CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`%`

SQL SECURITY DEFINER

VIEW `opportunities\_view` AS

SELECT DISTINCT

'Database' AS `Source.Name`,

`opportunity\_statuses`.`name` AS `Status`,

`opportunities`.`id` AS `Unique Code`,

CONCAT(`pre\_sales\_view`.`first\_name`,

' ',

`pre\_sales\_view`.`last\_name`) AS `Pre-Sales`,

CONCAT(`user\_view\_bde`.`first\_name`,

' ',

`user\_view\_bde`.`last\_name`) AS `BDM`,

`tech\_team\_view`.`tech\_team` AS `Tech Team`,

`opportunities`.`name` AS `Opportunity Name`,

(CASE

WHEN (`opp\_partner\_view`.`partner\_name` IS NULL) THEN `opp\_client\_view`.`client\_name`

ELSE CONCAT(`opp\_client\_view`.`client\_name`,

'/',

`opp\_partner\_view`.`partner\_name`)

END) AS `Client/Partner`,

`opportunities`.`received\_date` AS `Received Date`,

`opportunities`.`created\_at` AS `Created Date`,

`dev\_works\_view`.`task\_details` AS `Task Details`,

`opportunities`.`due\_date` AS `Submitted Date`,

(CASE

WHEN

((`opp\_status\_view`.`dev\_efforts` = 0)

OR (`opp\_status\_view`.`dev\_efforts` IS NULL))

THEN

0

ELSE `opp\_status\_view`.`dev\_efforts`

END) AS `Dev Efforts`,

(CASE

WHEN

((`opp\_status\_view`.`sdlc\_efforts` = 0)

OR (`opp\_status\_view`.`sdlc\_efforts` IS NULL))

THEN

0

ELSE `opp\_status\_view`.`sdlc\_efforts`

END) AS `SDLC Efforts`,

(CASE

WHEN

(((`opp\_status\_view`.`dev\_efforts` IS NULL)

AND (`opp\_status\_view`.`sdlc\_efforts` IS NULL))

OR ((`opp\_status\_view`.`dev\_efforts` + `opp\_status\_view`.`sdlc\_efforts`) IS NULL))

THEN

0

ELSE (`opp\_status\_view`.`dev\_efforts` + `opp\_status\_view`.`sdlc\_efforts`)

END) AS `Total Efforts`,

`dev\_works\_view`.`estimate\_type` AS `Type of Estimate`,

(CASE

WHEN (`clients`.`is\_partner` = 'yes') THEN 'Partner'

ELSE 'End Client'

END) AS `Client Type`,

`opportunities`.`business\_type` AS `Business Type`,

`geographies`.`name` AS `Geography`,

`client\_size\_view`.`opportunity\_type` AS `Opportunity Type`,

`business\_domains`.`name` AS `Domain`,

`business\_domains`.`name` AS `New Domain`,

`latest\_status\_view`.`comment` AS `Next Action Planned/Expected`,

`opportunities`.`due\_date` AS `Tentative Submission Date`,

`priorities`.`name` AS `Priority`,

(CASE

WHEN (`opportunities`.`mark\_as\_won` = 'yes') THEN 'Won'

ELSE ''

END) AS `Result`,

`opportunities`.`requirement` AS `Remarks`,

`tech\_name\_view`.`tech\_name` AS `Technology 1`,

`tech\_stack\_view`.`tech\_name` AS `Technology Stack`,

`opp\_qualify\_view`.`grade` AS `Grade`

FROM

(((((((((((((((((((`opportunities`

LEFT JOIN `opportunity\_statuses` ON ((`opportunities`.`status` = `opportunity\_statuses`.`id`)))

LEFT JOIN `pre\_sales\_view` ON ((`opportunities`.`presales\_owner\_id` = `pre\_sales\_view`.`id`)))

LEFT JOIN `user\_view\_bde` ON ((`opportunities`.`bdm\_bde\_id` = `user\_view\_bde`.`id`)))

LEFT JOIN `tech\_team\_view` ON ((`opportunities`.`id` = `tech\_team\_view`.`opportunity\_id`)))

LEFT JOIN `opp\_client\_view` ON ((`opportunities`.`client\_id` = `opp\_client\_view`.`client\_id`)))

LEFT JOIN `opp\_partner\_view` ON ((`opportunities`.`partner\_id` = `opp\_partner\_view`.`partner\_id`)))

LEFT JOIN `dev\_works\_view` ON ((`opportunities`.`id` = `dev\_works\_view`.`opportunity\_id`)))

