In Hive Editor:

* create table companies\_review(Indexnumber int,Company String,location1 String, dates string, Job\_title String,Summary String,Pross string, Cons string, Rating int, wbstars int, cvstars int, costars int, cbstars int, smstars int) row format delimited fields terminated by ',' stored as textfile TBLPROPERTIES ("skip.header.line.count"="1");
* LOAD DATA INPATH '/user/s\_mujahid4\_tcs/employee\_review.csv' OVERWRITE INTO TABLE companies\_review
* Create table emprev as SELECT IndexNumber, company, location1, split(dates, ';')[1] as Yr, job\_title, Summary, Pross, cons, Rating, CASE when wbstars IS NULL THEN 3 else wbstars END AS wb, CASE when cvstars IS NULL THEN 4 else cvstars END AS cv, CASE when costars IS NULL THEN 4 else costars END AS co, CASE when cbstars IS NULL THEN 4 else cbstars END as cb, CASE when smstars IS NULL THEN 3 else smstars END as sm, CASE when location1 is null then null when location1 like "%;%" then "USA"

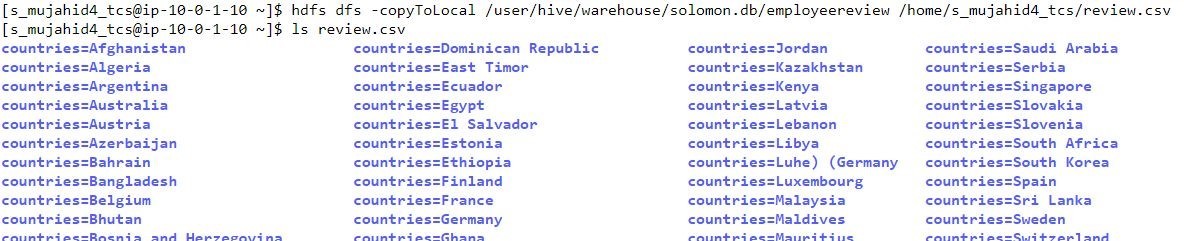
when location1 like "%(%)%" then substr(location1,(instr(location1,'(')+1),length(substr(location1,(instr(location1,'(')+1)))-1) END as Cntry FROM companies\_review;

* set hive.exec.dynamic.partition.mode=nonstrict
* create table employeereview(Indexnumber int,Company String,location1 String, country string, Yr string, Job\_title String, Summary String, Pros string, Cons string, Rating int, wb int, cv int, co int, cb int, sm int) partitioned by (Countries string) clustered by (Yr) Into 10 buckets;
* INSERT into employeereview Partition (countries) SELECT IndexNumber, company, location1, cntry, Yr, job\_title, Summary, Pross, cons, Rating, wb, cv, co, cb, sm, cntry from emprev;
* SELECT \* from employeereview;

Write the final relation schema to review.csv file in your HDFS home directory. In web console:

$ hdfs dfs -copyToLocal /user/hive/warehouse/solomon.db/employeereview /home/s\_mujahid4\_tcs/review.csv

$ ls review.csv

**Screenshot:**

Using the over-all rating fields display trend:

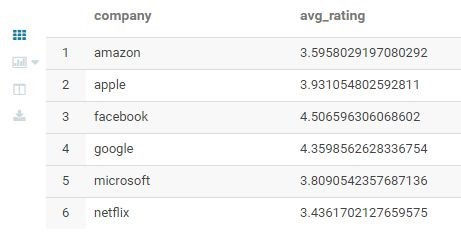
1. Globally by company

Identify trends at 25%, 50%, 75%

1. Globally by company per year Identify trends at 25%, 50%, 75%
2. By company by country (Identify trends for each company by country Identify trends at 25%, 50%, 75%

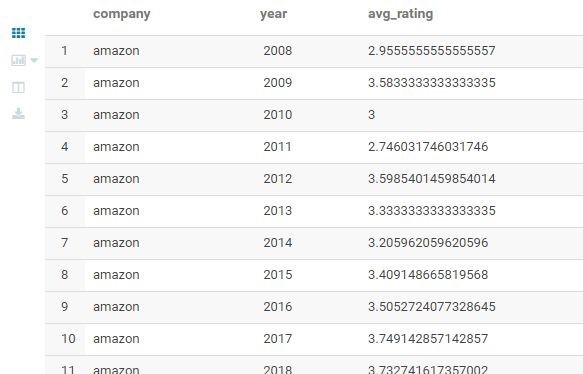
For 25 percent:

