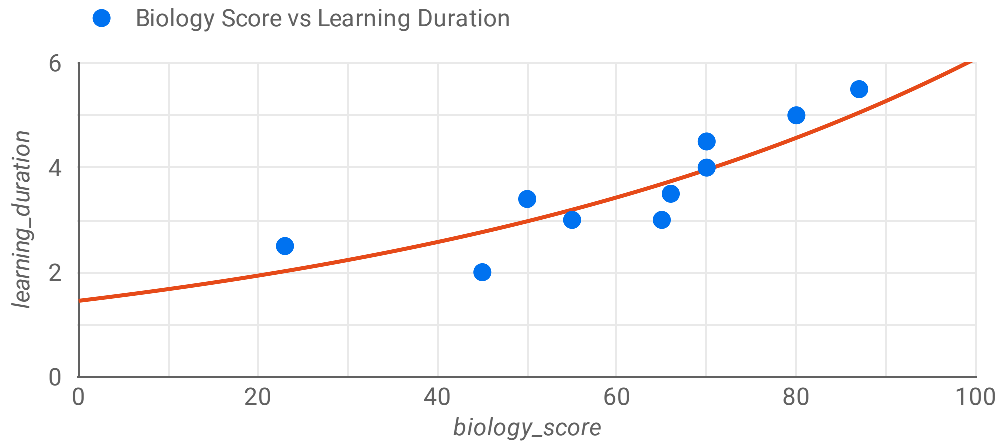
Dashboard Pelatihan Data Analisis dan Visualiasi by Skill Academy

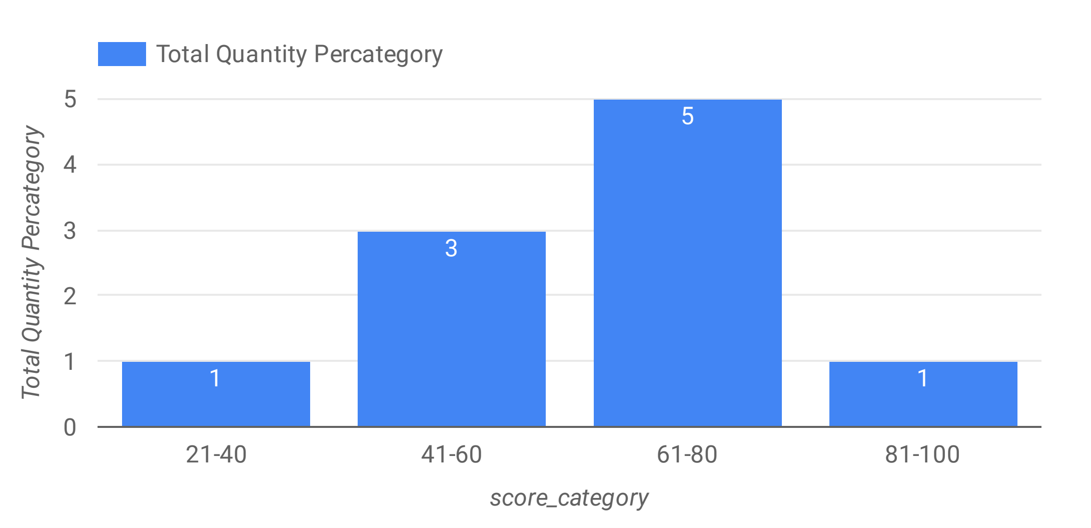
Created By : Muhammad Pajrul Palah

Analisa data nilai ujian biologi siswa dan jumlah durasi waktu belajar sebelum ujian biologi

	Name	biology_score ▾	learning_duration
1.	Cahya	87	5.5
2.	Resa	80	5
3.	Dimas	70	4
4.	Anisa	70	4.5
5.	Eka	66	3.5
6.	Via	65	3
7.	Syifa	55	3
8.	Fatur	50	3.4
9.	Hakim	45	2
10.	Rahman	23	2.5



Dari diagram scatter diatas menunjukan variabel x sebagai score masing-masing siswa sebanyak 10 dan y sebagai total durasi belajar perjam, dimana dari diagram tersebut menunjukan bahwa variabel y mempengaruhi terhadap nilai ujian biologi. (semakin banyak durasi belajar maka semakin besar pula nilai ujian tersebut)



Berdasarkan kategori terdapat 5 siswa dengan jumlah terbanyak dengan nilai di antara 61-80

Median ▾	Average	Variance
65.5	61.1	309.69

Rata-rata dan Median Berada pada nilai 60-65 artinya mengindikasikan bahwa distribusi data cenderung simetris atau memiliki sedikit skewness (kecondongan). Namun, nilai varians yang cukup besar 309.6, menunjukkan bahwa data cenderung memiliki variasi yang signifikan atau tersebar jauh dari nilai rata-ratanya yaitu diantara 21-40 sampai 81-100

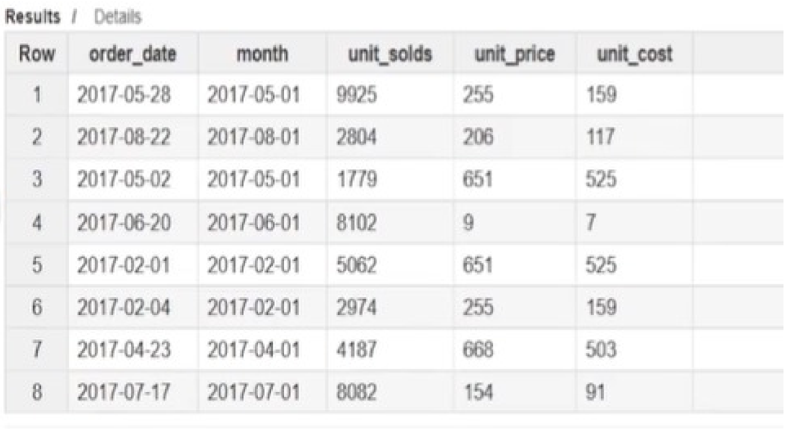
	Name	learning_duration	Model Regresi ▾	biology_score
1.	Cahya	5.5	88.47	87
2.	Resa	5	81.33	80
3.	Anisa	4.5	74.19	70
4.	Dimas	4	67.05	70
5.	Eka	3.5	59.91	66
6.	Fatur	3.4	58.48	50
7.	Via	3	52.77	65
8.	Syifa	3	52.77	55
9.	Rahman	2.5	45.63	23
1...	Hakim	2	38.49	45

Dari tabel diatas adalah merupakan penggunaan model regresi untuk memprediksi berapa kemungkinan nilai yang didapatkan berdasarkan total waktu belajar dan dilihat bahwa antara nilai asli dan prediksi sangat mendekati sama.
Hasil tersebut didapat dengan formula sebagai berkut:
biology_score = 9.93 + (14.28 x learning_duration (in hour))
maka akan didapatkan sebagai hasil diatas.

Structure Query Languange (SQL)

Nested Query, Sub Query, Windows Function, Leg

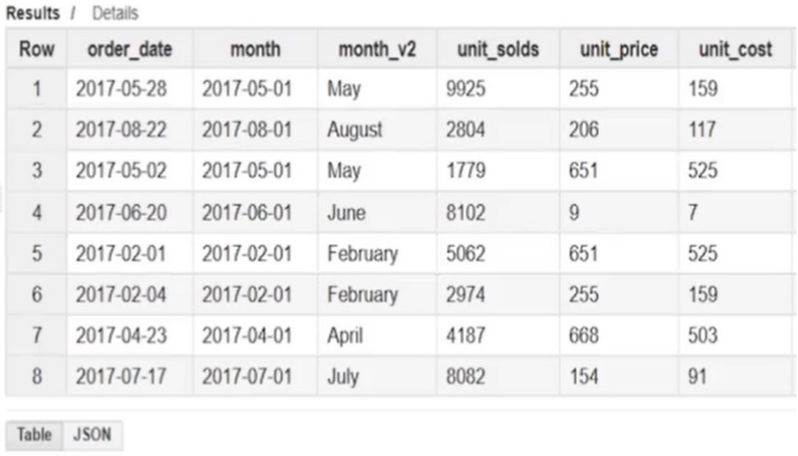
```
#PRAKTIK MEMBUAT TABEL YANG BERISI BULAN
SELECT
   order_date,
   DATE_TRUNK(order_date, MONTH) AS month,
   unit_solds,
   unit_price,
   unit_cost
FROM
  'dummy_dataset.record'
```



Results / Details

Row	order_date	month	unit_solds	unit_price	unit_cost
1	2017-05-28	2017-05-01	9925	255	159
2	2017-08-22	2017-08-01	2804	206	117
3	2017-05-02	2017-05-01	1779	651	525
4	2017-06-20	2017-06-01	8102	9	7
5	2017-02-01	2017-02-01	5062	651	525
6	2017-02-04	2017-02-01	2974	255	159
7	2017-04-23	2017-04-01	4187	668	503
8	2017-07-17	2017-07-01	8082	154	91

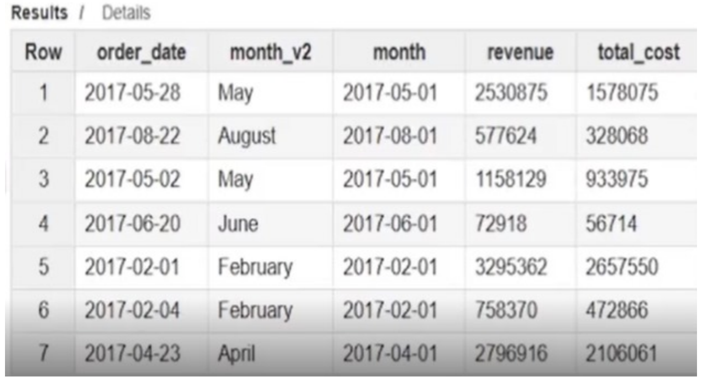
```
SELECT
   order_date,
   DATE_TRUNK(order_date, MONTH) AS month,
   FORMAT_TIMESTAMP("%B", TIMESTAMP(order_date)) AS month_v2
   unit_solds,
   unit_price,
   unit_cost
FROM
  'dummy_dataset.record'
```



