	School & Education	457	11	40	9	9
15	Books	336	4	10	4	4
Total rows: 15 of 15 Query complete 00:00:03.301						

```
-- Query No.10
SELECT
    sku.category,
    COUNT(DISTINCT CASE WHEN pay.payment_method = 'cod' THEN ord.id END) AS cod,
    COUNT(DISTINCT CASE WHEN pay.payment_method = 'Easypay' THEN ord.id END) AS easypay,
    COUNT(DISTINCT CASE WHEN pay.payment_method = 'Payaxis' THEN ord.id END) AS payaxis,
    COUNT(DISTINCT CASE WHEN pay.payment_method = 'customercredit' THEN ord.id END) AS customercredit,
    COUNT(DISTINCT CASE WHEN pay.payment_method = 'jazzwallet' THEN ord.id END) AS jazzwallet
FROM
    order_detail AS ord
LEFT JOIN
    payment_detail pay
    ON pay.id = ord.payment_id
LEFT JOIN
    sku_detail sku
    ON sku.id = ord.sku_id
WHERE
    is_valid = 1 AND
    order_date BETWEEN '2022-01-01' AND '2022-12-31'
GROUP BY
    sku.category
ORDER BY
    cod DESC;
```

Final Project Python

Overview

Data yang digunakan adalah [Hotel Booking Demand](#) dari Jurnal Internasional milik Antonio. N, de Almeida. A, dan Nunes. L

pengerjaan diawali dengan import library python yang kita butuhkan dalam proses pengerjaan

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
sns.set()
```

kemudian menggunakan library pandas untuk input data

```
df_hotels = pd.read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-02-11/hotels.csv')
df_hotels.reset_index(inplace=True)
df_hotels = df_hotels.rename(columns={'index':'id'})
df_hotels['id'] = df_hotels['id'] + 1
```

Results

Nomor 1

Buatlah fungsi dengan :

1 argumen berupa dataframe untuk mengecek data type,
untuk mengecek jumlah null value,
untuk mengecek percent null value,
serta jumlah unique value tiap kolom yang ada di sebuah dataframe

```
def check_values(df):  
    data = []  
    for co in df.columns:  
        data.append([co, \n  
                    df[co].dtype, \n  
                    df[co].isna().sum(), \n  
                    round(100*(df[co].isnull().sum()/len(df_hotels.index)),2), \n  
                    df[co].nunique()\n  
                    ])  
  
    return pd.DataFrame(data=data, columns=['Nama Column', 'Tipe Data', 'Jumlah Null', 'Persen Null', 'Jumlah Unik'])
```

	Nama Column	Tipe Data	Jumlah Null	Persen Null	Jumlah Unik
0	id	int64	0	0.00	119390
1	hotel	object	0	0.00	2
2	is_canceled	int64	0	0.00	2
3	lead_time	int64	0	0.00	479
4	arrival_date_year	int64	0	0.00	3
5	arrival_date_month	object	0	0.00	12
6	arrival_date_week_number	int64	0	0.00	53
7	arrival_date_day_of_month	int64	0	0.00	31
8	stays_in_weekend_nights	int64	0	0.00	17
9	stays_in_week_nights	int64	0	0.00	35
10	adults	int64	0	0.00	14
11	children	float64	4	0.00	5
12	babies	int64	0	0.00	5
13	meal	object	0	0.00	5
14	country	object	488	0.41	177
15	market_segment	object	0	0.00	8
16	distribution_channel	object	0	0.00	5
17	is_repeated_guest	int64	0	0.00	2
18	previous_cancellations	int64	0	0.00	15
19	previous_bookings_not_canceled	int64	0	0.00	73
20	reserved_room_type	object	0	0.00	10
21	assigned_room_type	object	0	0.00	12
22	booking_changes	int64	0	0.00	21
23	deposit_type	object	0	0.00	3

Results

Nomor 2

Ada berapa berapa jumlah pengunjung yang membatalkan reservasi dan yang tidak? Dan dari jumlah tersebut buatlah kesimpulan mengenai proporsisi masing-masing!

