##### create table loan\_table\_1 (id int,member\_id int,loan\_amnt int,funded\_amnt int,funded\_amnt\_inv int,term string,init\_rate string,installment float,grade string,subgrade string, emp\_title string,emp\_length string,home\_ownership string,annual\_inc bigint,verification\_status string,issue\_d string,loan\_status string,pymt\_plan string,url string,

##### desc string,purpose string,title string,zip\_code string,addr\_state string,dti float,delinq\_2yrs int,earliest\_cr\_line string,inq\_last\_6mths int, mths\_since\_last\_delinq string,mths\_since\_last\_record string,open\_acc int,pub\_rec int,revol\_bal int,revol\_util string,total\_acc int,initial\_list\_status string,out\_prncp float,out\_prncp\_inv double, total\_pymnt float,total\_pymnt\_inv float,total\_rec\_int float,total\_rec\_late\_fee float,recoveries float,collection\_recovery\_fee float,last\_pymnt\_d string, last\_pymnt\_amnt float,next\_pymnt\_d string,last\_credit\_pull\_d string,collections\_12\_mths\_ex\_med float,mths\_since\_last\_major\_derog string,policy\_code int, application\_type string,annual\_inc\_joint string,dti\_joint string,verification\_status\_joint string,acc\_now\_delinq int,tot\_coll\_amt string,tot\_cur\_bal string, open\_acc\_6m string,open\_il\_6m string,open\_il\_12m string,open\_il\_24m string,mths\_since\_rcnt\_il string,total\_bal\_il string,il\_util string, open\_rv\_12m string,open\_rv\_24m string,max\_bal\_bc string,all\_util string,total\_credit\_rv string,inq\_fi string,total\_fi\_tl string,inq\_last\_12m string)

##### ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

##### WITH SERDEPROPERTIES

##### (

##### "separatorChar" = ",",

##### "quoteChar" = "\""

##### ) STORED AS TEXTFILE

##### tblproperties('serialization.null.format'='','skip.header.line.count'='2');

##### LOAD DATA INPATH '/user/anabig114231/LoanStats3a.csv' INTO TABLE loan\_table\_1;

##### LOAD DATA INPATH '/user/anabig114231/LoanStats3b.csv' INTO TABLE loan\_table\_1;

##### LOAD DATA INPATH '/user/anabig114231/LoanStats3c.csv' INTO TABLE loan\_table\_1;

##### LOAD DATA INPATH '/user/anabig114231/LoanStats3d.csv' INTO TABLE loan\_table\_1;

##### create table reject\_table1(amount\_requested string,Application\_date string,loan\_title string,risk\_score string,Debt\_To\_Income string,zipcode string, state string,employment\_length string,policy\_code string)

##### ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

##### WITH SERDEPROPERTIES

##### (

##### "separatorChar" = ",",

##### "quoteChar" = "\""

##### ) STORED AS TEXTFILE

##### tblproperties('serialization.null.format'='','skip.header.line.count'='2');

##### LOAD DATA INPATH '/user/anabig114231/RejectStatsA.csv' INTO TABLE reject\_table1;

##### LOAD DATA INPATH '/user/anabig114231/RejectStatsB.csv' INTO TABLE reject\_table1;

##### LOAD DATA INPATH '/user/anabig114231/RejectStatsD.csv' INTO TABLE reject\_table1;

##### ## Total Loan issuance by yearly & quarterly and calculate growth rate by quarter on quarter and year on year

##### select sum(loan\_amnt) as total\_loan\_issued,

##### substr(issue\_d,5) Year

##### from loan\_table\_1

##### group by substr(issue\_d,5) ;

##### 

##### 

##### select Year, (diff/previous)\*100 Percentage\_Growth from

##### (select substr(issue\_d,5) Year, sum(loan\_amnt) - lag(sum(loan\_amnt)) over( order by substr(issue\_d,5)) diff,

##### lag(sum(loan\_amnt)) over(order by substr(issue\_d,5)) previous from loan\_table\_1

##### group by substr(issue\_d,5)) t1

##### order by Year;

##### create table tbl\_quarter1 as SELECT id,substr(issue\_d,1,3) month,loan\_amnt,substr(issue\_d,5) as loan\_year FROM loan\_table\_1;