Results / Details

Row	order_date	month	month_v2	unit_solds	unit_price	unit_cost
1	2017-05-28	2017-05-01	May	9925	255	159
2	2017-08-22	2017-08-01	August	2804	206	117
3	2017-05-02	2017-05-01	May	1779	651	525
4	2017-06-20	2017-06-01	June	8102	9	7
5	2017-02-01	2017-02-01	February	5062	651	525
6	2017-02-04	2017-02-01	February	2974	255	159
7	2017-04-23	2017-04-01	April	4187	668	503
8	2017-07-17	2017-07-01	July	8082	154	91

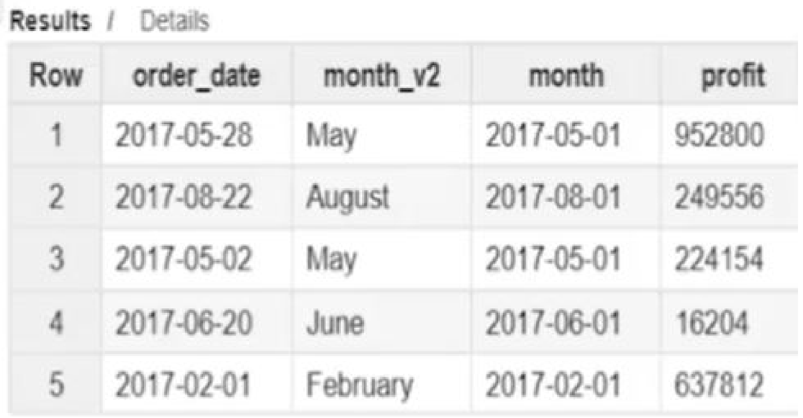
```
#PRAKTIK QUERY NESTED TOTAL REVENUE dan COST
SELECT
   order_date,
   month_v2,
   month,
   (unit_solds * unit_price) AS revenue,
   (unit_sold * unit_cost) AS total_cost,
FROM (
   SELECT
      order_date,
      DATE_TRUNK(order_date, MONTH) AS month,
      FORMAT_TIMESTAMP("%B", TIMESTAMP(order_date)) AS month_v2
      unit_solds,
      unit_price,
      unit_cost
   FROM
      'dummy_dataset.record')
```



Results / Details

Row	order_date	month_v2	month	revenue	total_cost
1	2017-05-28	May	2017-05-01	2530875	1578075
2	2017-08-22	August	2017-08-01	577624	328068
3	2017-05-02	May	2017-05-01	1158129	933975
4	2017-06-20	June	2017-06-01	72918	56714
5	2017-02-01	February	2017-02-01	3295362	2657550
6	2017-02-04	February	2017-02-01	758370	472866
7	2017-04-23	April	2017-04-01	2796916	2106061

```
#PRAKTIK QUERY NESTED TOTAL PROFIT
SELECT
   order_date,
   month_v2,
   month,
   (revenue - total_cost) AS profit
FROM (
  SELECT
     order_date,
     month_v2,
     month,
     (unit_solds * unit_price) AS revenue,
     (unit_sold * unit_cost) AS total_cost,
  FROM (
     SELECT
        order_date,
        DATE_TRUNK(order_date, MONTH) AS month,
        FORMAT_TIMESTAMP("%B", TIMESTAMP(order_date)) AS month_v2
        unit_solds,
        unit_price,
        unit_cost
     FROM
        'dummy_dataset.record'))
```

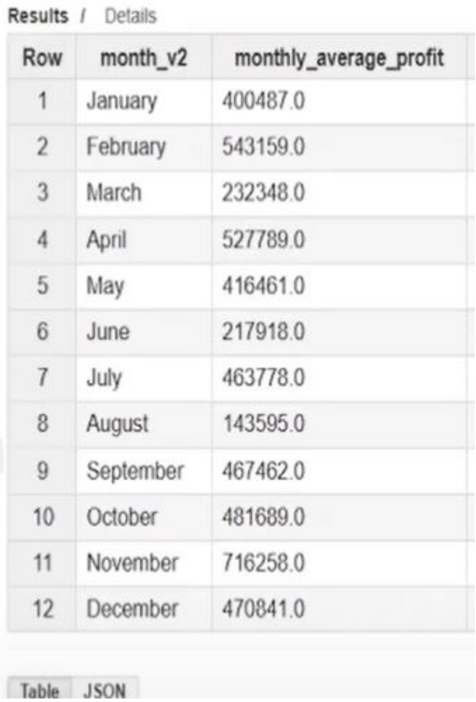


Results / Details

Row	order_date	month_v2	month	profit
1	2017-05-28	May	2017-05-01	952800
2	2017-08-22	August	2017-08-01	249556
3	2017-05-02	May	2017-05-01	224154
4	2017-06-20	June	2017-06-01	16204
5	2017-02-01	February	2017-02-01	637812

#PRAKTIK QUERY NESTED MONTHLY AVERAGE PROFIT

```
SELECT
  month_v2,
  ROUND(avg_profit,0) AS monthly_average_profit
FROM (
  SELECT
    month_v2
    month,
    AVG(profit) AS avg_profit
  FROM (
   SELECT
     order_date,
     month_v2,
     month,
     (revenue - total_cost) AS profit
   FROM (
     SELECT
       order_date,
       month_v2,
       month,
       (unit_solds * unit_price) AS revenue,
       (unit_sold * unit_cost) AS total_cost,
     FROM (
       SELECT
         order_date,
         DATE_TRUNK(order_date, MONTH) AS month,
         FORMAT_TIMESTAMP("%B", TIMESTAMP(order_date)) AS month_v2
         unit_solds,
         unit_price,
         unit_cost
       FROM
         'dummy_dataset.record')))
GROUP BY
  month_v2,
  month
ORDER BY
  month
```

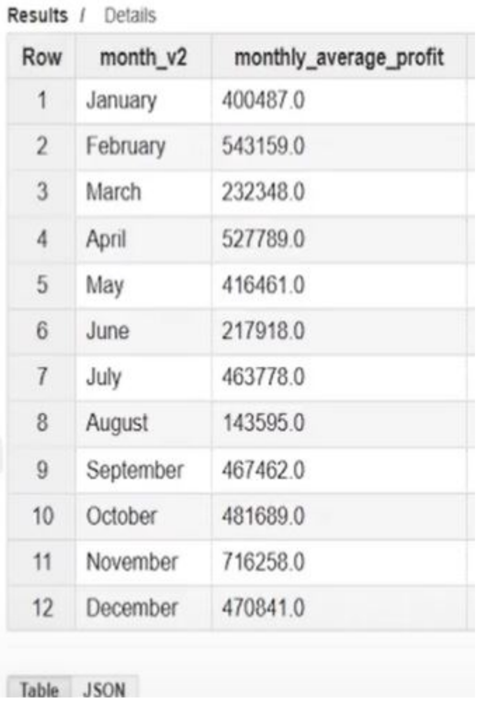


Results / Details

Row	month_v2	monthly_average_profit
1	January	400487.0
2	February	543159.0
3	March	232348.0
4	April	527789.0
5	May	416461.0
6	June	217918.0
7	July	463778.0
8	August	143595.0
9	September	467462.0
10	October	481689.0
11	November	716258.0
12	December	470841.0

#PRAKTIK SUB QUERY (all process from Nested Query)

```
WITH
raw_table AS (
  SELECT
    order_date,
    DATE_TRUNK(order_date, MONTH) AS month,
    FORMAT_TIMESTAMP("%B", TIMESTAMP(order_date)) AS month_v2
    unit_solds,
    unit_price,
    unit_cost
  FROM
  '   dummy_dataset.record'),

  cal_the_rev_and_tc AS (
  SELECT
     order_date,
     month_v2,
     month,
     (unit_solds * unit_price) AS revenue
     (unit_sold * unit_cost) AS total_cost,
   FROM (
     raw_table),

   calc_the_profit AS (
   SELECT
     order_date,
     month,
     month_v2,
     (revenue -  total_cost) AS profit
   FROM
     cal_the_rev_and_tc),

      calc_avg_profit AS (
    SELECT
      month_V2,
      month,
      AVG(profit) AS avg_profit
     FROM
      calc_the_profit
     GROUP BY
       month_v2,
       month)

 SELECT
   month_v2,
   ROUND(avg_profit, 0) AS monthly_average_profit
FROM
   calc_avg_profit
ORDER By
   month
```

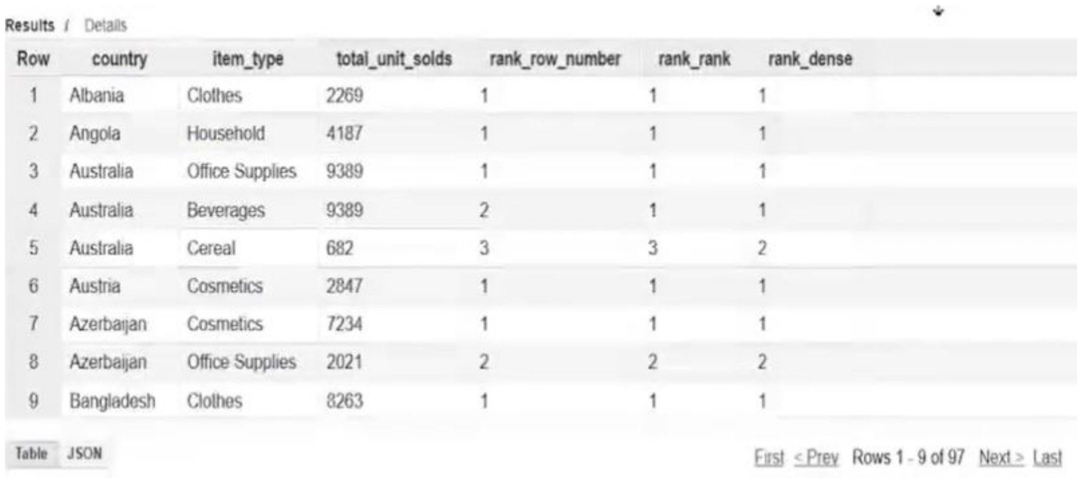


Results / Details

Row	month_v2	monthly_average_profit
1	January	400487.0
2	February	543159.0
3	March	232348.0
4	April	527789.0
5	May	416461.0
6	June	217918.0
7	July	463778.0
8	August	143595.0
9	September	467462.0
10	October	481689.0
11	November	716258.0
12	December	470841.0

