**Corporate Reporting**

1.Project Plan: Project Pan is designed and is uploaded at Trello board at [Link](https://trello.com/c/8YipM1aX/168-project-plan-for-corporate-reporting)

2.The Data fields of fitness data are studied and analysed to know their functionality and scope for reporting and below is the analysis:

|  |
| --- |
| **Time Stamp:** The FIT Profile defines the date\_time type as an uint32 that represents the number of seconds since midnight on December 31, 1989 UTC. (UTC is AEDT + 10). |
| **Position\_lat/Position\_long:** Garmin GPS devices are set by factory default to lat/long DM. This means it is set to latitude and longitude in degrees and minutes, with decimal minutes. |
| i.e., it is represented as 156° 44', 72° 10'. |
| **Distance:** The Garmin automotive devices can show distance in either miles/feet (statute units) or kilometres/meters (metric units) |
| **Enhanced\_altitude/Altitude:** The device will measure changes in air pressure to determine your elevation. This information is recorded during your activity and is used to report elevation related information in Garmin Connect. Elevation calibrated by GPS is accurate to +/-400 feet with a strong GPS signal. If the values of altitude are too large to be fit in Altitude, then enhanced altitude is used. |
| **Ascent:** a climb or walk to the summit of a mountain or hill/an instance of rising or moving up through the air. Total Ascent provides a total of all increases to elevation (also known as elevation gain). Average Ascent provides an average of all ascents recorded during an activity. Maximum Elevation provides the highest elevation achieved. |
| **Grade:** Data field for Garmin devices that calculates the slope (or grade) of the hill you are walking on. It publishes the grade value (in %) to Garmin Connect so you can have a timeline inside your activity. |
| **Calories:** This is the total of active and resting calories that are calculated during a recorded activity on your device (from the moment that you start the timer for the activity to the moment you stop the timer). Speed/Distance Algorithm: This is the most basic method of determining calories. It is represented in calories/Kcal. |
| **Speed/Enhanced Speed:** It is distance by total time sent on an activity. It is calculated in m/sec or m/h. If the values of speed are too large to be fit in speed, then enhanced speed is used. |
| **Heart\_rate:** heart rate values can be set as absolute or relative values. Absolute values represent beats per minute (bpm) for heart rate, or watts for power. |
| **Temperature:** The Temperature widget will display the ambient air temperature near the barometric altimeter port. This reading can be affected by body heat.It is is represented in Fahrenheit. |
| **Cadence:** The cadence fields in a FIT file represent RPMs. For cycling 1 RPM equals one full rotation of the cranks. For running 1 RPM represents a step. |
| **Power**: Power values can be set as absolute or relative values. Absolute values represent watts for power. |
| **Left\_right\_balance:** It shows as a percentage the power separately put out by the left and right leg. |
| **Gps\_accuracy:**  It represent the drift with accurate gps values. GPS location accuracy is around 3 meters (10 feet), 95% of the time on Garmin devices. This means, at any given time, your device will save your location within 3 meters of your actual location. |
| **Product\_Name:** It describes the product used for recording the activity. |
| **Serial\_Number:** Most Garmin devices will have a unique serial number listed on the back or bottom of the device. |
| **Age:** Available on select Garmin watches, Fitness Age is an estimate of how fit you are compared to your actual age. Compatible Garmin watches will measure your Fitness Age differently, depending on which device you have. Fitness age is an estimate of how fit you are compared to your actual age. |
| **Gender:** It Shows gender of registered person of the device. |
| **Weight:** It gives the weight of the person in kgs. |
| **FTP:** Functional Threshold Power (FTP) is a measurement from power meters. It is the highest power level you can maintain for one hour without growing fatigued. FTP is beneficial because it provides an outlook on performance ability. |
| **Session\_ID:** It is unique id generated for each session performed by user. |
| **User\_ID:** It is the unique ID generated for every user. |

3. Queries for Views are created to fetch out different fields:

All the Queries were done on `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy` table. A copy of `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data’ is made and data is updated such that there is current year data to have visual output of all queries.

**DISTANCE:** Total Distance travelled by the user in last month, last week, previous day are fetched out by using the below queries.