##### create table quarter\_table1 as select id as id,case

##### when lower(month) in ('jan','feb','mar') then 1

##### when lower(month) in('apr','may','jun') then 2

##### when lower(month) in ('jul','aug','sep') then 3

##### when lower(month) in ('oct','nov','dec') then 4

##### else 0 end as year\_quarter, loan\_amnt as total\_amnt,loan\_year as loan\_year from tbl\_quarter1;

##### select year\_quarter, sum(total\_amnt) quarter\_wise\_loan

##### from quarter\_table1

##### where year\_quarter > 0

##### group by year\_quarter;

##### 

##### select loan\_year, year\_quarter, (diff/previous)\*100 Percentage\_Growth

##### from (select loan\_year , year\_quarter, sum(total\_amnt) - lag(sum(total\_amnt))

##### over( order by loan\_year,year\_quarter) diff,

##### lag(sum(total\_amnt)) over(order by loan\_year,year\_quarter) previous from quarter\_table1

##### group by loan\_year, year\_quarter ) t2

##### where year\_quarter <> 0 and not isnull((diff/previous)\*100)

##### 

##### 

##### 

##### 

##### # Percentage of loans based on reported loan purpose

##### create table loanstble\_1 as SELECT sum(loan\_amnt) as amount\_loan, purpose as purpose from loan\_table\_1 group by purpose;

##### create table loanstble\_2 as select sum(loan\_amnt) as total\_loan from loan\_table\_1;

##### create table loanstble\_3 as select loanstble\_1.purpose as purpose, loanstble\_1.amount\_loan as loan,loanstble\_2.total\_loan as total\_loan from loanstble\_1 cross join loanstble\_2;

##### select purpose as purpose, 100\*(loan/total\_loan) as Percent\_of\_total from loanstble\_3;

##### 

##### 

##### Loan Issuance by state – classify the states based on loan issuance by $50+ MM, $25-50 MM, $10-25 MM and $0-10 MM

Create table state\_category\_1 as Select addr\_state as state, sum(loan\_amnt) as total\_amnt From loan\_table\_1

Group by addr\_state;

Select state as addr\_state, total\_amnt as total\_loan, case

When total\_amnt <1000000 then 'LOW'

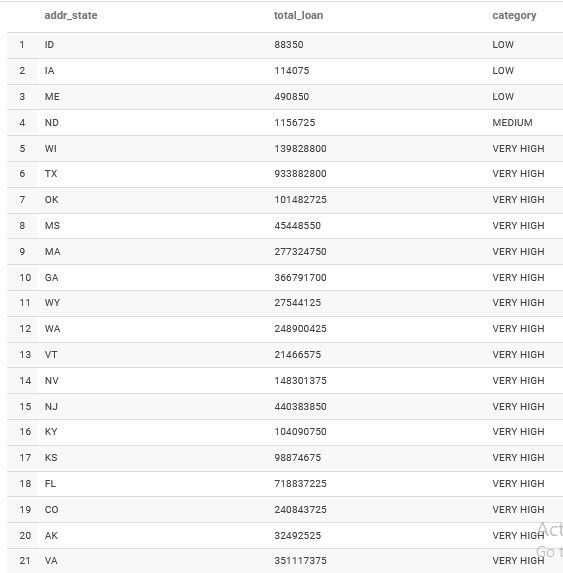
When total\_amnt <2500000 then 'MEDIUM'

When total\_amnt <5000000 then 'HIGH'

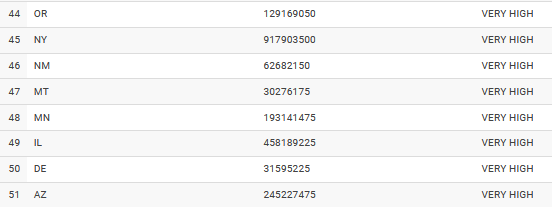
When total\_amnt >5000000 then 'VERY HIGH'

