Task 3

The dataset used here is bankmarketing.csv

1. SELECT

To select specific columns, like age, job, and y:

```sql
SELECT age, job, y
FROM `imposing-vista-456007-p4.bankmarketing.bank` LIMIT 10
```

```
age   job          y
  56  housemaid    FALSE
  57  services     FALSE
  37  services     FALSE
  40  admin.       FALSE
  56  services     FALSE
  45  services     FALSE
  59  admin.       FALSE
  41  blue-collar  FALSE
  24  technician   FALSE
  25  services     FALSE
```

2.WHERE

```sql
SELECT age, job, y
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE age > 40
LIMIT 20;
```

```
age   job          y

56    housemaid    FALSE

57    services     FALSE

56    services     FALSE

45    services     FALSE

59    admin.       FALSE

41    blue-collar  FALSE

41    blue-collar  FALSE

57    housemaid    FALSE

54    retired      FALSE

46    blue-collar  FALSE
```

50	blue-collar	FALSE
55	blue-collar	FALSE
55	retired	FALSE
41	technician	FALSE
59	technician	FALSE
54	technician	FALSE
55	unknown	FALSE
46	admin.	FALSE
59	technician	FALSE

3. **ORDER BY**

To sort results — e.g., oldest to youngest:

```
SELECT age, job, y
FROM `imposing-vista-456007-p4.bankmarketing.bank`
ORDER BY age DESC
LIMIT 10;
```

Row	age	job	y
1	98	retired	true
2	98	retired	true
3	95	retired	false
4	94	retired	false
5	92	retired	false
6	92	retired	true
7	92	retired	true
8	92	retired	true
9	91	retired	false
10	91	retired	false

4. **GROUP BY**

To group by a category in terms of job.

```sql
SELECT job, COUNT(*) AS total_yes
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE job = 'services'
GROUP BY job
ORDER BY total_yes DESC;
```

```
Row     job     total_yes

1       services3969
```

```sql
SELECT job, COUNT(*) AS total_yes
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE job = 'retired'
GROUP BY job
ORDER BY total_yes DESC;
```

```
Row     job     total_yes

1       retired 1720
```

```sql
SELECT *
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE job = 'retired'
LIMIT 10;
```

```
age     job     marital education       default housingloan     contact month   day_of_week
        duration        campaign        pdays   previous        poutcome        emp_var_rate
        cons_price_idx cons_conf_idx euribor3m      nr_employed

1       54      retired marriedbasic.9yunknown  yes     yes     telephone       may
        mon     174     1       999     0       nonexistent     1.1     93.994  -36.4   4.857
        5191.0

2       55      retired single  high.school     no      yes     no      telephone       may
        mon     342     1       999     0       nonexistent     1.1     93.994  -36.4   4.857
        5191.0

3       60      retired divorced        university.degree       unknown no      no
        telephone       may     mon     514     1       999     0       nonexistent     1.1
        93.994  -36.4   4.857   5191.0

4       56      retired marriedbasic.4yno       yes     no      telephone       may     mon     102
        2       999     0       nonexistent     1.1     93.994  -36.4   4.857   5191.0

5       54      retired marriedhigh.school      unknown no      no      telephone       may
        mon     130     1       999     0       nonexistent     1.1     93.994  -36.4   4.857
        5191.0
```

```
6       57      retired  marriedunknown       unknown      no      no      telephone       may
        mon     611      2        999     0       nonexistent  1.1     93.994  -36.4   4.857
        5191.0

7       58      retired  marrieduniversity.degree     no       no      no      telephone       may
        mon     132      1        999     0       nonexistent  1.1     93.994  -36.4   4.857
        5191.0

8       43      retired  marriedbasic.4yunknown       no       no      telephone       may
        mon     410      3        999     0       nonexistent  1.1     93.994  -36.4   4.857
        5191.0

9       57      retired  marriedhigh.school   no      no       no      telephone       may
        mon     238      2        999     0       nonexistent  1.1     93.994  -36.4   4.857
        5191.0

10      51      retired  marriedprofessional.course   no       no      no      telephone       may
        mon     118      3        999     0       nonexistent  1.1     93.994  -36.4   4.857
        5191.0
```