LEFT JOIN `opp\_status\_view` ON ((`opportunities`.`id` = `opp\_status\_view`.`id`)))

LEFT JOIN `clients` ON ((`opportunities`.`client\_id` = `clients`.`id`)))

LEFT JOIN `geographies` ON ((`clients`.`geography\_id` = `geographies`.`id`)))

LEFT JOIN `client\_size\_view` ON ((`opportunities`.`client\_id` = `client\_size\_view`.`client\_id`)))

LEFT JOIN `business\_domains` ON ((`opportunities`.`business\_domain\_id` = `business\_domains`.`id`)))

LEFT JOIN `latest\_status\_view` ON ((`opportunities`.`id` = `latest\_status\_view`.`opportunity\_id`)))

LEFT JOIN `priorities` ON ((`opportunities`.`priority\_id` = `priorities`.`id`)))

LEFT JOIN `tech\_name\_view` ON ((`opportunities`.`id` = `tech\_name\_view`.`opportunity\_id`)))

LEFT JOIN `tech\_stack\_view` ON ((`opportunities`.`id` = `tech\_stack\_view`.`opportunity\_id`)))

LEFT JOIN `opp\_qualify\_view` ON ((`opportunities`.`opportunity\_qulification\_points\_id` = `opp\_qualify\_view`.`id`)))))

WHERE

(`opportunities`.`deleted\_at` IS NULL)

GROUP BY `latest\_status\_view`.`comment` , `tech\_name\_view`.`tech\_name` , `client\_size\_view`.`opportunity\_type` , `tech\_team\_view`.`tech\_team` , `opp\_status\_view`.`dev\_efforts` , `opp\_status\_view`.`sdlc\_efforts` , `opportunities`.`id` , `pre\_sales\_view`.`first\_name` , `pre\_sales\_view`.`last\_name` , `user\_view\_bde`.`first\_name` , `user\_view\_bde`.`last\_name` , `opportunities`.`name` , `opp\_client\_view`.`client\_name` , `opp\_partner\_view`.`partner\_name` , `dev\_works\_view`.`estimate\_type` , `dev\_works\_view`.`task\_details` , `opportunities`.`due\_date`

Other small views

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`%`

SQL SECURITY DEFINER

VIEW `client\_size\_view` AS

SELECT DISTINCT

`opportunities`.`client\_id` AS `client\_id`,

`clients`.`name` AS `name`,

`company\_sizes`.`size` AS `size`,

(CASE

WHEN (`company\_sizes`.`size` = '1 to 10') THEN 'Startup'

WHEN (`company\_sizes`.`size` = '11 to 50') THEN 'SmallBiz'

WHEN (`company\_sizes`.`size` = '51 to 100') THEN 'Midsize'

WHEN (`company\_sizes`.`size` = '101 to 500') THEN 'Midsize'

WHEN (`company\_sizes`.`size` = '501 to 1000') THEN 'Enterprise'

WHEN (`company\_sizes`.`size` = '1001 to 5000') THEN 'Enterprise'

WHEN (`company\_sizes`.`size` = '> 5000') THEN 'Enterprise'

WHEN (`company\_sizes`.`size` = 'Not Known') THEN 'Enterprise'

ELSE ''

END) AS `opportunity\_type`

FROM

((`opportunities`

LEFT JOIN `clients` ON ((`opportunities`.`client\_id` = `clients`.`id`)))

LEFT JOIN `company\_sizes` ON ((`clients`.`company\_size\_id` = `company\_sizes`.`id`)))

Array Creation

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`%`

SQL SECURITY DEFINER

VIEW `dev\_works\_view` AS

SELECT

`opportunity\_status\_logs`.`opportunity\_id` AS `opportunity\_id`,

GROUP\_CONCAT(`opportunity\_status\_logs`.`comment`

SEPARATOR ', ') AS `task\_details`,

GROUP\_CONCAT(`development\_works`.`name`

SEPARATOR ', ') AS `estimate\_type`

FROM

((`opportunities`

LEFT JOIN `opportunity\_status\_logs` ON ((`opportunities`.`id` = `opportunity\_status\_logs`.`opportunity\_id`)))

LEFT JOIN `development\_works` ON ((`opportunity\_status\_logs`.`development\_work\_id` = `development\_works`.`id`)))