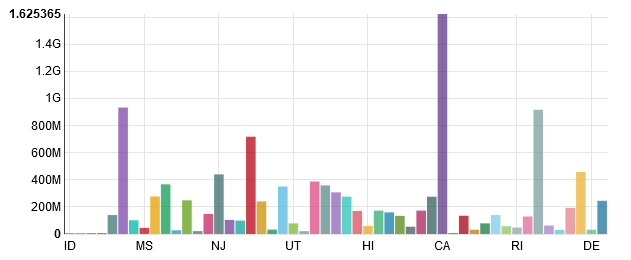
End As Category from state\_category\_1

Order by Category;









**## Find the last quarter average interest rates by different term loans and overall**

##### create table interest\_table1 as SELECT term as t\_term,regexp\_replace(int\_rate,'[^0-9.]','') as intrest\_rate from loan\_table\_1;

##### SELECT avg(intrest\_rate)as averageintrestrate\_byterm,t\_term as term FROM interest\_table1 GROUP BY t\_term;

##### SELECT avg(intrest\_rate)as averageintrestrate\_overall FROM interest\_table1;

##### 

##### 

##### 

##### ## Find the historical returns by loan grade(Historical performance by grade for all issued loans) and overall

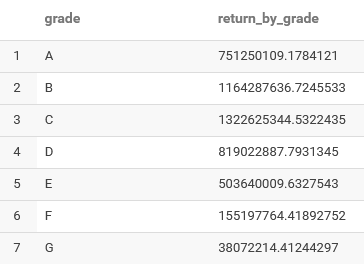
Create table returns\_table as select grade as grd\_1, sum(loan\_amnt) as loan, sum (total\_pymnt) as repay

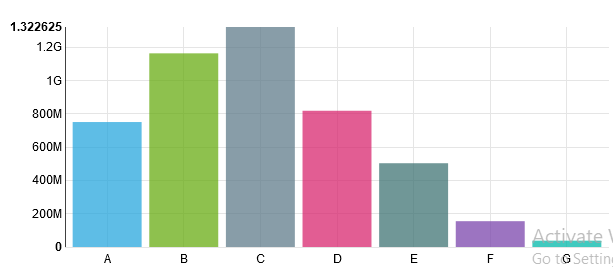
from loan\_table\_1 group by grade;

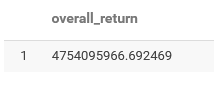
Create table history\_table as select grd\_1 as grade, (loan-repay) as return from returns\_table;

Select grade,sum( return) as return\_by\_grade from history\_table group by grade;

Select sum(return) as Overall\_return from history\_table;





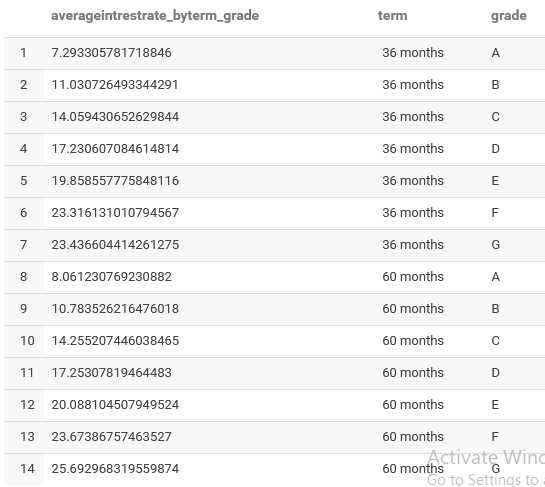


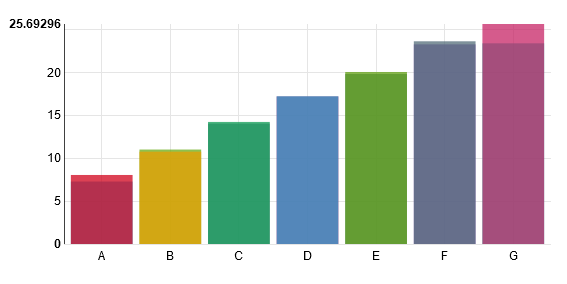
**## Find the historical average interest rates by loan terms and loan grades (also for overall)**