1. **SELECT + WHERE**

all clients who are married and have a housing loan.

```sql
SELECT age, job, marital, housing
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE marital = 'married' AND housing = 'yes'
LIMIT 20;
```

age	job	marital	housing
24	blue-collar	married	yes
25	blue-collar	married	yes
25	admin.	married	yes
24	blue-collar	married	yes
25	blue-collar	married	yes
25	services	married	yes
23	blue-collar	married	yes
24	admin.	married	yes
25	admin.	married	yes
25	services	married	yes
24	blue-collar	married	yes
25	blue-collar	married	yes
25	housemaid	married	yes
23	services	married	yes
25	self-employed	married	yes
25	blue-collar	married	yes
24	services	married	yes

25	blue-collar	married	yes
24	admin.	married	yes

2. **SELECT + ORDER BY**

List the top 10 oldest clients:

```
SELECT age, job, education
FROM `imposing-vista-456007-p4.bankmarketing.bank`
ORDER BY age DESC
LIMIT 10;
```

Row	age	job	education
1	98	retired	basic.4y
2	98	retired	basic.4y
3	95	retired	basic.6y
4	94	retired	basic.9y
5	92	retired	unknown
6	92	retired	unknown
7	92	retired	unknown
8	92	retired	unknown
9	91	retired	university.degree
10	91	retired	university.degree

```
SELECT marital, COUNT(*) AS total_clients
FROM `imposing-vista-456007-p4.bankmarketing.bank`
GROUP BY marital;
```