#PRAKTIK WINDOWS FUNCTIONS

```
WITH
   raw_data AS
   (

   SELECT
       DISTINCT record.*,
       countries.country
   FROM
       'dummy_dataset.record' AS record
   LEFT JOIN
       'dummy_dataset.countries' AS countries
       ON
       record.country_id = countries.country_id
    ),
    total_solds AS
    (

    SELECT
         country,
         item_type,
         SUM (unit_solds) AS total_unit_solds
    FROM
         raw_data
    GROUP BY
         country,
         item_type
     )

    SELECT
          country,
          item_type,
          total_unit_sold,
          ROW_NUMBER () OVER (PARTITION BY country ORDER BY total_unit_solds DESC) AS rank_row_number,
          RANK () OVER (PARTITION BY country ORDER BY total_unit_solds DESC) AS rank_rank,
          DENSE_RANK () OVER (PARTITION BY country ORDER BY total_unit_solds DESC) AS rank_dense
   FROM
           total_solds
```

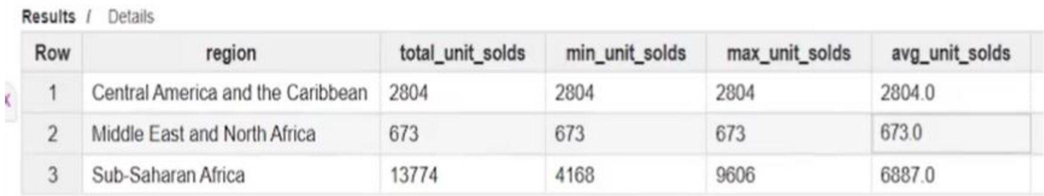


Results / Details

Row	country	item_type	total_unit_solds	rank_row_number	rank_rank	rank_dense
1	Albania	Clothes	2269	1	1	1
2	Angola	Household	4187	1	1	1
3	Australia	Office Supplies	9389	1	1	1
4	Australia	Beverages	9389	2	1	1
5	Australia	Cereal	682	3	3	2
6	Austria	Cosmetics	2847	1	1	1
7	Azerbaijan	Cosmetics	7234	1	1	1
8	Azerbaijan	Office Supplies	2021	2	2	2
9	Bangladesh	Clothes	8263	1	1	1

#PRAKTIK WINDOWS FUNCTIONS

```
WITH
   raw_data AS
   (

   SELECT
       DISTINCT record.*,
       countries.country
   FROM
       'dummy_dataset.record' AS record
   LEFT JOIN
       'dummy_dataset.countries' AS countries
       ON
       record.country_id = countries.country_id
    )

  SELECT
        DISTINCT region,
        SUM(unit_solds) OVER(PARTITION BY region) AS total_unit_solds,
        MIN(unit_solds) OVER(PARTITION BY region) AS min_unit_solds,
        MAX(unit_solds) OVER(PARTITION BY region) AS max_unit_solds,
        AVG(unit_solds) OVER(PARTITION BY region) AS avg_unit_solds
  FROM
        raw_data
  WHERE
         order_date >= '2017-08-01' AND order_date <= '2017-08-31'
```

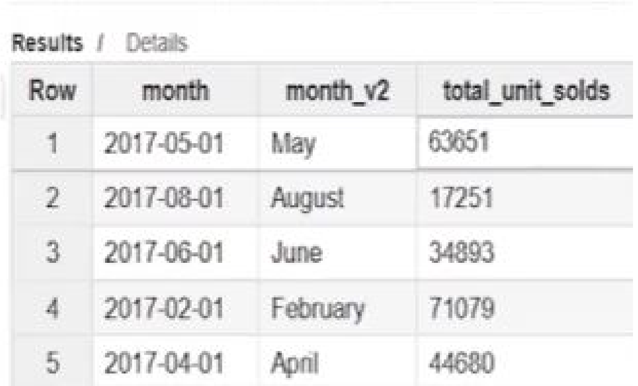


Results / Details

Row	region	total_unit_solds	min_unit_solds	max_unit_solds	avg_unit_solds
1	Central America and the Caribbean	2804	2804	2804	2804.0
2	Middle East and North Africa	673	673	673	673.0
3	Sub-Saharan Africa	13774	4168	9606	6887.0

#PRAKTIK FUNGSI LAG

```
SELECT
FROM
  (
   SELECT
     DATE_TRUNC(order_date, MONTH) AS month,
     FORMAT_TIMESSTAMP(order_date)) AS month_v2,
     SUM(unit_solds) As total_unit_sold
   FROM
      'dummy_dataset.record'
    GROUP BY
       month,
       month_v2
    )
)
```

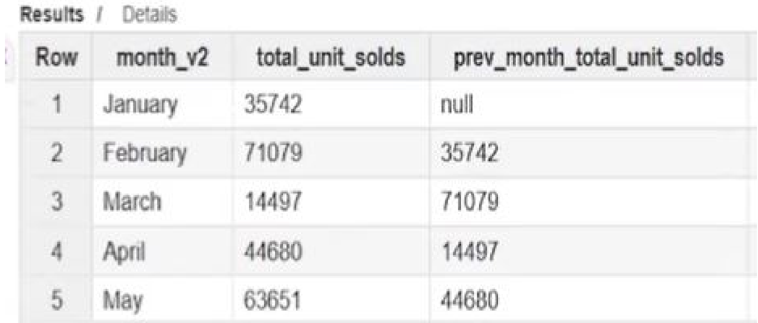


Results / Details

Row	month	month_v2	total_unit_solds
1	2017-05-01	May	63651
2	2017-08-01	August	17251
3	2017-06-01	June	34893
4	2017-02-01	February	71079
5	2017-04-01	April	44680

#PRAKTIK FUNGSI LAG

```
SELECT
FROM
  (
   SELECT
     month_v2,
     Total_unit_solds.
     LAG(total_unit_solds) OVER (ORDER BY month)
     AS prev_month_total_unit_solds
   FROM
    (
    SELECT
      DATE_TRUNC(order_date, MONTH) AS month,
      FORMAT_TIMESSTAMP(order_date)) AS month_v2,
      SUM(unit_solds) As total_unit_sold
    FROM
       'dummy_dataset.record'
    GROUP BY
      month,
      month_v2
    )
)
```

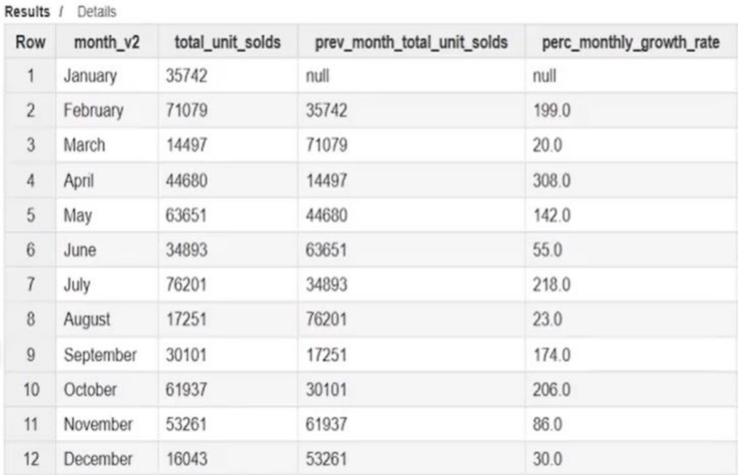


Results / Details

Row	month_v2	total_unit_solds	prev_month_total_unit_solds
1	January	35742	null
2	February	71079	35742
3	March	14497	71079
4	April	44680	14497
5	May	63651	44680

#PRAKTIK FUNGSI LAG

```
SELECT
  month_v2,
  total_unit_solds,
  prev_month_total_unit_solds,
  ROUND((total_unit_solds/prev_month_total_unit_solds)*100, 0)
  AS perc_total_monthly_growth_rate
FROM
  (
   SELECT
     month_v2,
     Total_unit_solds.
     LAG(total_unit_solds) OVER (ORDER BY month)
     AS prev_month_total_unit_solds
   FROM
    (
    SELECT
      DATE_TRUNC(order_date, MONTH) AS month,
      FORMAT_TIMESSTAMP(order_date)) AS month_v2,
      SUM(unit_solds) As total_unit_sold
    FROM
       'dummy_dataset.record'
    GROUP BY
      month,
      month_v2
    )
)
```

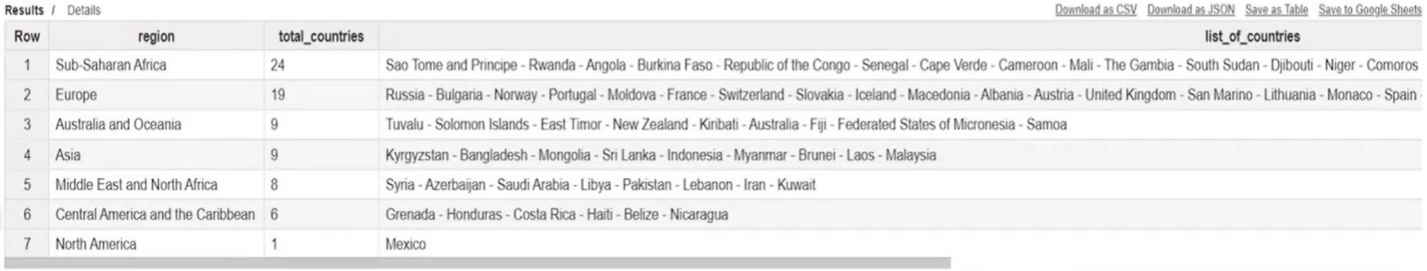


Results / Details

Row	month_v2	total_unit_solds	prev_month_total_unit_solds	perc_monthly_growth_rate
1	January	35742	null	null
2	February	71079	35742	199.0
3	March	14497	71079	20.0
4	April	44680	14497	308.0
5	May	63651	44680	142.0
6	June	34893	63651	55.0
7	July	76201	34893	218.0
8	August	17251	76201	23.0
9	September	30101	17251	174.0
10	October	61937	30101	206.0
11	November	53261	61937	86.0
12	December	16043	53261	30.0