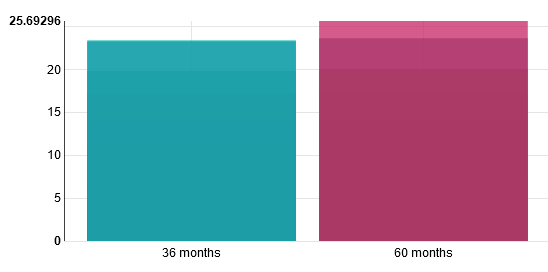
create table avg\_intrst as SELECT term as t\_term, grade as grd, regexp\_replace(int\_rate,'[^0-9.]','') as intrest\_rate from loan\_table\_1;

SELECT avg(intrest\_rate)as averageintrestrate\_byterm\_grade , t\_term as term, grd as grade FROM avg\_intrst GROUP BY t\_term, grd;

SELECT avg(intrest\_rate)as averageintrestrate\_overall from avg\_intrst;









##### ## What is percentage of loans by different loan grades by each year and loan term level (also for overall)

Create table counter as select grade as grade\_1, term as t\_term, substr(issue\_d,5) as Year ,

count(\*) as cnt from loan\_table\_1 group by grade,term, substr(issue\_d,5) ;

Create table cntr as select count(\*) as ctr from loan\_table\_1;

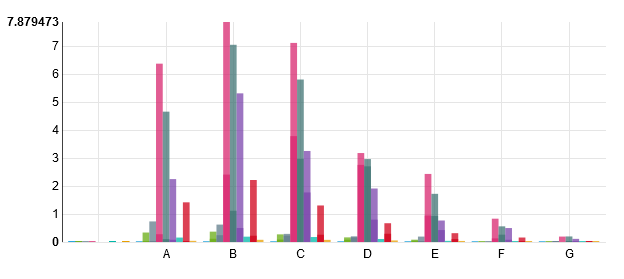
Create table loan\_final as select counter.grade\_1 as grade, counter.t\_term as term, counter.Year as yr, counter.cnt as loan ,cntr.ctr as total\_loan from counter cross join cntr;

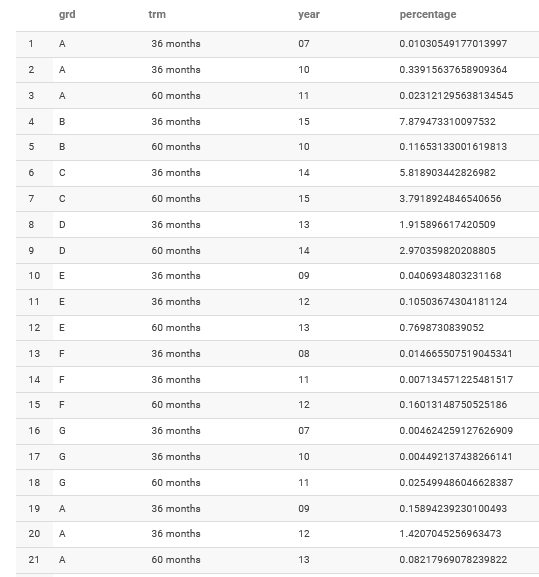
Create table overall\_counter as select substr(issue\_d,5) as Year , count(\*) as cnt from loan\_table\_1 group by substr(issue\_d,5) ;