Row	marital	total_clients
1	married	24928
2	single	11568
3	divorced	4612
4	unknown	80

```
SELECT age, job, education
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE age < 30 AND job = 'student'
ORDER BY age
```

```
limit 20;
```

```
Row     age     job             education

1       17      student         basic.9y

2       17      student         basic.9y

3       17      student         basic.9y

4       17      student         unknown

5       17      student         unknown

6       18      student         unknown

7       18      student         unknown

8       18      student         unknown

9       18      student         basic.6y

10      18      student high.school

11      18      student unknown

12      18      student unknown
```

```
SELECT education, AVG(age) AS avg_age
FROM `imposing-vista-456007-p4.bankmarketing.bank`
GROUP BY education
ORDER BY avg_age DESC;

Row     education       avg_age
1       illiterate      48.499999999999993
2       basic.4y        47.596503831417536
3       unknown 43.481224725592185
4       basic.6y        40.448952879581235
5       professional.course     40.080106809078835
6       basic.9y        39.061207609594724
7       university.degree       38.879191321498908
8       high.school     37.998213347346251

SELECT job, COUNT(*) AS yes_count
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE job = 'retired'
GROUP BY job
HAVING COUNT(*) > 100
ORDER BY yes_count DESC;
Row     job     yes_count
1       retired 1720

SELECT age, job, marital, y
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE age BETWEEN 30 AND 40
limit 20;
```

```
Row     age     job     marital           y
1       37      services       married    false
2       40      admin.         married    false
3       35      blue-collar    married    false
4       35      blue-collar    married    false
5       39      management     single     false
6       30      unemployed     married        false
7       37      admin.         Married        false
8       35      technician     married        false
9       39      self-employed  married        false
10      34      services       married        false
11      32      entrepreneur   married        false
12      38      admin.         Single         false
13      40      blue-collar    married        false
14      35      admin.         Married        false
15      39      housemaid      married        false
16      37      admin.         married         false
17      33      admin.         married        false
18      37      admin.         married        false
19      33      services       married        false
20      38      admin.         married         false
```

```
SELECT education, AVG(campaign) AS avg_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`
GROUP BY education
ORDER BY avg_campaign DESC
LIMIT 5;
```

```
Row     education            avg_campaign
1       basic.4y              2.6005747126436773
2       unknown               2.5961871750433283
3       professional.course   2.5861148197596906
4       high.school           2.5685759327377857
5       university.degree     2.5635272846811312
```

```
SELECT job, COUNT(*) AS married_yes
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE job = 'blue-collar' AND marital = 'married'
GROUP BY job
HAVING married_yes > 50
ORDER BY married_yes DESC;
```

```
Row     job        married_ yes
1       blue-collar    6687
```

```
SELECT job, COUNT(*) AS married_yes
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE housing = 'no' AND marital = 'married'
GROUP BY job
HAVING married_yes > 50
ORDER BY married_yes DESC;
```

```
Row     job        married_yes
1       blue-collar    3119
2       admin.         2336
3       technician     1663
4       services       1062
5       management     970
```

```
6       retired       601
7       entrepreneur   458
8       self-employed  411
9       housemaid      364
10      unemployed     267
```

```sql
SELECT job, COUNT(*) AS married_yes
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE housing = 'no' AND contact = 'cellular'
GROUP BY job
HAVING married_yes > 50
ORDER BY married_yes DESC;
```

```
Row     job        married_yes
1       admin.        2996
2       blue-collar    2170
3       technician     1925
4       services       981
5       management     849
6       retired       543
7       self-employed  390
8       entrepreneur   344
9       student       282
10      housemaid      276
```

```sql
SELECT age, job, campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE campaign > (
    SELECT AVG(campaign)
    FROM `imposing-vista-456007-p4.bankmarketing.bank`
)
LIMIT 10;
```

```
Row     age     job          campaign
1       49      blue-collar    3
2       42      blue-collar    3
3       43      services       3
4       36      admin.        3
5       40      services       3
6       43      retired        3
7       43      management     3
8       36      technician     3
9       53      services       3
10      43      admin.        3
```

```sql
SELECT *
FROM (
    SELECT job, AVG(campaign) AS avg_campaign
    FROM `imposing-vista-456007-p4.bankmarketing.bank`
    GROUP BY job
) AS job_balances
WHERE avg_campaign > 2.6005747126436773;
```

```
Row     job            avg_campaign
1       housemaid      2.6396226415094359
2       admin.        2.6234887737478316