#PRAKTIK FUNGSI AGREGASI

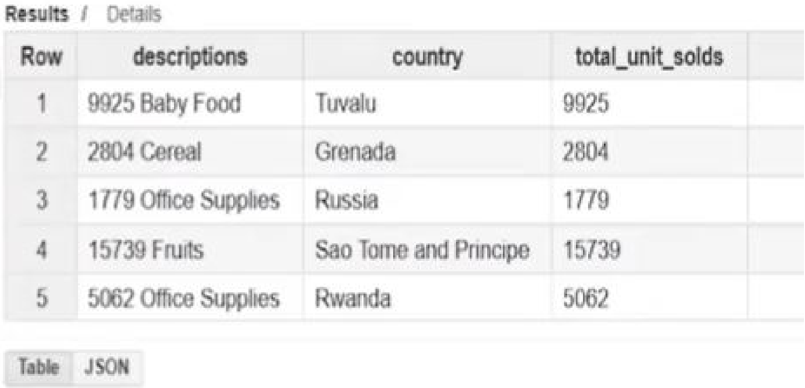
```
SELECT
    region,
    COUNT(DISTINCT country) AS total_countries,
    STRING_AGG(DISTINCT country, " - " ) AS lis_of_countries

FROM
        'dummy_dataset.countries'
     GROUP BY
         region
     ORDER BY
         total_countries DESC
```



Download as CSV Download as JSON Save as Table Save to Google Sheets

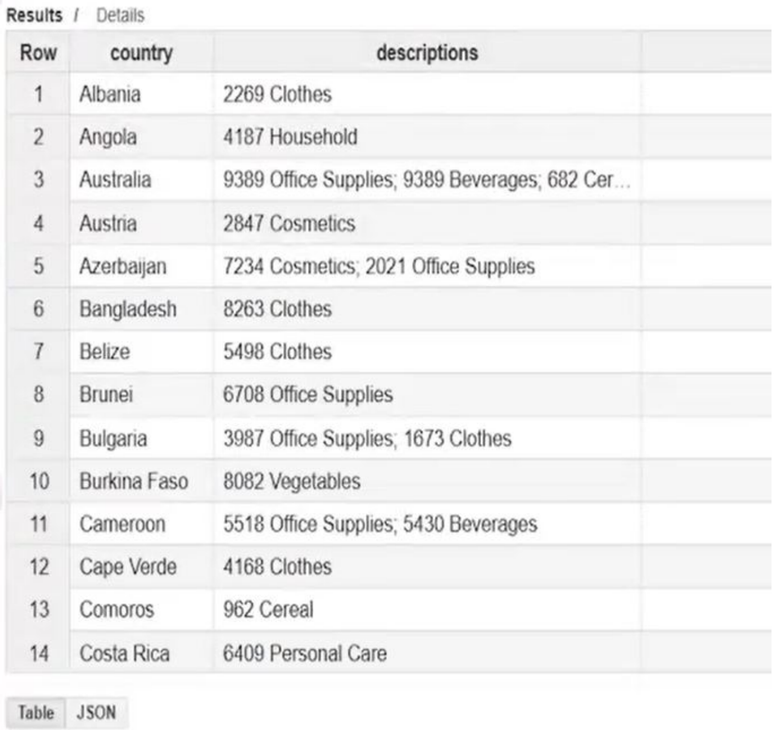
Row	region	total_countries	list_of_countries
1	Sub-Saharan Africa	24	Sao Tome and Principe - Rwanda - Angola - Burkina Faso - Republic of the Congo - Senegal - Cape Verde - Cameroon - Mali - The Gambia - South Sudan - Djibouti - Niger - Comoros
2	Europe	19	Russia - Bulgaria - Norway - Portugal - Moldova - France - Switzerland - Slovakia - Iceland - Macedonia - Albania - Austria - United Kingdom - San Marino - Lithuania - Monaco - Spain -
3	Australia and Oceania	9	Tuvalu - Solomon Islands - East Timor - New Zealand - Kiribati - Australia - Fiji - Federated States of Micronesia - Samoa
4	Asia	9	Kyrgyzstan - Bangladesh - Mongolia - Sri Lanka - Indonesia - Myanmar - Brunei - Laos - Malaysia
5	Middle East and North Africa	8	Syria - Azerbaijan - Saudi Arabia - Libya - Pakistan - Lebanon - Iran - Kuwait
6	Central America and the Caribbean	6	Grenada - Honduras - Costa Rica - Haiti - Belize - Nicaragua
7	North America	1	Mexico

```
WITH
  raw_data AS
  (
  SELECT
     CONCAT(CAST(SUM(unit_solds) AS STRING), " ", record.item.type) AS descriptions,
     DATE_TRUNC(order_date, MONTH) AS month,
     countries.region,
     countries.country,
     SUM(unit_solds) AS total_unit_solds
   FROM
     'dummy_dataset.record' AS record
   LEFT JOIN
       'dummy_dataset.countries' AS countries
    ON
        record.country_id = countries.country_id
    GROUP BY
         countries.region,
         countries.country,
         record.item_type
    )
```



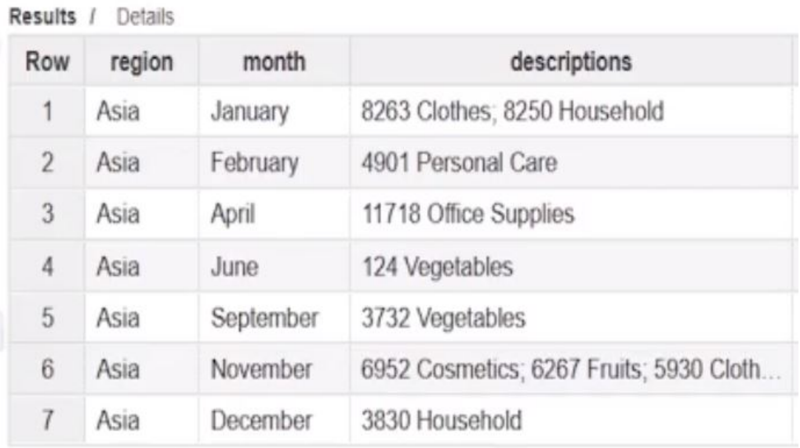
Row	descriptions	country	total_unit_solds
1	9925 Baby Food	Tuvalu	9925
2	2804 Cereal	Grenada	2804
3	1779 Office Supplies	Russia	1779
4	15739 Fruits	Sao Tome and Principe	15739
5	5062 Office Supplies	Rwanda	5062

```
  SELECT
     region,
     country,
     STRING_AGG(descriptions, ";  " ORDER BY total_unit_solds DESC) AS descriptions,
 FROM
     raw_data
  GROUP BY
        region,
        country,
    ORDER BY
   region,
   country
```



Row	country	descriptions
1	Albania	2269 Clothes
2	Angola	4187 Household
3	Australia	9389 Office Supplies; 9389 Beverages; 682 Cer...
4	Austria	2847 Cosmetics
5	Azerbaijan	7234 Cosmetics; 2021 Office Supplies
6	Bangladesh	8263 Clothes
7	Belize	5498 Clothes
8	Brunei	6708 Office Supplies
9	Bulgaria	3987 Office Supplies; 1673 Clothes
10	Burkina Faso	8082 Vegetables
11	Cameroon	5518 Office Supplies; 5430 Beverages
12	Cape Verde	4168 Clothes
13	Comoros	962 Cereal
14	Costa Rica	6409 Personal Care

```
 SELECT
   * EXCEPT (month_order)
  FROM
    (
    SELECT
        region,
        month_order,
        month
        STRING_AGG(descriptions, ";  " ORDER BY total_unit_solds DESC) AS descriptions,
    FROM
        raw_data
    WHERE
         region = 'Asia'
    GROUP BY
         region,
         month_order,
         month
      )
    ORDER BY
         region,
         month_order,
```

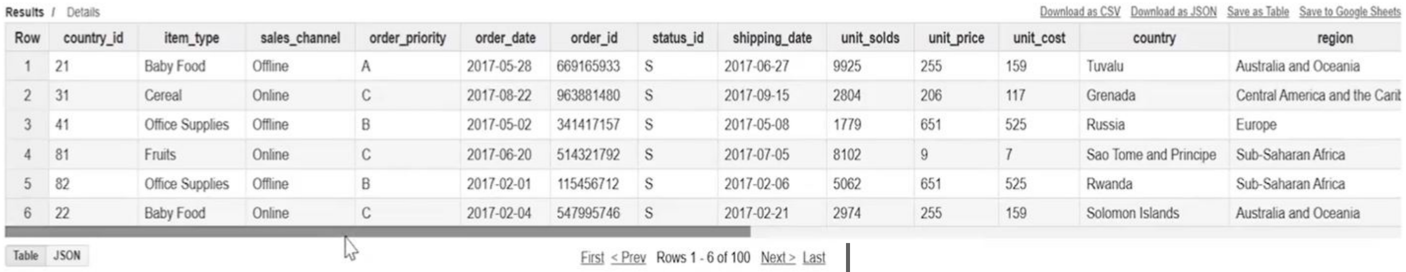


Results / Details

Row	region	month	descriptions
1	Asia	January	8263 Clothes; 8250 Household
2	Asia	February	4901 Personal Care
3	Asia	April	11718 Office Supplies
4	Asia	June	124 Vegetables
5	Asia	September	3732 Vegetables
6	Asia	November	6952 Cosmetics; 6267 Fruits; 5930 Cloth...
7	Asia	December	3830 Household