```
df_hotels['is_canceled'].value_counts()
```

```
0    75166  
1    44224  
Name: is_canceled, dtype: int64
```

```
df_hotels['is_canceled'].value_counts(normalize=True)
```

```
0    0.629584  
1    0.370416  
Name: is_canceled, dtype: float64
```

```
sns.countplot(data=df_hotels, x='is_canceled')  
plt.title('Cancelled', fontsize=20)  
plt.show()
```



Results

Nomor 3

A. Untuk "City Hotel", berapa persen reservasi yang dibatalkan?

B. Untuk "Resort Hotel", berapa persen reservasi yang dibatalkan?

Di hotel jenis apa ditemukan proporsi reservasi yang dibatalkan lebih besar?

```
df_hotels['is_canceled'][df_hotels['hotel'] == 'City Hotel'].value_counts(1)
```

```
0    0.58273
1    0.41727
Name: is_canceled, dtype: float64
```

```
len(df_hotels[(df_hotels.hotel=='City Hotel')&(df_hotels.is_canceled==1)]) / len(df_hotels[(df_hotels.hotel=='City Hotel')])
```

```
0.41726963317786464
```

```
df_hotels['is_canceled'][df_hotels['hotel'] == 'Resort Hotel'].value_counts()
```

```
0    28938
1    11122
Name: is_canceled, dtype: int64
```

```
len(df_hotels[(df_hotels.hotel=='Resort Hotel')&(df_hotels.is_canceled==1)]) / len(df_hotels[(df_hotels.hotel=='Resort Hotel')])
```

```
0.27763354967548676
```

Results

Nomor 4

Lakukan filter sehingga hanya menampilkan data pengunjung yang tidak membatalkan reservasi. Dan simpan hasilnya dalam variabel `df_checkout`.

```
df_checkout = df_hotels[df_hotels['is_canceled'] == 0]  
df_checkout.shape
```

```
(75166, 33)
```

Results

Nomor 5

A. Tampilkan jumlah reservasi tiap bulan kedatangan untuk masing-masing jenis hotel.

B. Lalu di bulan apa terdapat reservasi yang paling banyak di masing-masing jenis hotel? Buatlah kesimpulan apakah trennya sama di kedua jenis hotel?

C. Lakukan seperti point B namun dengan nama bulan yang sudah di-mapping menjadi bulan dalam angka

5A

```
df_checkout.groupby(['hotel', 'arrival_date_month'], sort=0).size()
```

hotel	arrival_date_month	
Resort Hotel	July	3137
	August	3257
	September	2102
	October	2577
	November	1976
	December	2017
	January	1868
	February	2308
	March	2573
	April	2550
	May	2535
	June	2038
City Hotel	July	4782
	August	5381
	September	4290
	October	4337
	November	2696
	December	2392
	January	2254
	February	3064
	March	4072
	April	4015
	May	4579
	June	4366

dtype: int64

Results

Nomor 5

A. Tampilkan jumlah reservasi tiap bulan kedatangan untuk masing-masing jenis hotel.

B. Lalu di bulan apa terdapat reservasi yang paling banyak di masing-masing jenis hotel? Buatlah kesimpulan apakah trennya sama di kedua jenis hotel?

C. Lakukan seperti point B namun dengan nama bulan yang sudah di-mapping menjadi bulan dalam angka

5B

```
df_checkout.groupby(['hotel', 'arrival_date_month'])\
['id'].nunique().sort_values(ascending=False)
```

hotel	arrival_date_month	
City Hotel	August	5381
	July	4782
	May	4579
	June	4366
	October	4337
Resort Hotel	September	4290
	March	4072
	April	4015
	August	3257
	July	3137
City Hotel	February	3064
	November	2696
Resort Hotel	October	2577
	March	2573
	April	2550
	May	2535
City Hotel	December	2392
Resort Hotel	February	2308
City Hotel	January	2254
Resort Hotel	September	2102
	June	2038
	December	2017
	November	1976
	January	1868

Name: id, dtype: int64

Results

5C

Nomor 5

A. Tampilkan jumlah reservasi tiap bulan kedatangan untuk masing-masing jenis hotel.

B. Lalu di bulan apa terdapat reservasi yang paling banyak di masing-masing jenis hotel? Buatlah kesimpulan apakah trennya sama di kedua jenis hotel?

C. Lakukan seperti point B namun dengan nama bulan yang sudah di-mapping menjadi bulan dalam angka