**MONTHLY\_REPORT:**

SELECT userID,sum(distance) AS TOTAL\_DISTANCE

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE) - 1

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT (YEAR from CURRENT\_DATE)

group by userID

order by userID;

**WEEKLY\_REPORT:**

SELECT userID,sum(distance) AS TOTAL\_DISTANCE

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (DAY from DATE\_AEST) > EXTRACT(DAY FROM CURRENT\_DATE) - 7 and

EXTRACT (DAY from DATE\_AEST) <= EXTRACT(DAY FROM CURRENT\_DATE)

and EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE)

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT(YEAR FROM CURRENT\_DATE)

group by userID

order by userID;

**DAILY\_REPORT:**

SELECT userID,sum(distance) AS TOTAL\_DISTANCE

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (DAY from DATE\_AEST) = EXTRACT(DAY FROM CURRENT\_DATE) - 1 and

EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE)

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT(YEAR FROM CURRENT\_DATE)

group by userID,date\_AESt

order by userID;

**CALORIES:** Total Calories burned by the user in last month, last week, previous day are fetched out by using the below queries.

**MONTHLY\_REPORT:**

SELECT userID,sum(Calories) AS TOTAL\_Calories\_Burned

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE) - 1

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT (YEAR from CURRENT\_DATE)

group by userID

order by userID;

**WEEKLY\_REPORT:**

SELECT userID,sum(Calories) AS TOTAL\_Calories\_Burned

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (DAY from DATE\_AEST) > EXTRACT(DAY FROM CURRENT\_DATE) - 7 and

EXTRACT (DAY from DATE\_AEST) <= EXTRACT(DAY FROM CURRENT\_DATE)

and EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE)

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT(YEAR FROM CURRENT\_DATE)

group by userID

order by userID;

**DAILY\_REPORT:**

SELECT userID,sum(Calories) AS TOTAL\_Calories\_Burned

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (DAY from DATE\_AEST) = EXTRACT(DAY FROM CURRENT\_DATE) - 1 and

EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE)

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT(YEAR FROM CURRENT\_DATE)

group by userID,date\_AESt

order by userID;

**WEIGHT:** Average weight maintained by the user in last month, last week, previous day are fetched out by using the below queries.

**MONTHLY\_REPORT:**

SELECT userID,avg(weight) AS Weight

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE) - 1

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT (YEAR from CURRENT\_DATE)

group by userID

order by userID;

**WEEKLY\_REPORT:**

SELECT userID,avg(weight) AS Weight

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (DAY from DATE\_AEST) > EXTRACT(DAY FROM CURRENT\_DATE) - 7 and

EXTRACT (DAY from DATE\_AEST) <= EXTRACT(DAY FROM CURRENT\_DATE)

and EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE)

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT(YEAR FROM CURRENT\_DATE)

group by userID

order by userID;

**DAILY\_REPORT:**

SELECT userID,avg(weight) AS Weight

FROM `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (DAY from DATE\_AEST) = EXTRACT(DAY FROM CURRENT\_DATE) - 1 and

EXTRACT (MONTH from DATE\_AEST) = EXTRACT(MONTH FROM CURRENT\_DATE)

and EXTRACT (YEAR from DATE\_AEST) = EXTRACT(YEAR FROM CURRENT\_DATE)

group by userID,date\_AESt

order by userID;

4.Views are created for the above drafted queries:

**Steps to Create Views:**

1. Copy the query on to new tab as below:

A screenshot of a computer

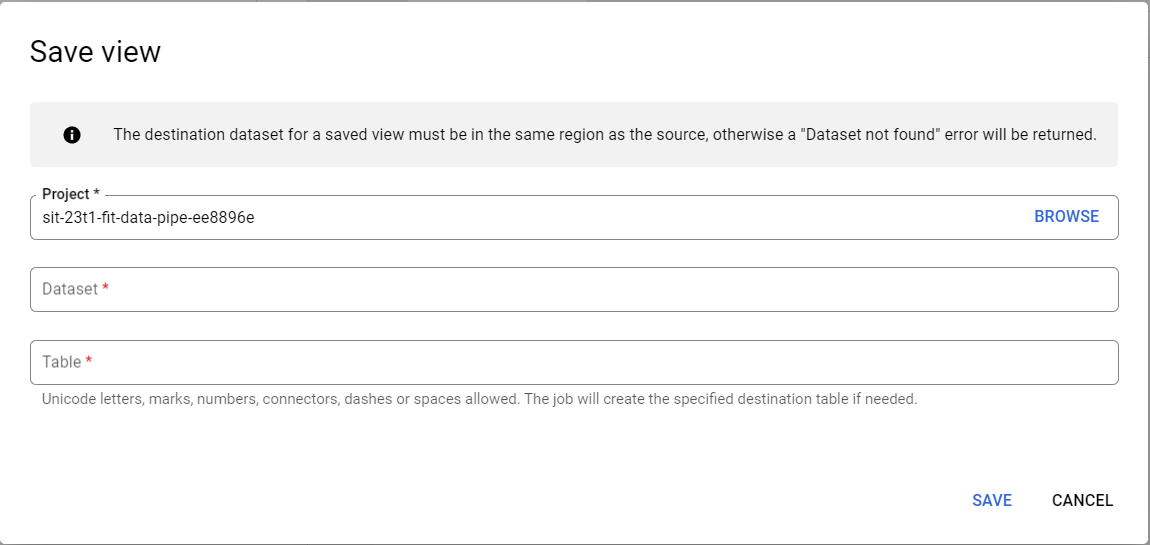
Description automatically generated with medium confidence