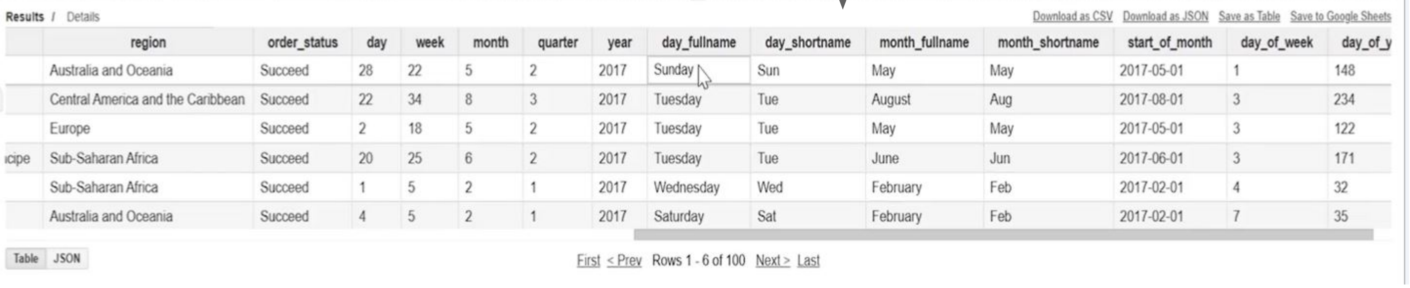
#PRAKTIK SQL DATE AND TIMESTAMP FUNCIONS

```
SELECT
    record.*,
    country,
    region,
    order_status,
    EXTRACT(DAY FROM order_date) AS day,
    EXTRACT(WEEK FROM order_date) AS week,
    EXTRACT(MONTH FROM order_date) AS month,
    EXTRACT(QUARTER FROM order_date) AS quarter,
    EXTRACT(YEAR FROM order_date) AS year,
    FORMAT_DATE("%A", order_date) AS day_fullname,
    FORMAT_DATE("%a", order_date) AS day_shortname,
    FORMAT_DATE("%B", order_date) AS month_fullname,
    FORMAT_DATE("%b", order_date) AS month_shortname
    DATE_TRUNC(order_date, MONTH) AS start_of_month,
    EXTRACT(DAYOFWEEK FROM order_date) AS day_of_week,
    CAST(FORMAT_DATE("%j", order_date) AS INT64) AS day_of_year
FROM
    'dummy_dataset.record' AS record
LEFT JOIN
    'dummy_dateset.countries' AS countries
 ON
     record.country_id = countries.country_id
LEFT JOIN
     'dummy_dataset.status' AS status
  ON
      record.status_id = status.status_id
```



Results / Details

Row	country_id	item_type	sales_channel	order_priority	order_date	order_id	status_id	shipping_date	unit_solds	unit_price	unit_cost	country	region
1	21	Baby Food	Offline	A	2017-05-28	669165933	S	2017-06-27	9925	255	159	Tuvalu	Australia and Oceania
2	31	Cereal	Online	C	2017-08-22	963881480	S	2017-09-15	2804	206	117	Grenada	Central America and the Carit
3	41	Office Supplies	Offline	B	2017-05-02	341417157	S	2017-05-08	1779	651	525	Russia	Europe
4	81	Fruits	Online	C	2017-06-20	514321792	S	2017-07-05	8102	9	7	Sao Tome and Principe	Sub-Saharan Africa
5	82	Office Supplies	Offline	B	2017-02-01	115456712	S	2017-02-06	5062	651	525	Rwanda	Sub-Saharan Africa
6	22	Baby Food	Online	C	2017-02-04	547995746	S	2017-02-21	2974	255	159	Solomon Islands	Australia and Oceania



Results / Details

	region	order_status	day	week	month	quarter	year	day_fullname	day_shortname	month_fullname	month_shortname	start_of_month	day_of_week	day_of_y
	Australia and Oceania	Succeed	28	22	5	2	2017	Sunday	Sun	May	May	2017-05-01	1	148
	Central America and the Caribbean	Succeed	22	34	8	3	2017	Tuesday	Tue	August	Aug	2017-08-01	3	234
	Europe	Succeed	2	18	5	2	2017	Tuesday	Tue	May	May	2017-05-01	3	122
ıcipe	Sub-Saharan Africa	Succeed	20	25	6	2	2017	Tuesday	Tue	June	Jun	2017-06-01	3	171
	Sub-Saharan Africa	Succeed	1	5	2	1	2017	Wednesday	Wed	February	Feb	2017-02-01	4	32
	Australia and Oceania	Succeed	4	5	2	1	2017	Saturday	Sat	February	Feb	2017-02-01	7	35

#PRAKTIK SQL DATE AND TIMESTAMP FUNCIONS
# MENAMBAHKAN FUNGSI WHERE AGAR DIMULAI 30 HARI DARI TANGGAL 1

```
SELECT *
   FROM
   (
   record.*,
   country,
   region,
   order_status,
   EXTRACT(DAY FROM order_date) AS day,
   EXTRACT(WEEK FROM order_date) AS week,
   EXTRACT(MONTH FROM order_date) AS month,
   EXTRACT(QUARTER FROM order_date) AS quarter,
   EXTRACT(YEAR FROM order_date) AS year,
   FORMAT_DATE("%A", order_date) AS day_fullname,
   FORMAT_DATE("%a", order_date) AS day_shortname,
   FORMAT_DATE("%B", order_date) AS month_fullname,
   FORMAT_DATE("%b", order_date) AS month_shortname
   DATE_TRUNC(order_date, MONTH) AS start_of_month,
   EXTRACT(DAYOFWEEK FROM order_date) AS day_of_week,
   CAST(FORMAT_DATE("%j", order_date) AS INT64) AS day_of_year
FROM
   'dummy_dataset.record' AS record
LEFT JOIN
   'dummy_dateset.countries' AS countries
 ON
    record.country_id = countries.country_id
LEFT JOIN
    'dummy_dataset.status' AS status
  ON
     record.status_id = status.status_id
  )
WHERE
    order_date >= DATE_SUB(DATE("2017-01-01"), INTERVAL 30 DAY)
```

#PRAKTIK SQL DATE AND TIMESTAMP FUNCIONS
# MENAMBAHKAN WINDOWS FUNCTION UNTUK MENCARI PROFIT DI MASING-MASING REGION PERHARI

```
WITH
    raw_data AS
   (
 SELECT
   record.*,
   country,
   region,
   order_status,
   EXTRACT(DAY FROM order_date) AS day,
   EXTRACT(WEEK FROM order_date) AS week,
   EXTRACT(MONTH FROM order_date) AS month,
   EXTRACT(QUARTER FROM order_date) AS quarter,
   EXTRACT(YEAR FROM order_date) AS year,
   FORMAT_DATE("%A", order_date) AS day_fullname,
   FORMAT_DATE("%a", order_date) AS day_shortname,
   FORMAT_DATE("%B", order_date) AS month_fullname,
   FORMAT_DATE("%b", order_date) AS month_shortname
   DATE_TRUNC(order_date, MONTH) AS start_of_month,
   EXTRACT(DAYOFWEEK FROM order_date) AS day_of_week,
   CAST(FORMAT_DATE("%j", order_date) AS INT64) AS day_of_year,
   (unit_solds*unit_price) AS total_gross_revenue,
   (unit_sold * unit_cost) AS total_cost
FROM
   'dummy_dataset.record' AS record
LEFT JOIN
   'dummy_dateset.countries' AS countries
 ON
    record.country_id = countries.country_id
LEFT JOIN
    'dummy_dataset.status' AS status
 ON
     record.status_id = status.status_id
  ),

add_profit_column As
  (
  SELECT
     DISTINCT
     order_id,
     order_date,
     region,
     (total_gross_revenue - totsl_cost) AS total_profit
  FROM
     raw_data
      ),
```

#PRAKTIK SQL DATE AND TIMESTAMP FUNCIONS
# MENAMBAHKAN WINDOWS FUNCTION UNTUK MENCARI PROFIT DI MASING-MASING REGION PERHARI

```
#LANJUTAN SUBQUERY SEBELUMNYA
each_item_type_to_column AS
   (
    SELECT
      order_date,
      SUM(total_profit) AS aus_ and_oce,
      NULL AS cen_am_and_car,
      NULL AS euro,
      NULL AS sub_sah_afr,
      NULL AS asia,
      NULL AS mid_east_and_north_afr,
      NULL AS north_am
  FROM
       add_profit_clumn
  WHERE
      region = 'Australia and Oceania'
   GROUP BY
      order_date
```

#MENGGUNAKAN FUNGSI UNION UNTUK MENGGABUNGKAN 2 SUBQUERY TERPISAH

```
  UNION ALL

  SELECT
      order_date,
      NULL AS aus_ and_oce,
      SUM(total_profit) AS cen_am_and_car,
      NULL AS euro,
      NULL AS sub_sah_afr,
      NULL AS asia,
      NULL AS mid_east_and_north_afr,
      NULL AS north_am
  FROM
       add_profit_clumn
    WHERE
      region = 'Central America and the Caribbean'
   GROUP BY
      order_date

 UNION ALL

  SELECT
      order_date,
      NULL AS aus_ and_oce,
      NULL AS cen_am_and_car,
      SUM(total_profit) AS euro,
      NULL AS sub_sah_afr,
      NULL AS asia,
      NULL AS mid_east_and_north_afr,
      NULL AS north_am
```

#PRAKTIK SQL DATE AND TIMESTAMP FUNCIONS
# MENAMBAHKAN WINDOWS FUNCTION UNTUK MENCARI PROFIT DI MASING-MASING REGION PERHARI

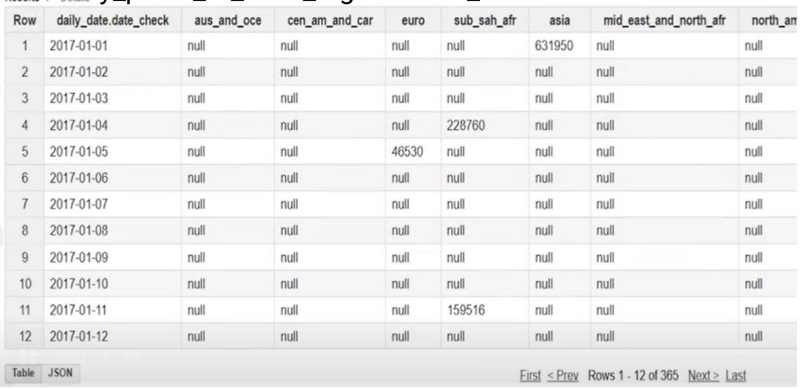
```
#LANJUTAN SUBQUERY SEBELUMNYA
  FROM
       add_profit_clumn
   WHERE
      region = 'Europe
   GROUP BY
      order_date

  UNION ALL

  SELECT
      order_date,
      NULL AS aus_ and_oce,
      NULL AS cen_am_and_car,
      NULL AS euro,
      SUM(total_profit) AS sub_sah_afr,
      NULL AS asia,
      NULL AS mid_east_and_north_afr,
      NULL AS north_am
  FROM
       add_profit_clumn
   WHERE
      region = ' Sub-Saharan Africa'
   GROUP BY
      order_date

  UNION ALL

  SELECT
      order_date,
      NULL AS aus_ and_oce,
      NULL AS cen_am_and_car,
      NULL AS euro,
      NULL AS sub_sah_afr,
      SUM(total_profit) AS asia,
      NULL AS mid_east_and_north_afr,
      NULL AS north_am
  FROM
       add_profit_clumn
   WHERE
      region = ' Asia'
   GROUP BY
      order_date
```