```
import calendar

month_dict = {month: index for index, month in enumerate(calendar.month_name) if month}
df_checkout['arrival_date_month_num'] = df_checkout['arrival_date_month'].map(month_dict)
df_checkout.groupby(['hotel', 'arrival_date_month_num'], sort=0).size()
```

hotel	arrival_date_month_num	
Resort Hotel	7	3137
	8	3257
	9	2102
	10	2577
	11	1976
	12	2017
	1	1868
	2	2308
	3	2573
	4	2550
	5	2535
	6	2038
City Hotel	7	4782
	8	5381
	9	4290
	10	4337
	11	2696
	12	2392
	1	2254
	2	3064
	3	4072
	4	4015
	5	4579
	6	4366

dtype: int64

Results

Nomor 6

A. Buat sebuah kolom baru bernama `arrival_date` yang berisi info lengkap tentang tahun, bulan, dan tanggal kedatangan.

B. Ubah kolom menjadi tipe datetime.

Hint: gabungkan tahun, bulan, dan tanggal menjadi format `yyyy-mm-dd`

```
df_checkout['arrival_date_month_num'].astype('str').str.pad(2,fillchar='0')
df_checkout['arrival_date'] = \
    df_checkout['arrival_date_year'].astype('str') + '-' + \
    df_checkout.arrival_date_month_num.astype('str').str.pad(2,fillchar='0') + '-' + \
    df_checkout.arrival_date_day_of_month.astype('str').str.pad(2,fillchar='0')
df_checkout['arrival_date'].tail()
```

```
<ipython-input-17-2e02c72b9bb4>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_checkout['arrival_date'] = \
119385    2017-08-30
119386    2017-08-31
119387    2017-08-31
119388    2017-08-31
119389    2017-08-29
Name: arrival_date, dtype: object
```

Results

Nomor 6

A. Buat sebuah kolom baru bernama `arrival_date` yang berisi info lengkap tentang tahun, bulan, dan tanggal kedatangan.

B. Ubah kolom menjadi tipe datetime.

Hint: gabungkan tahun, bulan, dan tanggal menjadi format `yyyy-mm-dd`

```
df_checkout['arrival_date'] = pd.to_datetime(df_checkout.arrival_date)
df_checkout['arrival_date'].tail()
```

```
<ipython-input-35-595891660c86>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy.

```
df_checkout['arrival_date'] = pd.to_datetime(df_checkout.arrival_date)
119385    2017-08-30
119386    2017-08-31
119387    2017-08-31
119388    2017-08-31
119389    2017-08-29
Name: arrival_date, dtype: datetime64[ns]
```

Results

Nomor 7

Mari kita bermain dengan time-series data menggunakan kolom `arrival_date`.

Buat dataframe yang menunjukkan sbb:
total reservasi harian (`df_reservasi_perhari`)

> ****(!) Stop and think!****

> Apa perbedaan data yang ditunjukkan oleh
`df_reservasi_perhari` dan `df_avg_reservasi_harian`?

```
df_reservasi_perhari = df_checkout.resample('D', on='arrival_date')\
    .size()\
    .reset_index()\
    .rename(columns={0: 'total_reservasi'})

df_reservasi_perhari
```

	arrival_date	total_reservasi
0	2015-07-01	103
1	2015-07-02	36
2	2015-07-03	37
3	2015-07-04	45
4	2015-07-05	37
...
788	2017-08-27	125
789	2017-08-28	147
790	2017-08-29	81
791	2017-08-30	62
792	2017-08-31	89
793 rows × 2 columns		

Results

Nomor 7

Mari kita bermain dengan time-series data menggunakan kolom `arrival_date`.

