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
```

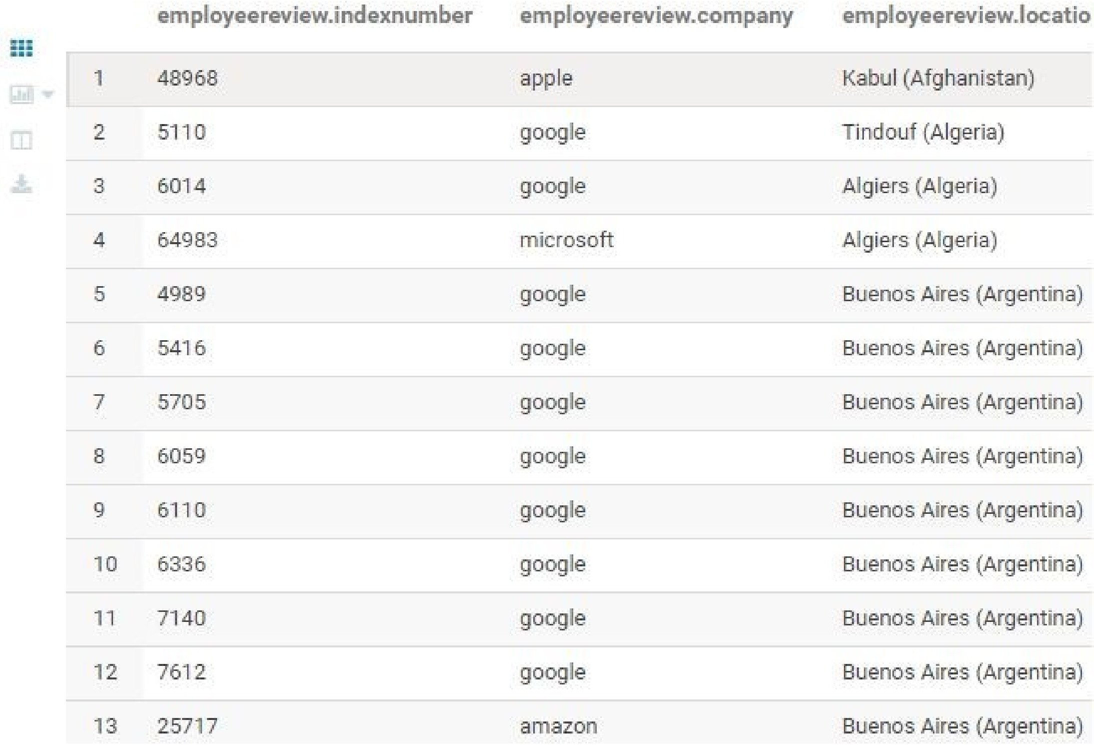
```
> Create table emprev as SELECT IndexNumber, company, location1, split(dates, ';')[1] as Yr, job_title,
Summary, Pross, cons, Rating,  CASE when wbstars IS NULL THEN  3 else wbstars END AS wb, CASE when
cvstars IS NULL THEN  4 else cvstars  END AS cv, CASE when costars IS NULL THEN  4 else costars 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;
```



	employeereview.indexnumber	employeereview.company	employeereview.locatio
1	48968	apple	Kabul (Afghanistan)
2	5110	google	Tindouf (Algeria)
3	6014	google	Algiers (Algeria)
4	64983	microsoft	Algiers (Algeria)
5	4989	google	Buenos Aires (Argentina)
6	5416	google	Buenos Aires (Argentina)
7	5705	google	Buenos Aires (Argentina)
8	6059	google	Buenos Aires (Argentina)
9	6110	google	Buenos Aires (Argentina)
10	6336	google	Buenos Aires (Argentina)
11	7140	google	Buenos Aires (Argentina)
12	7612	google	Buenos Aires (Argentina)
13	25717	amazon	Buenos Aires (Argentina)

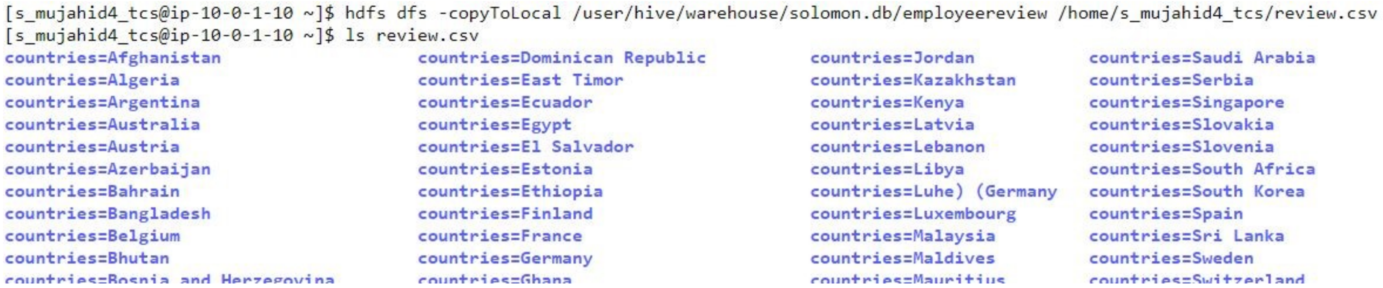
Write the final relation schema to review.csv file in your HDFS home directory.