#PRAKTIK SQL DATE AND TIMESTAMP FUNCIONS
# MENAMBAHKAN WINDOWS FUNCTION UNTUK MENCARI PROFIT DI MASING-MASING REGION PERHARI

```
#LANJUTAN SUBQUERY SEBELUMNYA
  FROM
       add_profit_clumn
   WHERE
      region = 'Europe
   GROUP BY
      order_date

  UNION ALL

  SELECT
      order_date,
      NULL AS aus_ and_oce,
      NULL AS cen_am_and_car,
      NULL AS euro,
      NULL AS sub_sah_afr,
      NULL AS asia,
       SUM(total_profit) AS mid_east_and_north_afr,
      NULL AS north_am
  FROM
       add_profit_clumn
   WHERE
      region = 'Middle East and North Africa'
   GROUP BY
      order_date

  UNION ALL

  SELECT
      order_date,
      NULL AS aus_ and_oce,
      NULL AS cen_am_and_car,
      NULL AS euro,
      NULL AS sub_sah_afr,
      NUL AS asia,
      NULL AS mid_east_and_north_afr,
      SUM(total_profit )AS north_am
  FROM
       add_profit_clumn
   WHERE
      region = 'North America'
   GROUP BY
      order_date
  ),
```

#PRAKTIK SQL DATE AND TIMESTAMP FUNCIONS
# MENAMBAHKAN WINDOWS FUNCTION UNTUK MENCARI PROFIT DI MASING-MASING REGION PERHARI

```
#LANJUTAN SUBQUERY SEBELUMNYA
 daily_profit_of_each_region AS
 (
 SELECT
      order_date,
      SUM(aus_and_oce) AS aus_and_oce,
      SUM(cen_am_and_car) AS cen_am_and_car
      SUM(euro) AS euro,
      SUM(sub_sah_afrr) AS sub_sah_afr,
      SUM(asia) AS asia,
      SUM(mid_east_and_north_afr) AS mid_east_and_north_afr,
      SUM(north_am) AS north_am
  FROM
      each_item_type_to_column
  GROUP BY
      order_date
   ),

date_for_check_AS
  (
  SELECT
       date_check
  FROM
       UNNEST(GENERATE_DATE_ARRAY("2017-01-01",
        "2017-12-31") AS date_check
   )

SELECT
       date_for_check AS daily_date,
       daily_profit_of_each_region.* EXCEPT(order_date)
 FROM
        date_for_check
  LEFT JOIN
        daily_profit_of_each_region
  ON
         date_for_check.date_check
         =
         daily_profit_of_each_region.order_date
```

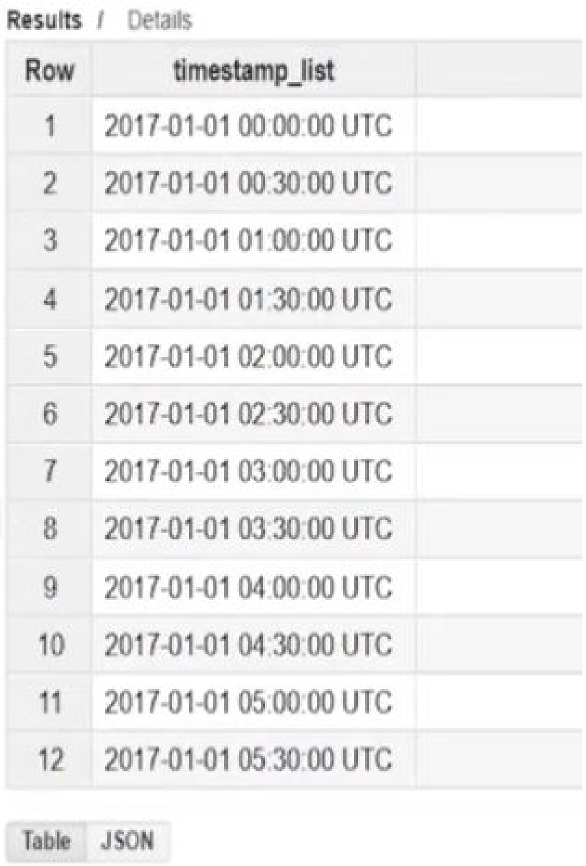


Results / Details

Row	daily_date.date_check	aus_and_oce	cen_am_and_car	euro	sub_sah_afr	asia	mid_east_and_north_afr	north_am
1	2017-01-01	null	null	null	null	631950	null	null
2	2017-01-02	null	null	null	null	null	null	null
3	2017-01-03	null	null	null	null	null	null	null
4	2017-01-04	null	null	null	228760	null	null	null
5	2017-01-05	null	null	46530	null	null	null	null
6	2017-01-06	null	null	null	null	null	null	null
7	2017-01-07	null	null	null	null	null	null	null
8	2017-01-08	null	null	null	null	null	null	null
9	2017-01-09	null	null	null	null	null	null	null
10	2017-01-10	null	null	null	null	null	null	null
11	2017-01-11	null	null	null	159516	null	null	null
12	2017-01-12	null	null	null	null	null	null	null

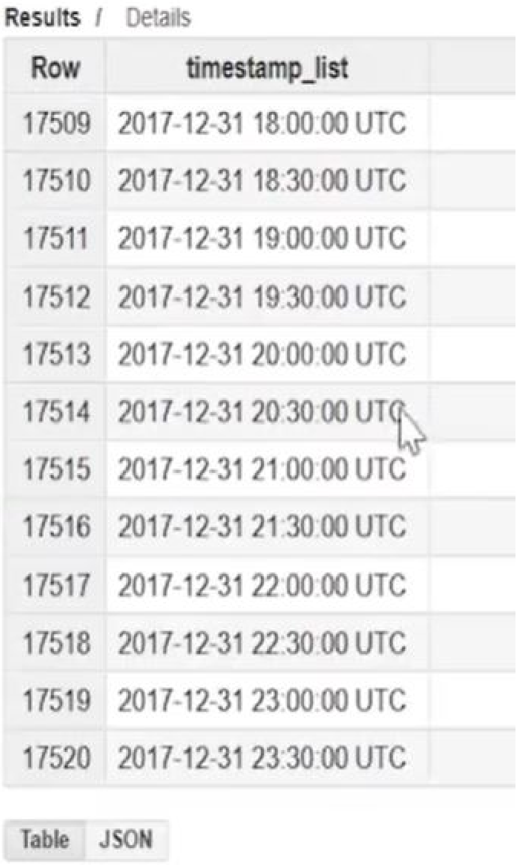
#PRAKTIK GENERATE TIMESTAMP ARRAY

```
SELECT
    timestamp_list
FROM
     UNNEST(GENERATE_DATE_ARRAY("2017-01-01",
      "2017-12-31 23:59:59", INTERVAL 30 MINUTE) AS timestamp_list
```



Results / Details

Row	timestamp_list
1	2017-01-01 00:00:00 UTC
2	2017-01-01 00:30:00 UTC
3	2017-01-01 01:00:00 UTC
4	2017-01-01 01:30:00 UTC
5	2017-01-01 02:00:00 UTC
6	2017-01-01 02:30:00 UTC
7	2017-01-01 03:00:00 UTC
8	2017-01-01 03:30:00 UTC
9	2017-01-01 04:00:00 UTC
10	2017-01-01 04:30:00 UTC
11	2017-01-01 05:00:00 UTC
12	2017-01-01 05:30:00 UTC



Results / Details

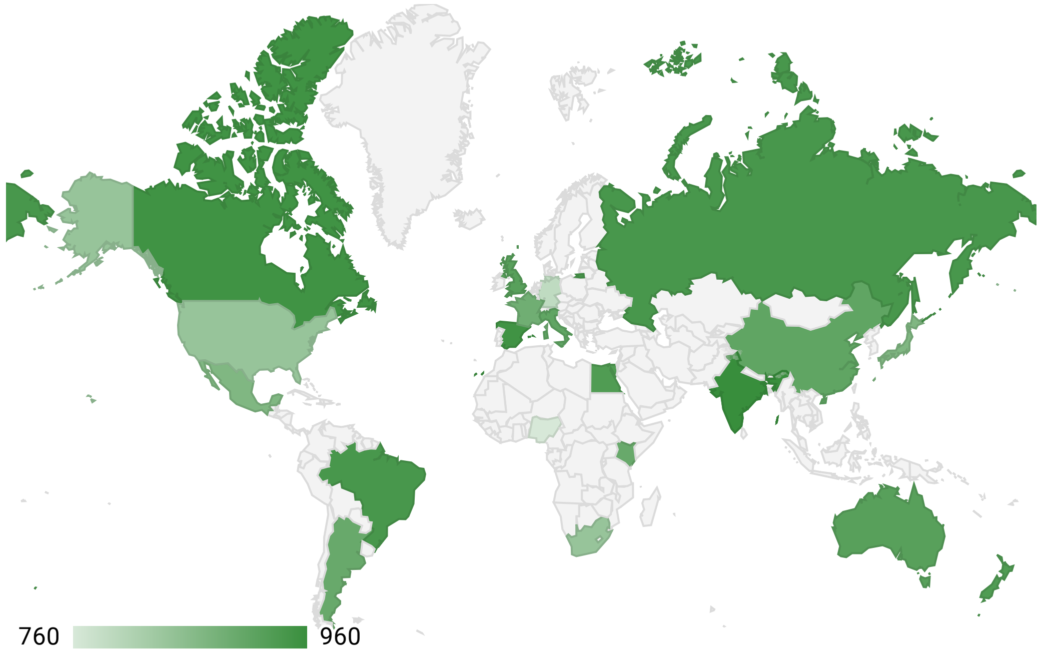
Row	timestamp_list
17509	2017-12-31 18:00:00 UTC
17510	2017-12-31 18:30:00 UTC
17511	2017-12-31 19:00:00 UTC
17512	2017-12-31 19:30:00 UTC
17513	2017-12-31 20:00:00 UTC
17514	2017-12-31 20:30:00 UTC
17515	2017-12-31 21:00:00 UTC
17516	2017-12-31 21:30:00 UTC
17517	2017-12-31 22:00:00 UTC
17518	2017-12-31 22:30:00 UTC
17519	2017-12-31 23:00:00 UTC
17520	2017-12-31 23:30:00 UTC