GROUP BY `opportunity\_status\_logs`.`opportunity\_id`

SECOND INSTANCE

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`%`

SQL SECURITY DEFINER

VIEW `tech\_name\_view` AS

SELECT

GROUP\_CONCAT(`technologies`.`name`

SEPARATOR ',') AS `tech\_name`,

`opportunity\_technologies`.`opportunity\_id` AS `opportunity\_id`

FROM

((`opportunities`

LEFT JOIN `opportunity\_technologies` ON ((`opportunities`.`id` = `opportunity\_technologies`.`opportunity\_id`)))

LEFT JOIN `technologies` ON ((`opportunity\_technologies`.`technology\_id` = `technologies`.`id`)))

GROUP BY `opportunity\_technologies`.`opportunity\_id`

Max Date Logic

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`%`

SQL SECURITY DEFINER

VIEW `max\_date\_view` AS

SELECT

`opportunities`.`id` AS `opportunity\_id`,

MAX(`opportunity\_status\_logs`.`created\_at`) AS `max\_date`

FROM

(`opportunities`

LEFT JOIN `opportunity\_status\_logs` ON ((`opportunities`.`id` = `opportunity\_status\_logs`.`opportunity\_id`)))

WHERE

(`opportunity\_status\_logs`.`created\_at` = (SELECT MAX(`opportunity\_status\_logs`.`created\_at`)))

GROUP BY `opportunities`.`id`

CASE WHEN

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`%`

SQL SECURITY DEFINER

VIEW `opp\_qualify\_view` AS

SELECT

`opportunity\_qulification\_points`.`id` AS `id`,

`opportunity\_qulification\_points`.`total\_qulification\_point` AS `total\_qulification\_point`,

`opportunity\_qulification\_points`.`total\_overall\_qulification\_point` AS `total\_overall\_qulification\_point`,

((`opportunity\_qulification\_points`.`total\_qulification\_point` / `opportunity\_qulification\_points`.`total\_overall\_qulification\_point`) \* 100) AS `percentage`,

(CASE

WHEN (((`opportunity\_qulification\_points`.`total\_qulification\_point` / `opportunity\_qulification\_points`.`total\_overall\_qulification\_point`) \* 100) BETWEEN 91 AND 100) THEN 'Very Highly Qualified'

WHEN (((`opportunity\_qulification\_points`.`total\_qulification\_point` / `opportunity\_qulification\_points`.`total\_overall\_qulification\_point`) \* 100) BETWEEN 76 AND 90) THEN 'High Qualified'

WHEN (((`opportunity\_qulification\_points`.`total\_qulification\_point` / `opportunity\_qulification\_points`.`total\_overall\_qulification\_point`) \* 100) BETWEEN 61 AND 75) THEN 'Qualified'

WHEN (((`opportunity\_qulification\_points`.`total\_qulification\_point` / `opportunity\_qulification\_points`.`total\_overall\_qulification\_point`) \* 100) BETWEEN 51 AND 60) THEN 'Low Qualified'

WHEN (((`opportunity\_qulification\_points`.`total\_qulification\_point` / `opportunity\_qulification\_points`.`total\_overall\_qulification\_point`) \* 100) BETWEEN 36 AND 50) THEN 'Very Low Qualified'

WHEN (((`opportunity\_qulification\_points`.`total\_qulification\_point` / `opportunity\_qulification\_points`.`total\_overall\_qulification\_point`) \* 100) BETWEEN 0 AND 36) THEN 'Very Low Qualified'

ELSE ''

END) AS `grade`

FROM

`opportunity\_qulification\_points`

SUM CALCULATION

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`%`

SQL SECURITY DEFINER

VIEW `opp\_status\_view` AS

SELECT

`opportunities`.`id` AS `id`,

SUM(`opp\_status\_logs\_view`.`dev\_efforts`) AS `dev\_efforts`,

SUM(`opp\_status\_logs\_view`.`sdlc\_efforts`) AS `sdlc\_efforts`

FROM

(`opportunities`

LEFT JOIN `opp\_status\_logs\_view` ON ((`opportunities`.`id` = `opp\_status\_logs\_view`.`opportunity\_id`)))