1. Click on Save view in the drop down next to save:

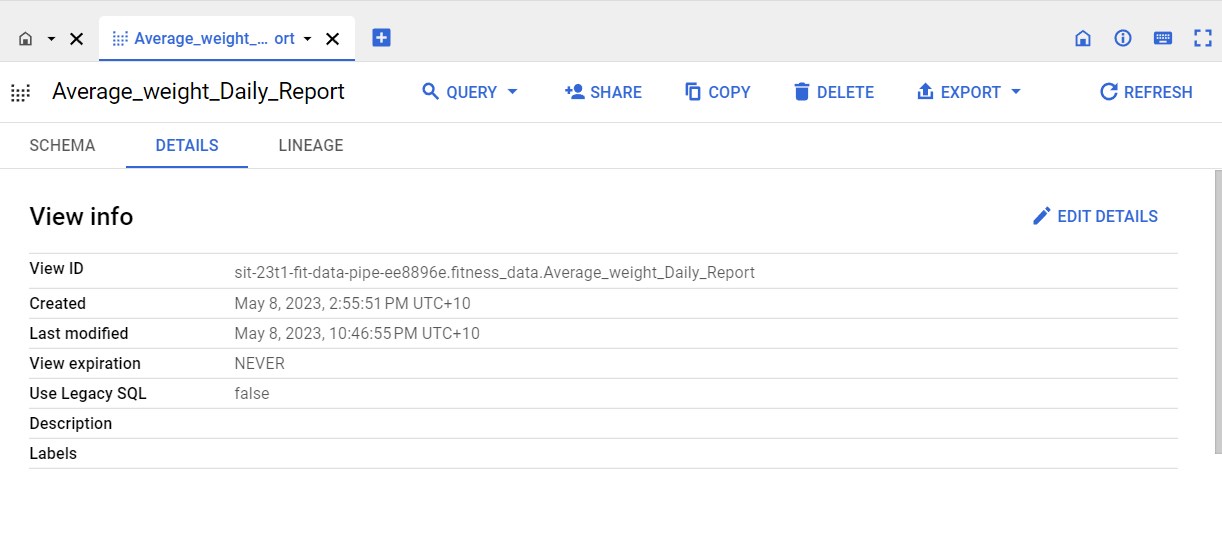
A screenshot of a computer

Description automatically generated with medium confidence

1. Now Give Dataset and View Name (as per choice) and click on save:



1. Now views are created, click on each view to see their details:



A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

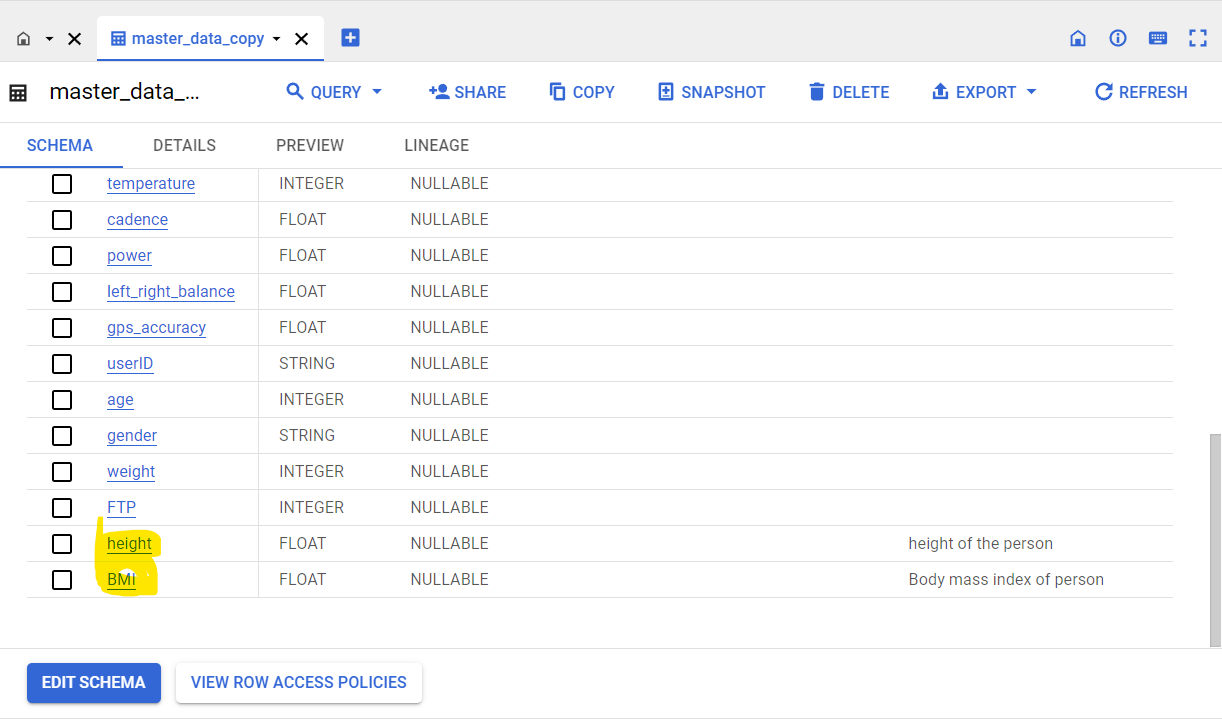
A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**5.** BMI is calculated for all the users in the table. To do this there are 2 columns added to the table Height and BMI.



Height Column is updated with few random values for different users using below query:

update `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

SET height = 165

where USERID= 'U1000006'

Now BMI value is calculated as follows:

update `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

SET BMI = weight\*10000/(height\*height)

where userid IN (SELECT distinct USERID from `sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`)

6. Queries are created to fetch out views for Monthly and Yearly BMI:

**BMI\_MONTHLY\_REPORT: (BMI of all users for previous month is generated)**

select userid,

avg(weight)AS weight,avg(height)as height,avg(BMI)as BMI,