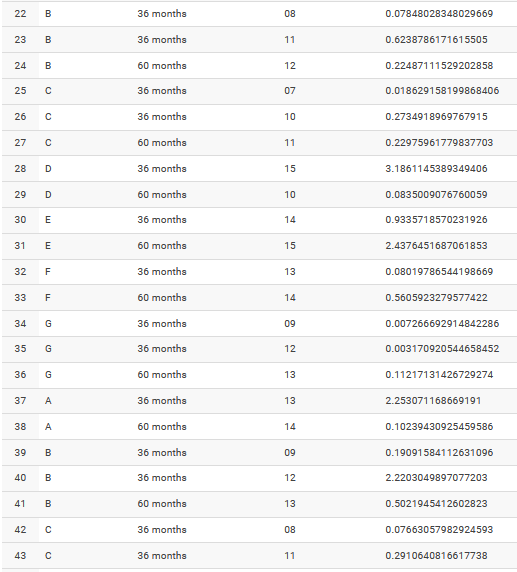
Create table loan\_final\_overall as select overall\_counter.Year as yr, overall\_counter.cnt as loan ,cntr.ctr as total\_loan from overall\_counter cross join cntr;

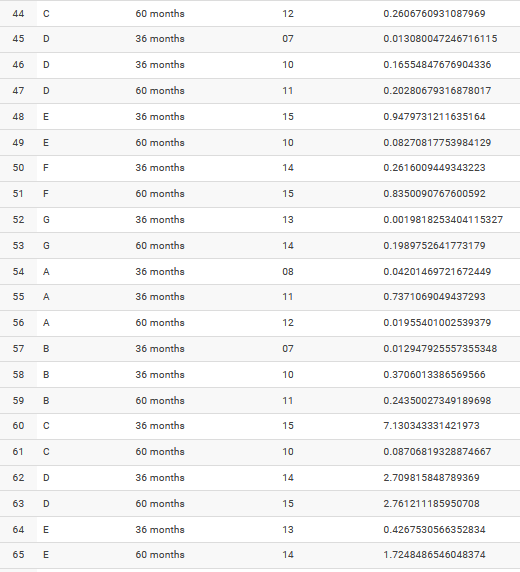
Select grade as GRD, term as TRM, yr as YEAR, 100\*(loan/total\_loan) as Percentage from loan\_final;

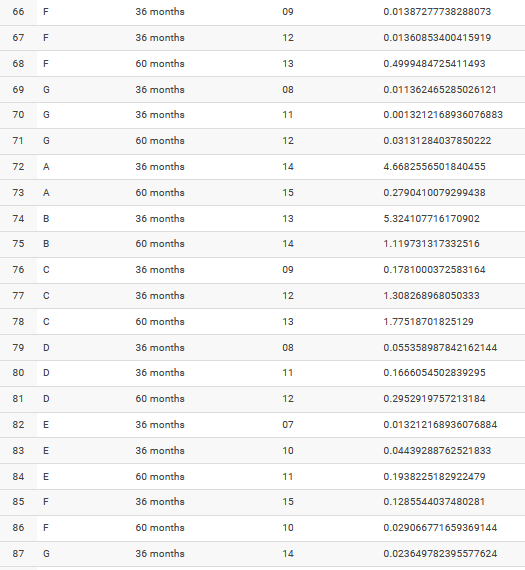
Select yr as YEAR, 100\*(loan/total\_loan) as Overall\_Percentage from loan\_final\_overall;