Buat dataframe yang menunjukkan sbb:
rata-rata reservasi harian di tiap minggu
(`df_avg_reservasi_harian`)

> ***(!) Stop and think!***

> Apa perbedaan data yang ditunjukkan oleh
`df_reservasi_perhari` dan `df_avg_reservasi_harian`?

```
df_avg_reservasi_harian_2 = df_reservasi_perhari.resample('W', on='arrival_date')['total_reservasi']\
                             .mean()\
                             .reset_index()
```

df_avg_reservasi_harian_2

	arrival_date	total_reservasi
0	2015-07-05	51.600000
1	2015-07-12	40.571429
2	2015-07-19	53.857143
3	2015-07-26	53.000000
4	2015-08-02	47.142857
...
109	2017-08-06	101.000000
110	2017-08-13	98.000000
111	2017-08-20	103.714286
112	2017-08-27	103.142857
113	2017-09-03	94.750000

114 rows × 2 columns

Results

Nomor 8

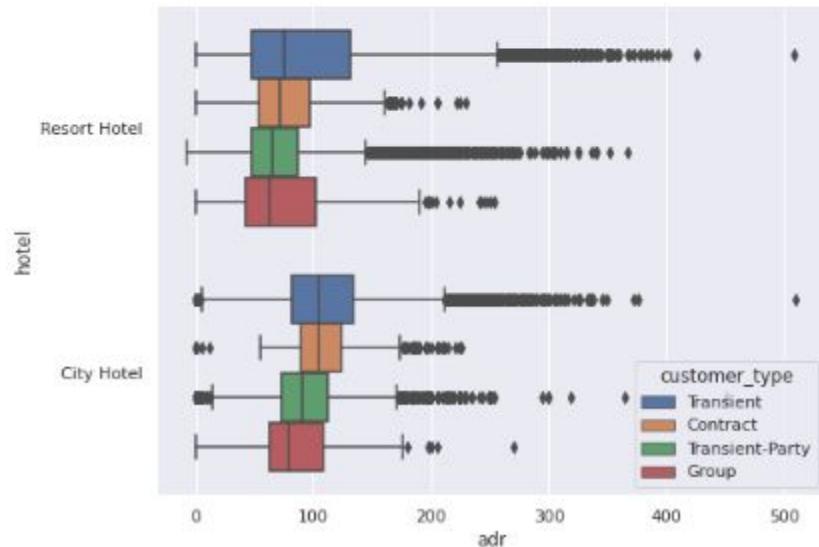
A. Berapa rata-rata ADR berdasarkan jenis hotel dan jenis customer (`customer_type`)?

B. Jenis customer mana yang memiliki ADR paling besar di masing-masing jenis hotel?

```
df_checkout.groupby(['hotel', 'customer_type'])['adr'].mean()

hotel      customer_type
City Hotel  Contract      108.929255
            Group        87.398712
            Transient    110.423280
            Transient-Party 93.705007
Resort Hotel Contract      78.581674
            Group        77.306575
            Transient     96.001928
            Transient-Party 77.204010
Name: adr, dtype: float64
```

```
plt.figure(figsize=(8,6))
sns.boxplot(data=df_checkout, x='adr', y='hotel', hue='customer_type')
plt.show()
```



Results

Nomor 9 (Bonus)

Dengan menggunakan dataframe `df_country` yang berisi informasi nama negara dan kode negaranya,

****Tampilkan**** 10 negara dengan jumlah booking terbesar!

```
df_country = pd.read_csv('https://gist.githubusercontent.com/tadast/8827699/raw/f5cac3d42d16b78348610fc4ec301e9234f82821/countries_codes_and_coordinates.csv')
df_country['code'] = df_country['Alpha-3 code'].str.replace(' ','').str.strip()
df_merged = pd.merge(df_checkout[['id', 'country']],
                     df_country[['Country', 'code']],
                     left_on='country',
                     right_on='code',
                     indicator=True,
                     how='left')
df_merged.Country.value_counts().head(10).sort_values(ascending=False)
```

Portugal	21071
United Kingdom	9676
France	8481
Spain	6391
Germany	6069
Ireland	2543
Italy	2433
Belgium	1868
Netherlands	1717
United States	1596

Name: Country, dtype: int64

Results

Nomor 10 (Bonus)

A. Berapa banyak tamu yang menginap untuk tiap reservasi?

B. Berdasarkan dataset, berapa jumlah tamu paling banyak? Tampilkan juga baris data reservasi yang memiliki jumlah tamu paling banyak.

```
df_checkout['total_guest'] = df_checkout.adults + df_checkout.children + df_checkout.babies
df_checkout['total_guest']
```

```
<ipython-input-63-658b0fb959af>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_checkout['total_guest'] = df_checkout.adults + df_checkout.children + df_checkout.babies
```

```
0      2.0
1      2.0
2      1.0
3      1.0
4      2.0
```

```
...
119385  2.0
119386  3.0
119387  2.0
119388  2.0
119389  2.0
```

```
Name: total_guest, Length: 75166, dtype: float64
```