case

when avg(BMI) < 18.5 then "Under weight"

when avg(BMI) >=18.5 and avg(BMI) < 25 then "Normal"

when avg(BMI) >=25 then "Obese"

end as Coach,

EXTRACT(month from date\_AEST) AS MONTH,

EXTRACT(YEAR from date\_AEST) AS YEAR

from

`sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (YEAR from DATE\_AEST) = EXTRACT (YEAR from CURRENT\_DATE) and

EXTRACT (MONTH from DATE\_AEST) = EXTRACT (MONTH from CURRENT\_DATE)-1

group by userid,month,year

**BMI\_YEARLY\_REPORT: (BMI of users for all months in the year so far is generated)**

select userid,

avg(weight)AS weight,avg(height)as height,avg(BMI)as BMI,

case

when avg(BMI) < 18.5 then "Under weight"

when avg(BMI) >=18.5 and avg(BMI) < 25 then "Normal"

when avg(BMI) >=25 then "Obese"

end as Coach,

EXTRACT(month from date\_AEST) AS MONTH

from

`sit-23t1-fit-data-pipe-ee8896e.fitness\_data.master\_data\_copy`

where EXTRACT (YEAR from DATE\_AEST) = EXTRACT (YEAR from CURRENT\_DATE) and

EXTRACT (MONTH from DATE\_AEST) <= EXTRACT (MONTH from CURRENT\_DATE)

group by userid,month

7. Views are created using the above queries:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with low confidence

8.Visual Representation of BMI is Developed:

**BMI\_YEARLY\_REPORT:**

1. Click on BMI\_YEARLY\_REPORT view in the big query explorer and View opens in the right pane as below:

A screenshot of a computer

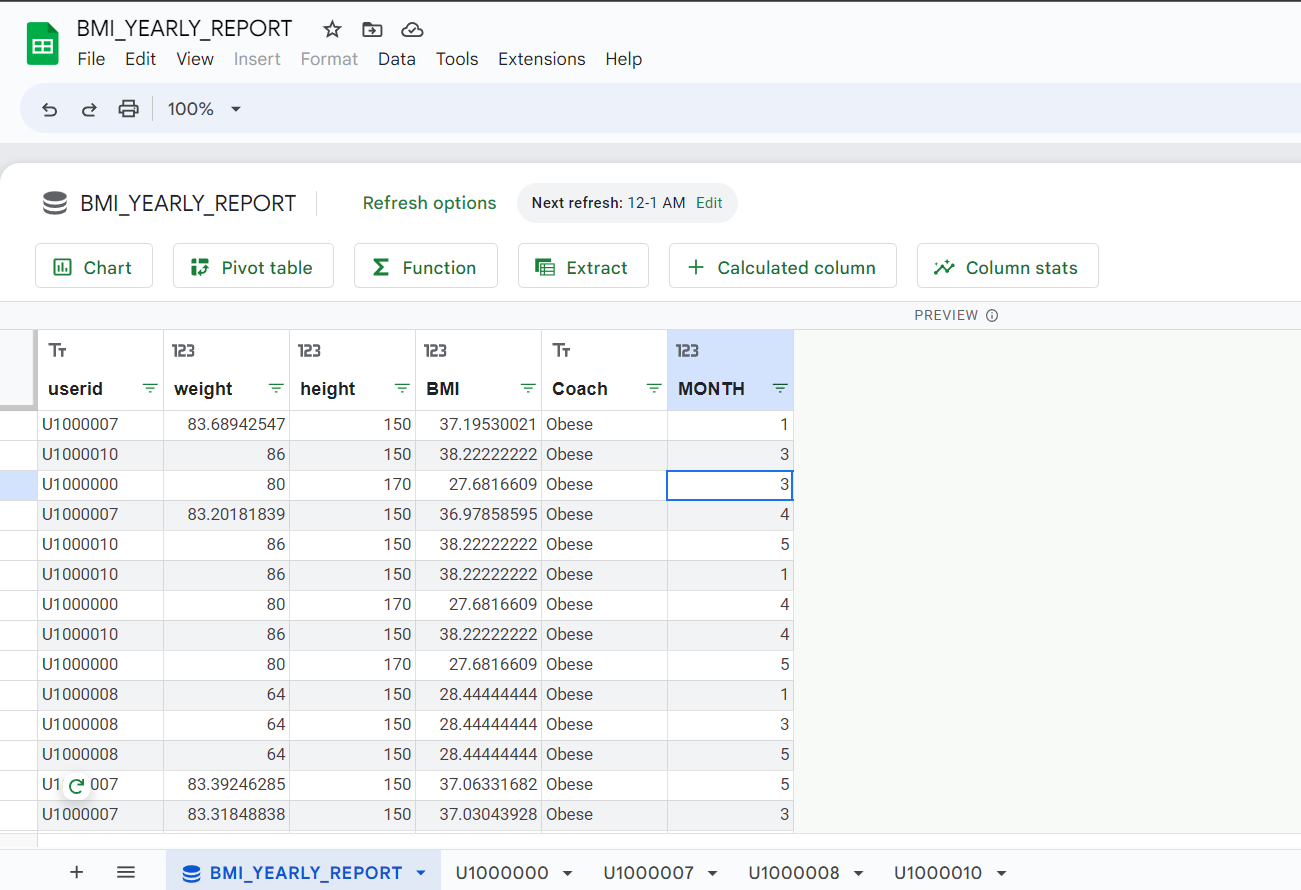
Description automatically generated with medium confidence