* create table emp\_review\_25percent as select \* from employeereview where rand() <= 0.25 cluster by rand();
* select company,avg(rating) as Avg\_Rating from emp\_review\_25percent group by company;

**Screenshot:**

* select company,yr as Year,avg(rating) as Avg\_Rating from emp\_review\_25percent group by company,yr;

**Screenshot:**

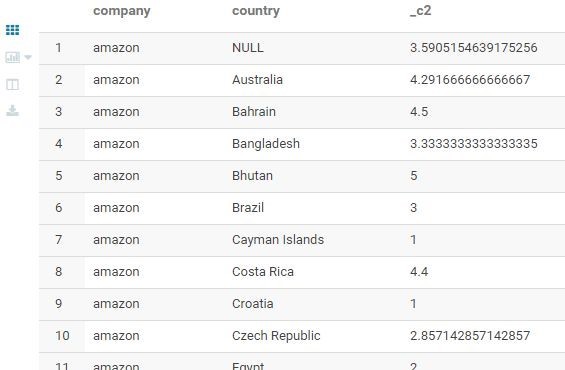


* select company,country,avg(rating) as Avg\_Rating from emp\_review\_25percent group by company,country; Screenshot:



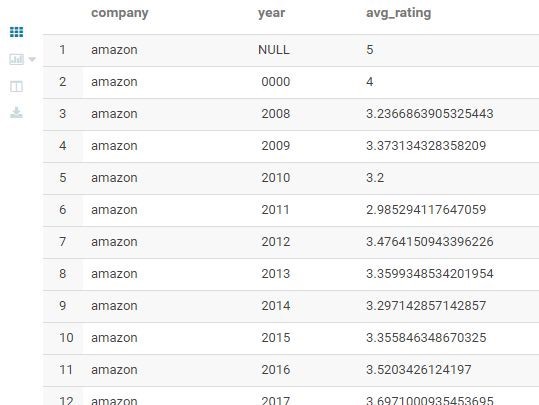
For 50 percent:

* create table emp\_review\_50percent as select \* from employeereview where rand() <= 0.50 cluster by rand();
* select company,avg(rating) as Avg\_Rating from emp\_review\_50percent group by company;
* select company,yr as year,avg(rating) as Avg\_Rating from emp\_review\_50percent group by company,yr
* select company,country,avg(rating) from emp\_review\_50percent group by company,country;



For 75 percent:

* create table emp\_review\_75percent as select \* from employeereview where rand() <= 0.75 cluster by rand();
* select company,avg(rating) as Avg\_Rating from emp\_review\_75percent group by company;
* select company,yr as year,avg(rating) as Avg\_Rating from emp\_review\_75percent group by company,yr;

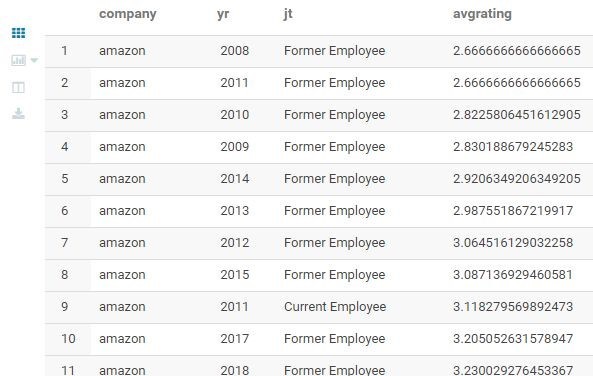


* select company,country,avg(rating) as Avg\_Rating from emp\_review\_75percent group by company,country;

# Display the impact of employee status on rating a company using the overall-ratings field by the company by year.

Show who are more likely to rate better between former and current employees

* SELECT company, yr, SPLlT(job\_title, '-')[0] as jt, avg(rating) as avgrating from employeereview GROUP BY company, yr, SPLlT(job\_title, '-')[0] ORDER BY company, avgrating



The overall rating field trend for each company over the years

* select company,yr, percentile(rating,array(0.25,0.5,0.75)) as yearly\_trend from employeereview where yr > 0 group by company,yr order by company,yr



# Display the impact of job role on rating a company using the overall-ratings field by the company by year.

* SELECT company, yr, SPLlT(job\_title, '-')[1] as jt, round(avg(rating),2) as avgrating from employeereview GROUP BY company, yr, SPLlT(job\_title, '-')[1] ORDER BY company, avgrating

# Display the relationship between the overall rating score vs. the rest of the rating field scores by company. Also, document your findings.