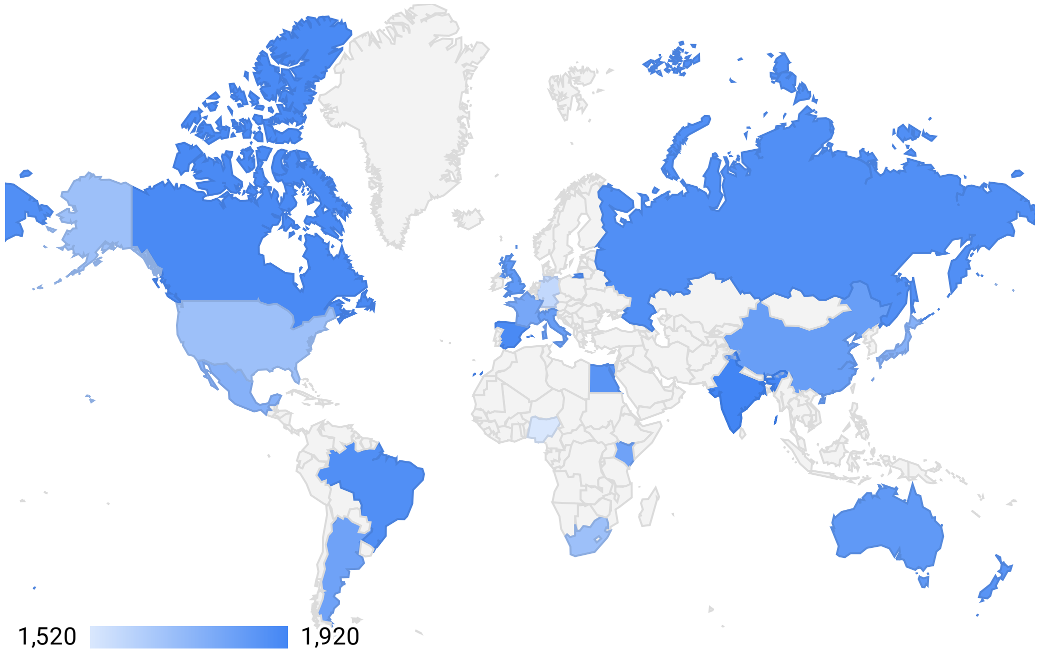
DATA VISUALIZATION WITH LOOKER STUDIO FOR SALES ANALYSIS

Created By: Muhammad Pajrul Palah

country ▾

Geo Chart Between Unit Solds VS Profit Per Each Country





Data Tables of Sales Analysis

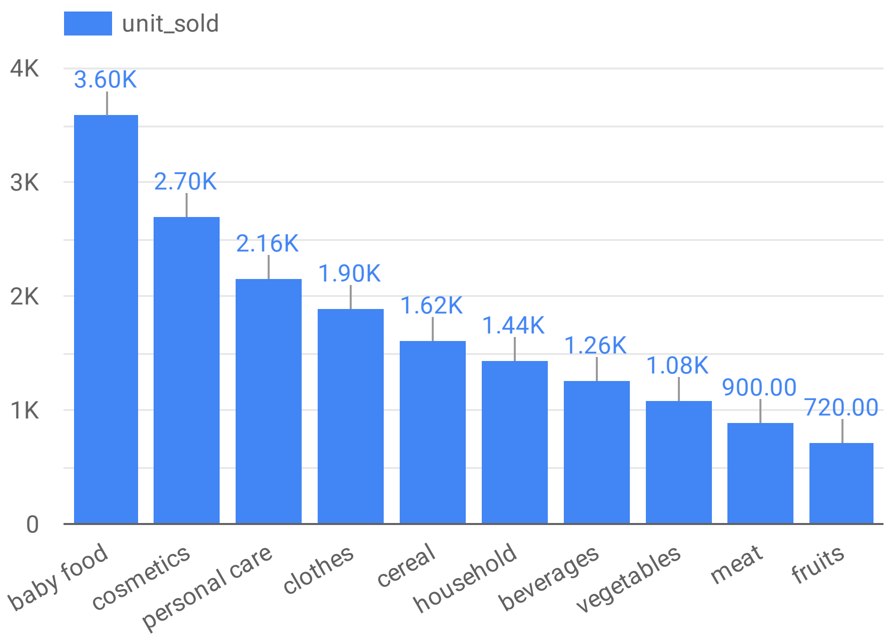
	Country Name ❶ ▴	Item Type	Unit Price	Unit Solds ❷ ▾
1.	Argentina	baby food	12	200
2.	Argentina	cosmetics	30	150
3.	Argentina	personal care	18	120
4.	Argentina	clothes	25	100
5.	Argentina	household	40	80
6.	Argentina	beverages	20	70
7.	Argentina	vegetables	10	60
8.	Argentina	meat	50	50
9.	Argentina	fruits	35	40
10.	Argentina	snack	8	30

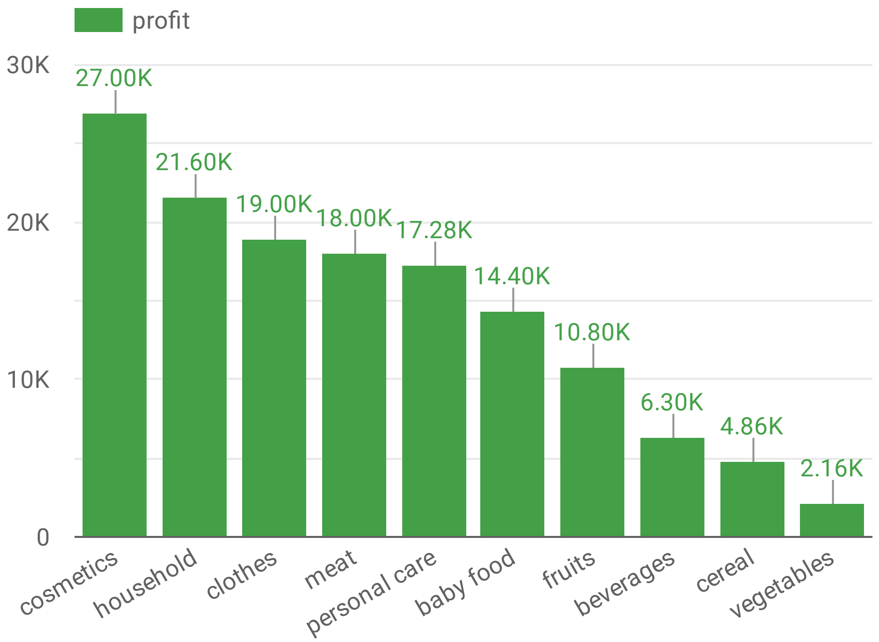
1 - 10 / 199 ‹ ›

	Country Name	Total Revenue ▾	Total Cost	Profit
1.	India	9.60K	7.68K	$1.92K
2.	Nigeria	7.60K	6.08K	$1.52K
3.	Spain	7.60K	5.70K	$1.90K
4.	Canada	7.60K	5.70K	$1.90K
5.	Brazil	7.52K	5.64K	$1.88K
6.	Russia	7.52K	5.64K	$1.88K
7.	New Zealand	7.44K	5.58K	$1.86K
8.	Egypt	7.44K	5.58K	$1.86K
9.	United Kingdom	7.36K	5.52K	$1.84K
10.	Australia	7.36K	5.52K	$1.84K