Results

Nomor 10 (Bonus)

A. Berapa banyak tamu yang menginap untuk tiap reservasi?

B. Berdasarkan dataset, berapa jumlah tamu paling banyak? Tampilkan juga baris data reservasi yang memiliki jumlah tamu paling banyak.

```
df_checkout[df_checkout.total_guest==df_checkout.total_guest.max()].T
```

	46619
id	46620
hotel	City Hotel
is_canceled	0
lead_time	37
arrival_date_year	2016
arrival_date_month	January
arrival_date_week_number	3
arrival_date_day_of_month	12
stays_in_weekend_nights	0
stays_in_week_nights	2
adults	2
children	0.0
babies	10
meal	BB
country	PRT

Final Project Data Visualization

Overview

Data yang digunakan adalah [Hotel Booking Demand](#) dari Jurnal Internasional milik Antonio. N, de Almeida. A, dan Nunes. L, dataset yang sudah *cleaned* dapat diakses [di sini](#).

Business Questions

1. Berapa total booking yang dibuat di masing-masing jenis hotel? Lebih banyak di hotel jenis yang telah terbooking apa? Jelaskan insight apa yang di dapat dari visualisasi tersebut.
2. Tunjukkan visualisasi yang membandingkan jumlah booking oleh turis lokal (local market, asal negara Portugal) dan booking oleh turis inbound (inbound tourism, asal dari negara lain). Dari mana booking paling banyak berasal? Jelaskan insight apa yang di dapat dari visualisasi tersebut.
3. Bagaimana pola ADR di tiap jenis hotel berdasarkan rata-rata ADR di tiap minggu? Apakah di kedua jenis hotel rata-rata ADR naik dan turun di periode (minggu/bulan/musim) yang sama? Jelaskan insight apa yang di dapat dari visualisasi tersebut.
4. Bagaimana cancellation rate dari masing-masing jenis hotel di tiap bulan? Hotel jenis apa yang memiliki cancellation rate paling tinggi? Jelaskan insight apa yang di dapat dari visualisasi tersebut.
5. Berapa jumlah cancelled bookings untuk masing-masing jenis market segment? Di market segment mana cancellation rate-nya paling tinggi? Jelaskan insight apa yang di dapat dari visualisasi tersebut.
6. Hitung persentase total pengunjung hotel di benua Eropa! Jelaskan insight apa yang di dapat dari visualisasi tersebut.

Results

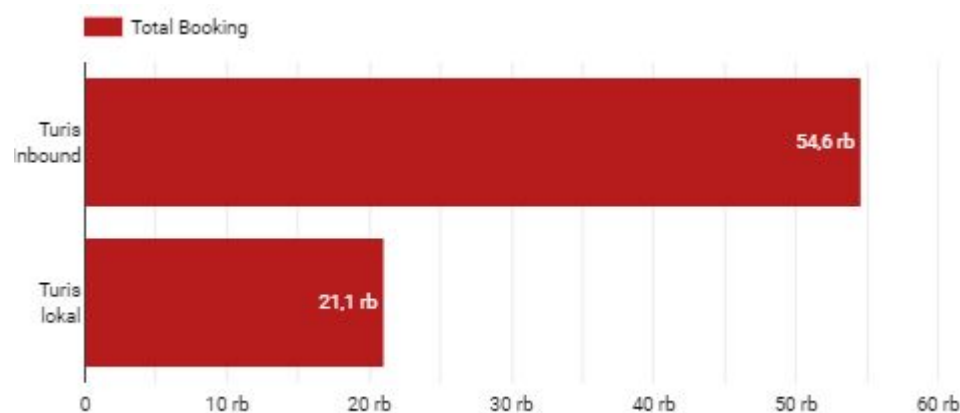
Berapa total booking yang dibuat di masing-masing jenis hotel? Lebih banyak di hotel jenis yang telah terbooking apa? Jelaskan insight apa yang di dapat dari visualisasi tersebut.