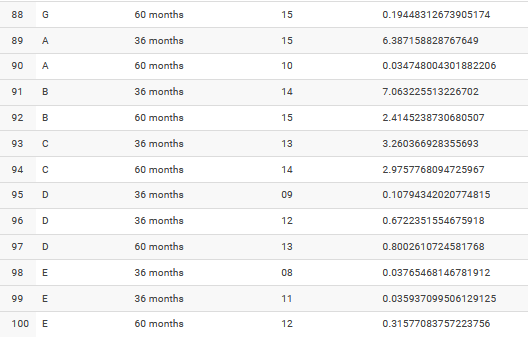


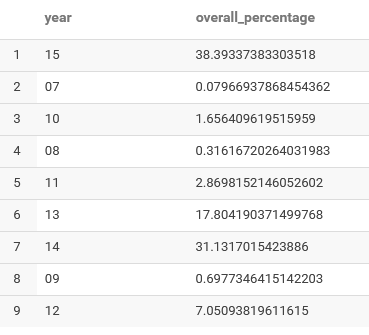












**## What is the loan performance details by different loan grades and overall**

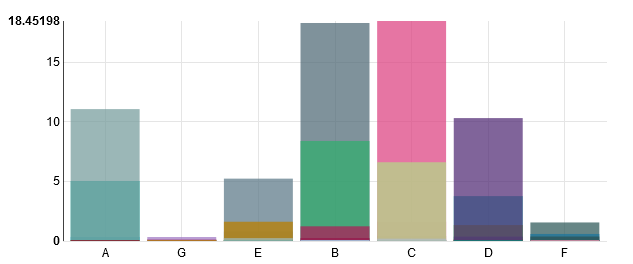
Create table stat as Select loan\_status as status, grade as loan\_grade, count (\*) as countr from loan\_table\_1 group by loan\_status,grade;

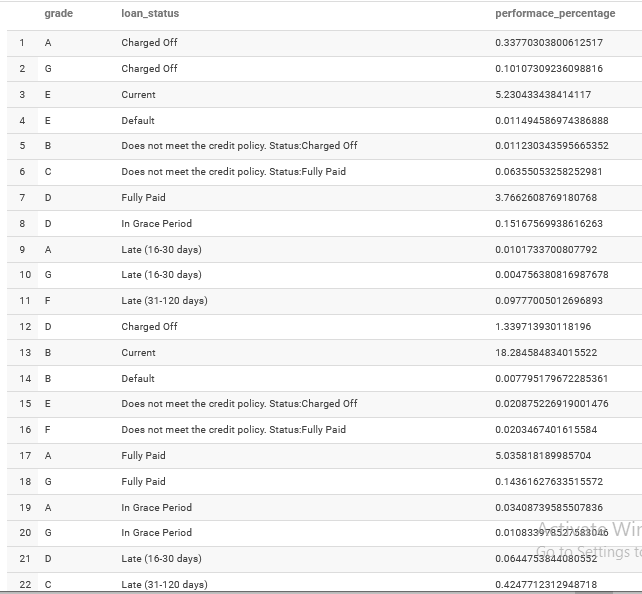
Create table ovr\_stat as select count(\*) as cntr from loan\_table\_1;

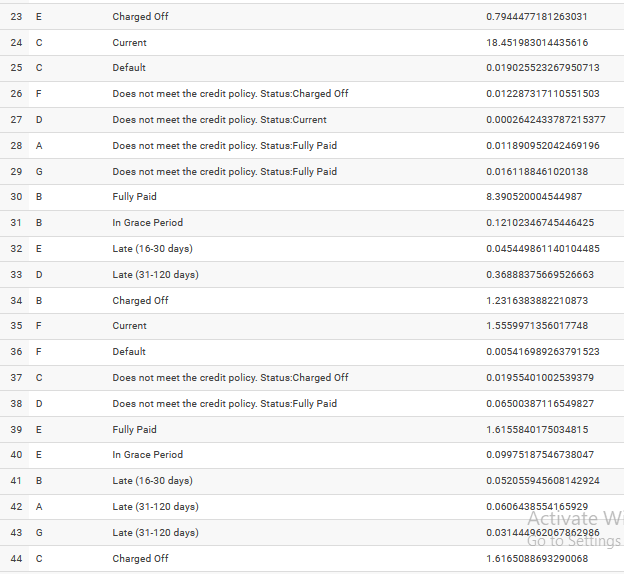
Create table performance as select stat.loan\_grade as grade , stat.status as loan\_status, stat.countr as per\_cnt, ovr\_stat.cntr as t\_cnt