1 - 10 / 20 ‹ ›

Bar Chart Unit Solds VS Profit of Each Item

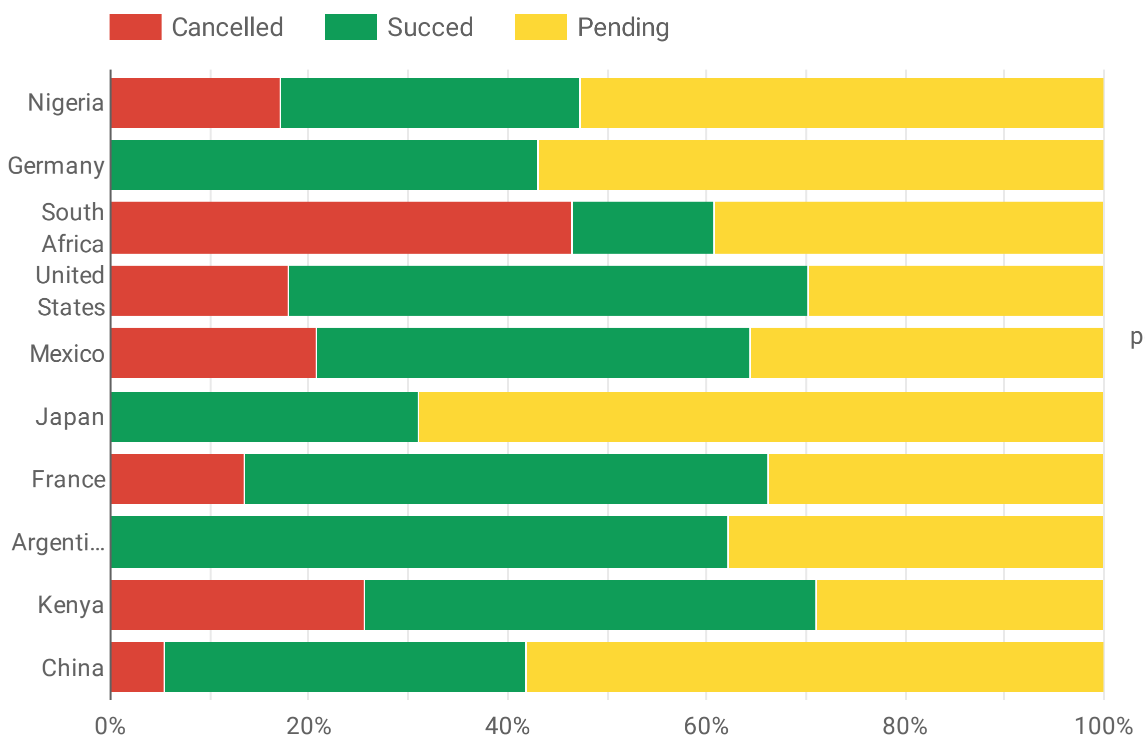


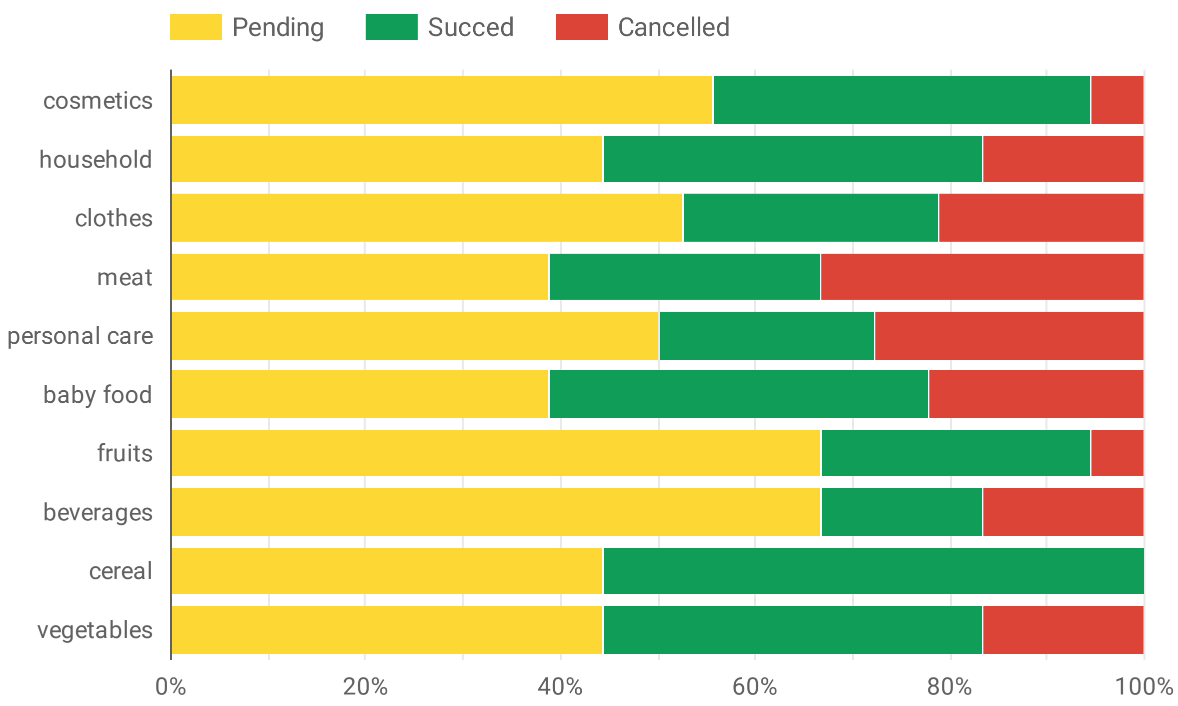


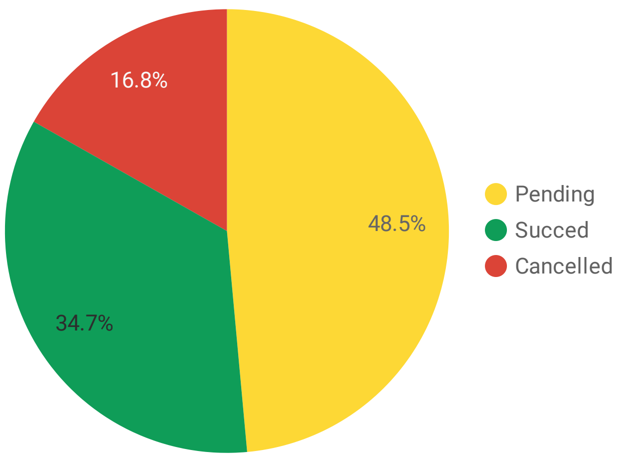
Pivot Table of Product Profit in Each Countries

										country / profit
item_type	India	Canada	Spain	Brazil	Russia	Egypt	New Zealand	Australia	United King...	Italy
cosmetics	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500	1,500
household	1,200	1,200	1,200	1,200	1,200	1,200	1,200	1,200	1,200	-
clothes	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
meat	1,000	1,000	1,000	-	-	1,000	1,000	1,000	1,000	1,000
personal care	960	960	960	960	960	960	960	960	960	960
baby food	800	800	800	800	800	800	800	800	800	800
fruits	600	-	-	600	600	600	600	600	600	600
beverages	350	350	350	350	350	350	350	-	-	350
cereal	270	270	270	270	270	270	270	270	270	270
vegetables	120	120	120	120	120	-	-	120	120	120
snack	-	60	60	60	60	60	60	60	60	60

Order Status Analysis







Total Product Types Cancelled in Each Countries

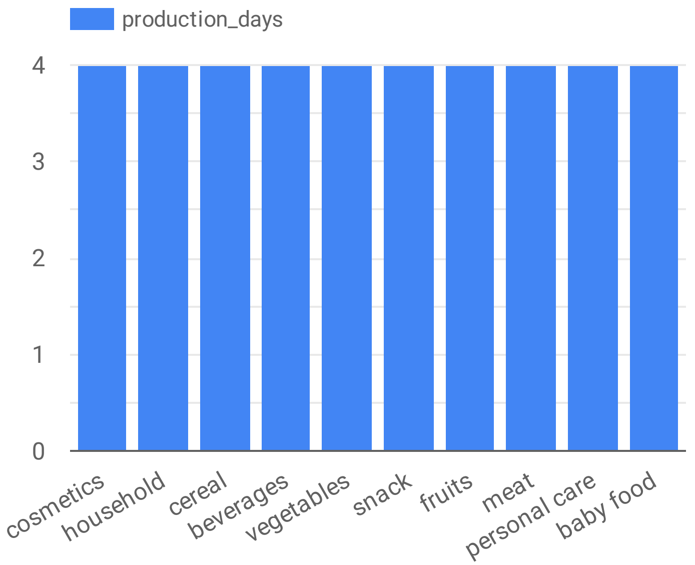
								item_type / unit_sold
country	baby food	personal...	clothes	meat	household	beverages	vegetabl...	cosmet
South Africa	200	-	100	50	-	-	-	
Italy	200	120	-	50	-	-	-	
Egypt	-	-	100	50	80	70	-	
Russia	-	120	-	-	-	70	60	
Kenya	200	-	-	-	-	-	-	
Spain	200	-	-	-	-	-	-	
New Zealand	-	-	-	50	-	-	-	1
Mexico	-	-	100	-	80	-	-	
United States	-	120	-	-	-	-	-	
India	-	-	100	50	-	-	-	

Production Days Analysis

	order_id	item_type	country	production_days...
1.	1003	cosmetics	Italy	4
2.	1005	household	Australia	4
3.	1006	cereal	Brazil	4
4.	1007	beverages	Egypt	4
5.	1009	vegetables	India	4
6.	1010	fruits	France	4
7.	1011	snack	Germany	4
8.	1012	clothes	South Afri...	4

1 - 100 / 199

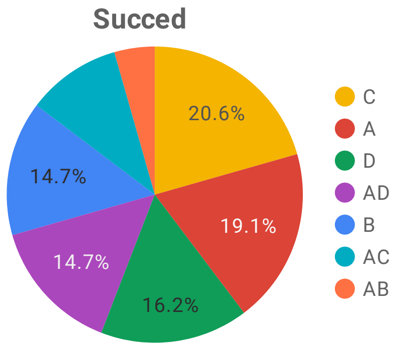
	item_type / production_days					
country	clothes	baby food	cosmetics	personal ca...	household	cereal
Japan	4	4	4	-	4	4
Italy	4	4	4	4	-	4
Kenya	4	4	4	4	4	-
Australia	4	4	4	4	4	4
Brazil	4	4	4	4	4	4
Egypt	4	4	4	4	4	4
Canada	4	4	4	4	4	4
India	4	4	4	4	4	4

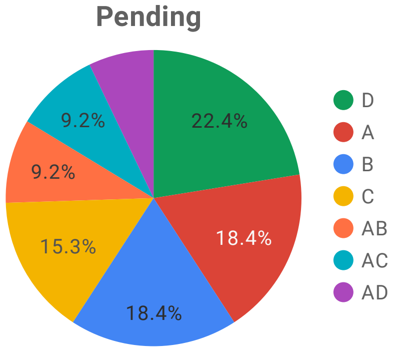


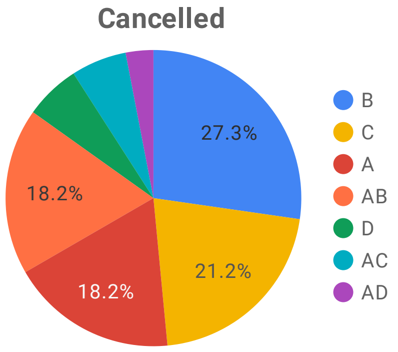
	country	order_priori...	item_type	order_status	production_days
1.	Italy	C	cosmetics	Succed	4
2.	Australia	AB	household	Cancelled	4
3.	Brazil	AC	cereal	Pending	4
4.	Egypt	AD	beverages	Cancelled	4
5.	India	B	vegetables	Succed	4
6.	France	C	fruits	Succed	4
7.	Germany	D	snack	Pending	4
8.	South Africa	A	clothes	Cancelled	4

1 - 100 / 199

Order Prioriity to Order Status (Succed, Pending and Cancelled)


Succed
● C
● A
● D
● AD
● B
● AC
● AB


Pending
● D
● A
● B
● C
● AB
● AC
● AD


Cancelled
● B
● C
● A
● AB
● D
● AC
● AD