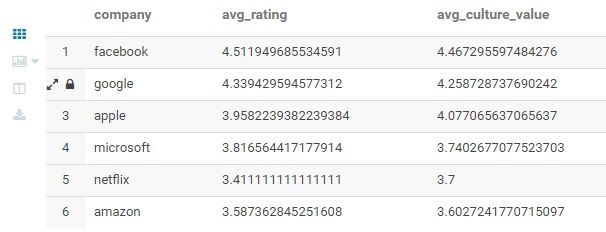
**Complete ratings**

* select company,round(avg(Rating),1) as Overall\_Rating, round(avg(wb),1) Work\_Balance,round(avg(cv),1) Culture\_Value,round(avg(co),1) Career\_Opportunities,round(avg(cb),1) Comp\_Benefit,round(avg(sm),1) Senior\_Management from employeereview group by company order by Overall\_Rating



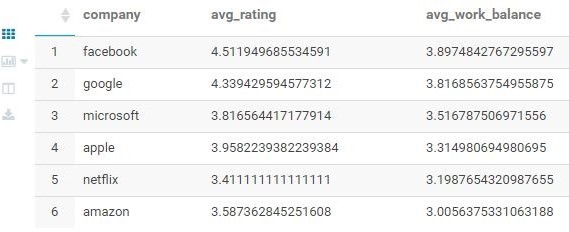
Overall vs Culture Values

* SELECT company, avg(rating) as Avg\_Rating, avg(cv) as Avg\_Culture\_Value FROM employeereview GROUP BY company ORDER BY Avg\_Culture\_Value DESC



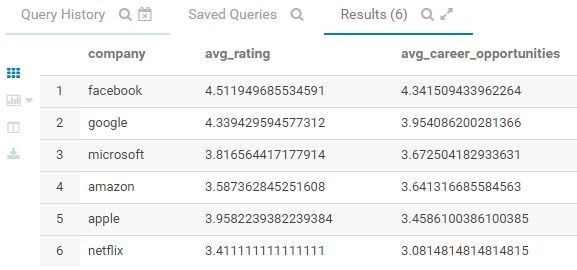
Overall vs Work balance

* SELECT company, avg(rating) as Avg\_Rating, avg(wb) as Avg\_Work\_Balance FROM employeereview GROUP BY company ORDER BY Avg\_Work\_Balance DESC



Overall vs Career Opportunities

* SELECT company, avg(rating) as Avg\_Rating, avg(co) as Avg\_Career\_Opportunities FROM employeereview GROUP BY company ORDER BY Avg\_Career\_Opportunities DESC



Overall vs Compensation benefit

* SELECT company, avg(rating) as Avg\_Rating, avg(cb) as Avg\_Comp\_Benefit FROM employeereview GROUP BY company ORDER BY Avg\_Comp\_Benefit DESC



Overall vs Senior Management

* SELECT company, avg(rating) as Avg\_Rating, avg(sm) as Avg\_Senior\_Management FROM employeereview GROUP BY company ORDER BY Avg\_Senior\_Management DESC

Document your findings for the following:

* 1. Which corporation is worth working for

From the above queries we can confirm that facebook has best ratings overall and is the corporation worth working for.

* 1. Classification of satisfied or unsatisfied employees

create table sat as select company,count(company) as sa from employeereview where rating>=4 group by company;

create table unsat as select company,count(company) as un from employeereview where rating<4 group by company;

select sat.company,round(((sat.sa/(unsat.un+sat.sa))\*100),2) as

satisfied,round((100-((sat.sa/(unsat.un+sat.sa))\*100)),2) as unsatisfied from sat,unsat where sat.company=unsat.company order by satisfied DESC;



The above data provides the classification of satisfies and unsatisfied employees.