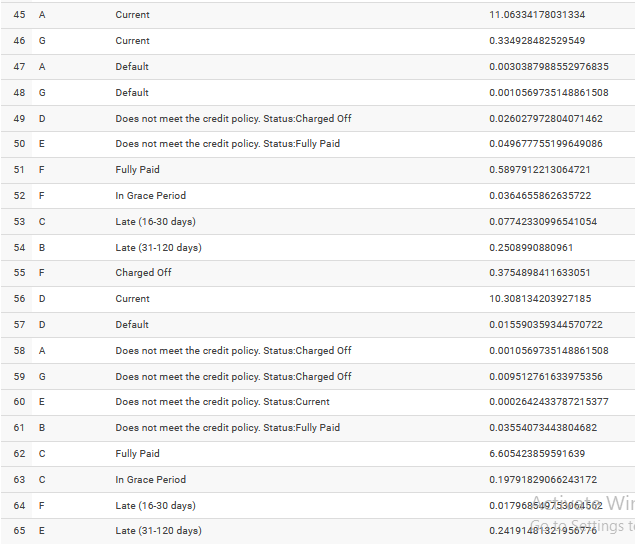
From stat cross join ovr\_stat;

Select grade, loan\_status, 100\*(per\_cnt/t\_cnt) as Performace\_Percentage from performance;









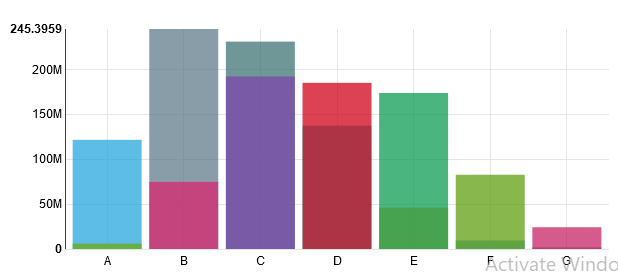
##### ## Find Net Annualized returns by vintage by different loan grades and different loan terms (also for overall

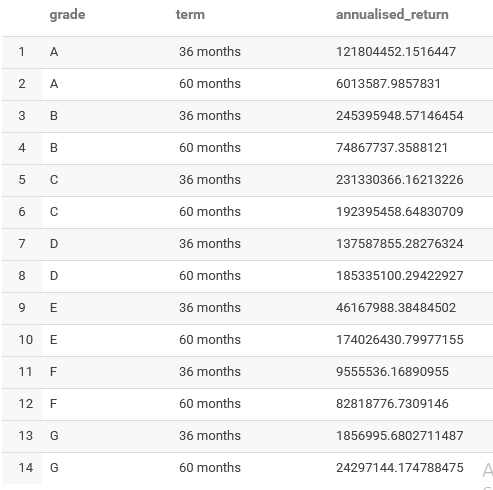
Create table annual as Select cast(regexp\_replace(int\_rate,'[^0-9.]',' ') as float) as roi, loan\_amnt as amnt, grade as grd, term as t\_term from loan\_table\_1;

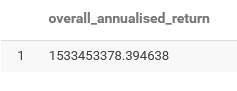
Create table annual\_return as Select grd as grade, t\_term as term, (roi\*amnt)/100 as rtrn from annual;

Select grade, term, sum(rtrn) as Annualised\_Return from annual\_return group by grade,term;

Select sum(rtrn) as Overall\_Annualised\_Return from annual\_return;







**## What is loan status migration over 9 months Net Chargeoffs:120+days delinquency)**

create table nt as select loan\_status as status, count(loan\_status) as cnt from loan\_table\_1

group by loan\_status

having loan\_status in ( 'In Grace Period', 'Late (16-30 days)', 'Late (31-120 days)', 'Default');

Create table charged\_off as Select count(loan\_status) as cnt1 from loan\_table\_1

GROUP BY loan\_status

Having loan\_status='Charged Off';

Create table migrate\_status as select nt.status as status, nt.cnt as count, charged\_off.cnt1 as count1 from nt cross join charged\_off;

Select status, 100\*(count/count1) as Migrtn\_Status from migrate\_status;