City Hotel memiliki total booking **lebih banyak** dari Resort Hotel

Results

Tunjukkan visualisasi yang membandingkan jumlah booking oleh turis lokal (local market, asal negara Portugal) dan booking oleh turis inbound (inbound tourism, asal dari negara lain). Dari mana booking paling banyak berasal? Jelaskan insight apa yang di dapat dari visualisasi tersebut.



Results

Bagaimana pola ADR di tiap jenis hotel berdasarkan rata-rata ADR di tiap minggu? Apakah di kedua jenis hotel rata-rata ADR naik dan turun di periode (minggu/bulan/musim) yang sama? Jelaskan insight apa yang di dapat dari visualisasi tersebut.



Secara menyeluruh resort hotel memiliki demand yang lebih kecil dibanding dengan City Hotel **kecuali pada bulan Juli Agustus**

Results

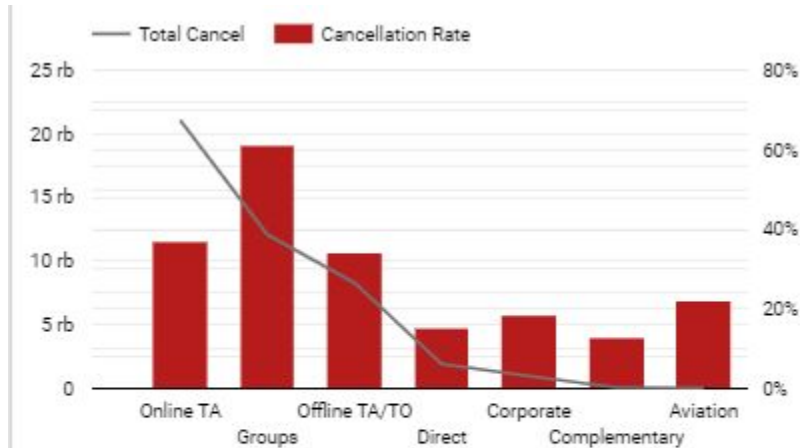
Bagaimana cancellation rate dari masing-masing jenis hotel di tiap bulan? Hotel jenis apa yang memiliki cancellation rate paling tinggi? Jelaskan insight apa yang di dapat dari visualisasi tersebut.



walaupun City hotel memiliki total booking yang lebih tinggi, akan tetapi City hotel memiliki **tingkat pembatalan yang lebih tinggi** juga

Results

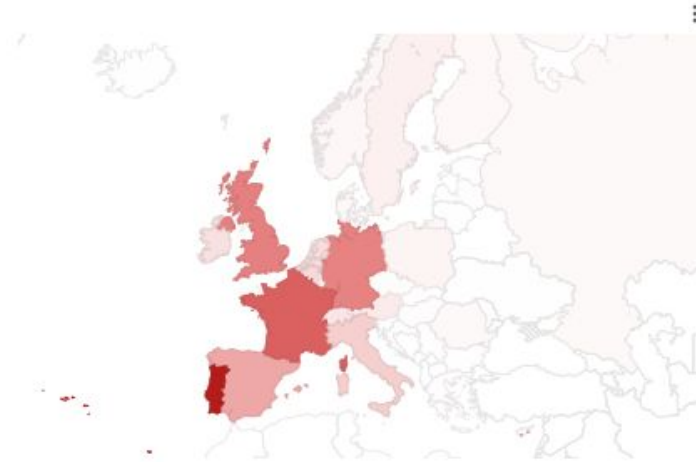
Berapa jumlah cancelled bookings untuk masing-masing jenis market segment? Di market segment mana cancellation rate-nya paling tinggi? Jelaskan insight apa yang di dapat dari visualisasi tersebut.



Untuk segment Online TA, walaupun **total cancelnya lebih tinggi** dari segment yang lain, tetapi **persentasenya lebih rendah** dibanding segment yang lain

Results

Hitung persentase total pengunjung hotel di benua Eropa! Jelaskan insight apa yang di dapat dari visualisasi tersebut.



Berdasarkan persentase kedatangan turis dari eropa, negara Prancis menjadi negara dengan **pengunjung tertinggi** setelah portugal dengan **13,99%**

Results

<https://lookerstudio.google.com/reporting/46d2584c-63bf-48e6-ae57-056b529e789c/page/JDgID>

Thank You