1. Click on drop down next to export label in the right pane and there will be option explore with sheets as below:

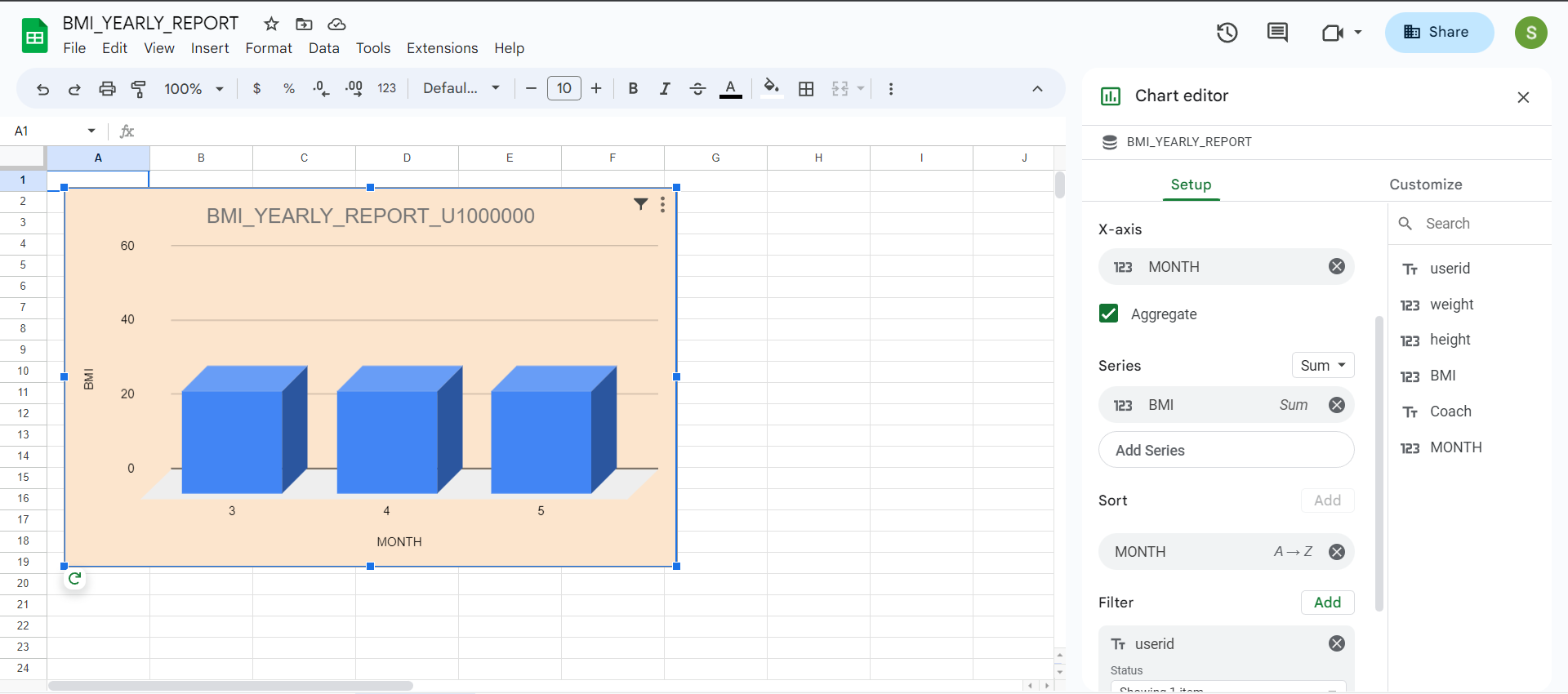
A screenshot of a computer

Description automatically generated with medium confidence

1. Click on Explore with sheets and new sheets opens with preview data from view result. Name that sheet to BMI\_YEARLY\_REPORT:



1. Now click on Chart Label to create charts and new sheet opens next to it:



1. Above Report is created by selecting userid in x-axis and BMI in series and sorted by MONTH and filtered by userid columns.
2. Select one user id from filter to fetch BMI of user for all months in the year.
3. Explore the customize options to style the dashboard.
4. Sheets for different users are created as below by selecting each user in each sheet:

A screenshot of a computer

Description automatically generated

1. In the first sheet where data preview is present click on Schedule Refresh option on top to schedule the refresh of reports:

A screenshot of a computer

Description automatically generated with medium confidence

1. All the sheets in the report can be scheduled for refresh:

A screenshot of a computer

Description automatically generated

1. Reports can be published to audience by clicking on share on top right and giving access to Deakin group only in view of security:

A screenshot of a computer

Description automatically generated with medium confidence

1. Now this report can be accessed using URL: <https://docs.google.com/spreadsheets/d/1xOI-TcMc_0_mIxODiJQKkvOOvC21zWWOXIEvgsvVZUo/edit?usp=sharing>

**BMI\_MONTHLY\_REPORT:**

BMI\_MONTHLY\_REPORT is also visualised using the above steps. But here data of all users for previous month is shown.so only one chart is created with userid in x-axis and BMI in series and sort by user id:

A screenshot of a computer

Description automatically generated with medium confidence

This report can be accessed using:

<https://docs.google.com/spreadsheets/d/1Cg5NDlH2vKCdIj6nt2cYlkssPWfPipauB3Zjih1osLA/edit?usp=sharing>