GROUP BY `opportunities`.`id`

MTD/QTD Measure

MTD/QTD/YTD =

var TodayDate = TODAY()

var Yesterday = TODAY()-1

var Tomorrow = TODAY()+1

var StartofWeek = CALCULATE(MAX('Date'[StartofWeek]),'Date'[Date] = TODAY())

var EndofWeek = CALCULATE(MAX('Date'[EndofWeek]),'Date'[Date] = TODAY())

var MonthStart = CALCULATE(STARTOFMONTH('Date'[Date]),YEAR('Date'[Date]) = YEAR(TodayDate), MONTH('Date'[Date]) = MONTH(TodayDate))

var CurrentMonthEnd = EOMONTH(today(),0)

var LastMonthStart = CALCULATE(STARTOFMONTH('Date'[Date]),YEAR('Date'[Date]) = YEAR(TodayDate), MONTH('Date'[Date]) = MONTH(TodayDate)-1)

var LastMonthEnd = EOMONTH(today(),-1)

var LastQuarterStart = CALCULATE(STARTOFQUARTER('Date'[Date]),YEAR('Date'[Date]) = YEAR(TodayDate), QUARTER('Date'[Date]) = QUARTER(TodayDate)-1)

var LastQuarterEnd = CALCULATE(ENDOFQUARTER('Date'[Date]),YEAR('Date'[Date]) = YEAR(TodayDate), QUARTER('Date'[Date]) = QUARTER(TodayDate)-1)

var LastYearStart = CALCULATE(STARTOFYEAR('Date'[Date]),YEAR('Date'[Date]) = YEAR(TodayDate)-1)

var LastYearEnd = CALCULATE(ENDOFYEAR('Date'[Date]),YEAR('Date'[Date]) = YEAR(TodayDate)-1)

var Result =

UNION(

ADDCOLUMNS(

CALENDAR( EndofWeek-13,StartofWeek-1),

"Selection", "Last Week",

"Sort", 2

),

ADDCOLUMNS(

CALENDAR( LastMonthStart,LastMonthEnd),

"Selection", "Last Month",

"Sort", 4

),

ADDCOLUMNS(

CALENDAR( LastQuarterStart,LastQuarterEnd),

"Selection", "Last Quarter",

"Sort", 5

),

ADDCOLUMNS(

CALENDAR( LastYearStart, LastYearEnd ),

"Selection", "Last Year",

"Sort", 6

),

ADDCOLUMNS(

CALENDAR( MonthStart,CurrentMonthEnd),

"Selection", "Current Month",

"Sort", 3

)

)

Return

Result

FORMATTING

Date Timestamp New = FORMAT('Lead Collection'[DateTimeStamp],"General Date")

Lead Selection =

SWITCH(

TRUE(),

'Lead Collection'[IsCleanDeduped]=0,"Gross Leads",

'Lead Collection'[IsCleanDeduped]=1,"Clean and Deduped Leads",

'Lead Collection'[IsCleanDeduped]=2,"Clean and Deduped Leads")

Week Ending Date =

//var StartofWeek = [DateTimeStamp] - WEEKDAY ( [DateTimeStamp] ) +1

//var EndofWeek = [DateTimeStamp] + 6 - WEEKDAY([DateTimeStamp],2)

var EOW = [DateTimeStamp] - MOD([DateTimeStamp],7)+6

//var a = .EndOfWeek([DateTimeStamp],Day.Monday)

Return

EOW

Week Starting Date =

//var StartofWeek = [DateTimeStamp] - WEEKDAY ( [DateTimeStamp] ) +1

//var EndofWeek = [DateTimeStamp] + 6 - WEEKDAY([DateTimeStamp],2)

var EOW = [DateTimeStamp] - WEEKDAY([DateTimeStamp],2)+1

//var a = .EndOfWeek([DateTimeStamp],Day.Monday)

Return

EOW

WeekNo = WEEKNUM('Lead Collection'[DateTimeStamp],15)

CALENDAR

Calendar = CALENDAR (DATE(2019,1,1), DATE(2025,12,31))

// CALENDAR(FIRSTDATE(Arrivals[Arrivals]), LASTDATE(Departure[DepartureDate]))

DYNAMIC SLICER BASED ON STATIC TABLE

KPI Metrics = VAR MySelection=

SELECTEDVALUE(KpiTable[KPI Name])

RETURN

SWITCH(

TRUE(),

MySelection="Proposals Submitted",[Proposals Submitted],

MySelection="Efforts Won (Hrs)",[Efforts Won],

MySelection="Closure Ratio by Leads (%)",CALCULATE('Pre Sales Fact'[Closure Ratio by Leads]\*100),

MySelection="Closure Ratio by Efforts (%)",CALCULATE('Pre Sales Fact'[Closure Ratio by Efforts]\*100),

MySelection="Qualified Opportunities",[Qualified Opportunities],

MySelection="+ 800 hr Projects",[+ 800 hr Projects],

MySelection="Opportunities End Clients",[Opportunities End Clients],

MySelection="Mid to Large org Opp",[Mid to Large org Opp],

0)