3       self-employed  2.660802251935253
4       unknown       2.6484848484848476
```

```
SELECT age, education, campaign
FROM imposing-vista-456007-p4.bankmarketing.bank AS b1
WHERE campaign > (
    SELECT AVG(campaign)
    FROM imposing-vista-456007-p4.bankmarketing.bank AS b2
    WHERE b1.education = b2.education
)
LIMIT 20;
Row     age     education       campaign
1       49      basic.4y               3
2       42      basic.9y               3
3       43      high.school            3
4       36      university.degree      3
5       40      high.school            3
6       43      basic.4y               3
7       43      university.degree      3
8       36      professional.course    3
9       53      high.school            3
10      43      basic.9y               3
11      40      basic.4y               3
12      47      basic.9y               4
13      51      professional.course    3
14      56      basic.4y               3
15      53      basic.9y               3
16      39      basic.9y               3
17      35      high.school            3
18      38      unknown                4
19      42      university.degree      3
20      54      university.degree      3

SELECT
  age,
  job,
  campaign,
  (SELECT AVG(campaign) FROM imposing-vista-456007-p4.bankmarketing.bank ) AS
overall_avg_balance
FROM imposing-vista-456007-p4.bankmarketing.bank
LIMIT 20;
Row     age     job     campaign        overall_avg_balance
1       56      housemaid      1        2.5675925026706832
2       57      services       1        2.5675925026706832
3       37      services       1        2.5675925026706832
4       40      admin.  1       2.5675925026706832
5       56      services       1        2.5675925026706832
6       45      services       1        2.5675925026706832
7       59      admin.  1       2.5675925026706832
8       41      blue-collar    1        2.5675925026706832
9       24      technician     1        2.5675925026706832
10      25      services       1        2.5675925026706832
11      41      blue-collar    1        2.5675925026706832
12      25      services       1        2.5675925026706832
13      29      blue-collar    1        2.5675925026706832
14      57      housemaid      1        2.5675925026706832
15      35      blue-collar    1        2.5675925026706832
16      54      retired 1       2.5675925026706832
17      35      blue-collar    1        2.5675925026706832
18      46      blue-collar    1        2.5675925026706832
```

```
19      50      blue-collar     1       2.5675925026706832
20      39      management      1       2.5675925026706832
```

```
SELECT  job, age
FROM (
    SELECT job, COUNT(*) AS age
    FROM `imposing-vista-456007-p4.bankmarketing.bank`
    WHERE job = 'retired'
    GROUP BY job
    ORDER BY age DESC
    LIMIT 3
);
```

```
Row    job      age

1      retired  1720
```

```
SELECT  AVG(campaign) AS avg_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`  ;
```

```
Row       avg_campaign

    1       2.5675925026706832
```

```
SELECT  AVG(pdays) AS avg_pdays
FROM `imposing-vista-456007-p4.bankmarketing.bank`  ;
```

```
Row    avg_pdays
    1      962.47545401573279
```

```
SELECT  SUM(campaign) AS total_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`  ;
```

```
Row    total_campaign
    1      105754
```

```
SELECT  SUM(pdays) AS total_pdays
FROM `imposing-vista-456007-p4.bankmarketing.bank`  ;
```

```
Row    total_pdays
    1      39642439
```

```
SELECT  education, AVG(campaign) AS avg_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank’
GROUP BY education
ORDER BY avg_campaign DESC;
```

```
Row     education             avg_campaign
1       basic.4y              2.6005747126436773
2       unknown                2.5961871750433283
3       professional.course    2.5861148197596906
4       high.school           2.5685759327377857
5       university.degree      2.5635272846811312
6       basic.6y              2.5562827225130871
7       basic.9y              2.5323407775020792
8       illiterate            2.2777777777777777
```

```sql
SELECT  marital, SUM(campaign) AS total_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`
GROUP BY marital
ORDER BY total_campaign DESC;
```

```
Row     marital         total_campaign

1       married          64135

2       single           29311

3       divorced        12053

4       unknown         255
```

```sql
SELECT
  AVG(age) AS avg_age,
  SUM(campaign) AS total_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE education = 'basic.6y';
```

```
Row     avg_age         total_campaign

1       40.448952879581235   5859
```

```sql
SELECT
  AVG(age) AS avg_age,
  SUM(campaign) AS total_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE day_of_week = 'mon';
```

```
Row     avg_age         total_campaign

1       40.4124970636601      22526
```

```sql
SELECT
  AVG(age) AS avg_age,
  SUM(campaign) AS total_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE month = 'may';
```

```
Row     avg_age         total_campaign

1       39.031084319848929   33593
```

```sql
SELECT job,SUM(campaign) AS total_campaign
FROM `imposing-vista-456007-p4.bankmarketing.bank`
GROUP BY job
HAVING total_campaign > 2.2344
ORDER BY total_campaign DESC;
```

```
Row     job        total_campaign

1       admin.           27342

2       blue-collar       23676
```

