-- Data Description

/\*\*

In order to have a better data description we usually check the Shape and Size of our dataset along with the general description of datasets such as count, unique values etc.

\*\*/

-- Check the Size of dataset

SELECT count(column\_name)

FROM caba-demo.stdnt.INFORMATION\_SCHEMA.COLUMNS

WHERE table\_name = 'stnd';

-- This showed we had 9 columns

select count(gender) as row\_count from `caba-demo.stdnt.stnd`;

-- 1000 Rows in the dataset

-- Check the column types

SELECT column\_name, data\_type

FROM caba-demo.stdnt.INFORMATION\_SCHEMA.COLUMNS

WHERE table\_name = 'stnd';

-- Check the metadata / Description of dataset

SELECT table\_name, column\_name, is\_nullable, data\_type, is\_partitioning\_column

FROM `caba-demo.stdnt`.INFORMATION\_SCHEMA.COLUMNS;

-- Check the subset of observations recorded in the dataset

select \* from `caba-demo.stdnt.stnd` limit 10;

-- Data Aggregation

/\*\*

It helps us understand the data trends and values based on the compact display of values

\*\*/

-- check the unique values alongwith the count of values for categorical columns

SELECT count(gender) as gender\_counter, gender FROM

`caba-demo.stdnt.stnd` group by gender;

SELECT count(race\_ethnicity) as race\_counter, race\_ethnicity FROM

`caba-demo.stdnt.stnd` group by race\_ethnicity;

SELECT count(parental\_level\_education) as parental\_level\_education\_counter, parental\_level\_education FROM

`caba-demo.stdnt.stnd` group by parental\_level\_education;

SELECT count(lunch) as lunch\_counter, lunch FROM

`caba-demo.stdnt.stnd` group by lunch;

-- check the min and max of numerical columns

SELECT min(math\_score) as math\_score\_min, max(math\_score) as math\_score\_max,

min(reading\_score) as reading\_score\_min, max(reading\_score) as reading\_score\_max,

min(writing\_score) as writing\_score\_min, max(writing\_score) as writing\_score\_max,

min(obtained\_score) as obtained\_score\_min, max(obtained\_score) as obtained\_score\_max

FROM `caba-demo.stdnt.stnd`;

-- Filter values

-- Filter values for group A

select \* from `caba-demo.stdnt.stnd`

where race\_ethnicity = "group A";

-- Filter values for Bachelor's and Master's Parental level of education

select \* from `caba-demo.stdnt.stnd`

where parental\_level\_education in ("master's degree", "bachelor's degree");

-- Sort Values

select \* from `caba-demo.stdnt.stnd` order by obtained\_score desc;

-- Data Summary

/\*\*

Summaries of data help us understand the detailed trends followed in datasets based on concise information using measures of location and spread

\*\*/

-- Summarize by mean values

select test\_preparation, avg(math\_score) as Mean\_Mathscore, avg(reading\_score) as Mean\_Readingscore, avg(writing\_score) as Mean\_Writingscore,

avg(obtained\_score) as Mean\_Obtainedscore from `caba-demo.stdnt.stnd` group by test\_preparation;

-- summarize by median

select PERCENTILE\_CONT(obtained\_score,0.5) OVER() AS median\_obtainedscore , PERCENTILE\_CONT(writing\_score,0.5) OVER() AS median\_writingscore,

PERCENTILE\_CONT(math\_score,0.5) OVER() AS median\_mathscore, PERCENTILE\_CONT(reading\_score,0.5) OVER() AS median\_readingscore

from `caba-demo.stdnt.stnd` limit 1;

--summarize by mode

SELECT

APPROX\_TOP\_COUNT(obtained\_score,1) AS mode\_obtainedscore, APPROX\_TOP\_COUNT(writing\_score,1) AS mode\_writingscore,

APPROX\_TOP\_COUNT(math\_score,1) AS mode\_mathscore, APPROX\_TOP\_COUNT(reading\_score,1) AS mode\_readingscore

from `caba-demo.stdnt.stnd` limit 1;

-- stddev

SELECT

STDDEV(obtained\_score) AS standard\_deviation

FROM `caba-demo.stdnt.stnd`;

-- creating temp tables with grouped information

drop table if exists `caba-demo.stdnt.groupbyparental`;

Create table `caba-demo.stdnt.groupbyparental`

as

(SELECT avg(obtained\_score) as Obtained\_score, parental\_level\_education FROM

`caba-demo.stdnt.stnd` group by parental\_level\_education);

select \* from `caba-demo.stdnt.groupbyparental` order by Obtained\_score;

-- Making quartiles by Obtained Score

drop table if exists `caba-demo.stdnt.IQR`;

Create table `caba-demo.stdnt.IQR` as (

select PERCENTILE\_CONT(obtained\_score,0.25) OVER() AS q1\_obtainedscore , PERCENTILE\_CONT(obtained\_score,0.75) OVER() AS q3\_obtainedscore,

(PERCENTILE\_CONT(obtained\_score,0.75) OVER()) - (PERCENTILE\_CONT(obtained\_score,0.25) OVER()) as IQR

from `caba-demo.stdnt.stnd` limit 1);

select (q1\_obtainedscore - (1.5\* IQR)) as maximum, (q3\_obtainedscore + (1.5\* IQR)) as minimum from

`caba-demo.stdnt.IQR`;

-- identify the outliers

select \* from `caba-demo.stdnt.stnd` where obtained\_score > 323.8 or obtained\_score < 86.8;