3	technician	17379
4	services	10271
5	management	7240
6	retired	4260
7	self-employed	3781
8	entrepreneur	3692
9	housemaid	2798
10	unemployed	2600
11	student	1841
12	unknown	874

Creating views in BigQuery is a great way to save reusable queries for reporting and analysis. A view is like a virtual table that stores a query.

1. View: Average Campaign by Job

```
CREATE OR REPLACE VIEW mydataset.view_avg_campaign_by_job AS

SELECT job, AVG(campaign) AS avg_campaign

FROM `imposing-vista-456007-p4.bankmarketing.bank`

GROUP BY job

ORDER BY avg_campaign DESC;
```

2. View: Subscriber Summary by Education

```
CREATE OR REPLACE VIEW mydataset.view_subscribers_by_education AS

SELECT education, COUNT(*) AS total_subscribers

FROM `imposing-vista-456007-p4.bankmarketing.bank`

WHERE y = 'yes'

GROUP BY education

ORDER BY total_subscribers DESC;
```

3. View: Campaign Contact Stats by Marital Status

```
CREATE OR REPLACE VIEW mydataset.view_campaign_stats_by_marital AS
SELECT
  marital,
  COUNT(*) AS total_clients,
  SUM(campaign) AS total_contacts,
  AVG(campaign) AS avg_contacts_per_client
FROM `imposing-vista-456007-p4.bankmarketing.bank`
GROUP BY marital
ORDER BY total_contacts DESC;
```

4.View: High Campaign Clients Over 50

```
CREATE OR REPLACE VIEW mydataset.view_high_campaign_seniors AS
SELECT age, job, campaign, y
FROM `imposing-vista-456007-p4.bankmarketing.bank`
WHERE age > 50 AND campaign > 2.2345
ORDER BY campaign DESC;
```

5. View: Subscription Rate by Job

```
CREATE OR REPLACE VIEW mydataset.view_subscription_rate_by_job AS
SELECT
  job,
  COUNT(*) AS total_clients,
  SUM(CASE WHEN y = 'yes' THEN 1 ELSE 0 END) AS total_yes,
  ROUND(100 * SUM(CASE WHEN y = 'yes' THEN 1 ELSE 0 END) / COUNT(*), 2) AS subscription_rate
FROM mydataset.bankmarketing
GROUP BY job
ORDER BY subscription_rate DESC;
```

BigQuery doesn't use traditional indexes like in MySQL or PostgreSQL, it has optimization strategies that simulate index-like performance. Here's what you can do to optimize your queries in BigQuery:

1. Use Partitioning

If your table has a date, timestamp, or even integer column, you can partition the table. This reduces the amount of scanned data.

If your table has a column contact_date, you can partition on it like this:

```
CREATE OR REPLACE TABLE imposing-vista-45600
p4.bankmarketing.bankmarketing_partitioned
PARTITION BY DATE(contact_date)
AS SELECT * FROM imposing-vista-456007-p4.bankmarketing.bank;
```

2. Use Clustering

Clustering sorts your data physically on disk. Good for repeated filtering, joining, or grouping by a specific column like `job`, `y`, `education`, etc.

```
CREATE OR REPLACE TABLE imposing-vista-456007-p4.bankmarketing _clustered

PARTITION BY DATE(contact_date)

CLUSTER BY job, y

AS SELECT * FROM imposing-vista-456007-p4.bankmarketing.bank;
```

3. Only Select Needed Columns

   Instead of SELECT *, specify the exact columns. This reduces scanned data.

   ```
   SELECT * FROM imposing-vista-456007-p4.bankmarketing.bank;
   SELECT age, job, y FROM imposing-vista-456007-p4.bankmarketing.bank ;
   ```

4. Use Filters Early (WHERE clause)

   Apply filters as early as possible to reduce the rows being processed.

   ```
   SELECT job, COUNT(*)
   FROM imposing-vista-456007-p4.bankmarketing.bank
   WHERE y = 'yes'
   GROUP BY job;
   ```

5. Materialize Heavy Queries (materialized views or temp tables)

   ```
   CREATE MATERIALIZED VIEW imposing-vista-456007-p4.bankmarketing.bank
   _avg_balance_by_job AS
   SELECT job, AVG(campaign) AS avg_campaign
   FROM imposing-vista-456007-p4.bankmarketing.bank
   GROUP BY job;
   ```

   In BigQuery, you can use JOIN operations like INNER JOIN, LEFT JOIN, and RIGHT JOIN to combine rows from two or more tables based on related columns.

1. INNER JOIN

Returns only rows that have matching values in both tables.

```
SELECT a.*, b.job
FROM `project.dataset.bank_marketing` a
INNER JOIN `project.dataset.other_table` b
ON a.id = b.id
```

2. LEFT JOIN

Returns all rows from the left table, and the matched rows from the right table. If no match, returns NULL on the right.

```
SELECT a.*, b.job

FROM `project.dataset.bank_marketing` a

LEFT JOIN `project.dataset.other_table` b

ON a.id = b.id
```

3. RIGHT JOIN

Returns all rows from the right table, and the matched rows from the left table. If no match, returns NULL on the left.

```
SELECT a.*, b.job

FROM `project.dataset.bank_marketing` a

RIGHT JOIN `project.dataset.other_table` b

ON a.id = b.id
```