In web console:

$ hdfs dfs -copyToLocal /user/hive/warehouse/solomon.db/employeereview /home/s_mujahid4_tcs/review.csv

$ ls review.csv

<u>Screenshot:</u>



```
[s_mujahid4_tcs@ip-10-0-1-10 ~]$ hdfs dfs -copyToLocal /user/hive/warehouse/solomon.db/employeereview /home/s_mujahid4_tcs/review.csv
[s_mujahid4_tcs@ip-10-0-1-10 ~]$ ls review.csv
countries=Afghanistan                countries=Dominican Republic        countries=Jordan            countries=Saudi Arabia
countries=Algeria                    countries=East Timor                countries=Kazakhstan        countries=Serbia
countries=Argentina                  countries=Ecuador                   countries=Kenya             countries=Singapore
countries=Australia                  countries=Egypt                     countries=Latvia            countries=Slovakia
countries=Austria                    countries=El Salvador               countries=Lebanon           countries=Slovenia
countries=Azerbaijan                 countries=Estonia                   countries=Libya             countries=South Africa
countries=Bahrain                    countries=Ethiopia                  countries=Luhe) (Germany    countries=South Korea
countries=Bangladesh                 countries=Finland                   countries=Luxembourg        countries=Spain
countries=Belgium                    countries=France                    countries=Malaysia          countries=Sri Lanka
countries=Bhutan                     countries=Germany                   countries=Maldives          countries=Sweden
countries=Bosnia and Herzegovina     countries=Ghana                     countries=Mauritius         countries=Switzerland
```

Using the over-all rating fields display trend:

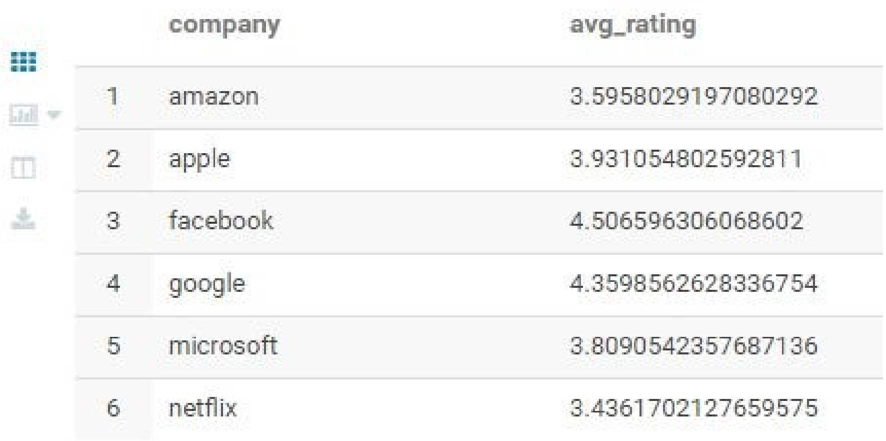
1. Globally by company
   Identify trends at 25%, 50%, 75%
2. Globally by company per year
   Identify trends at 25%, 50%, 75%
3. By company by country (Identify trends for each company by country
   Identify trends at 25%, 50%, 75%

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;

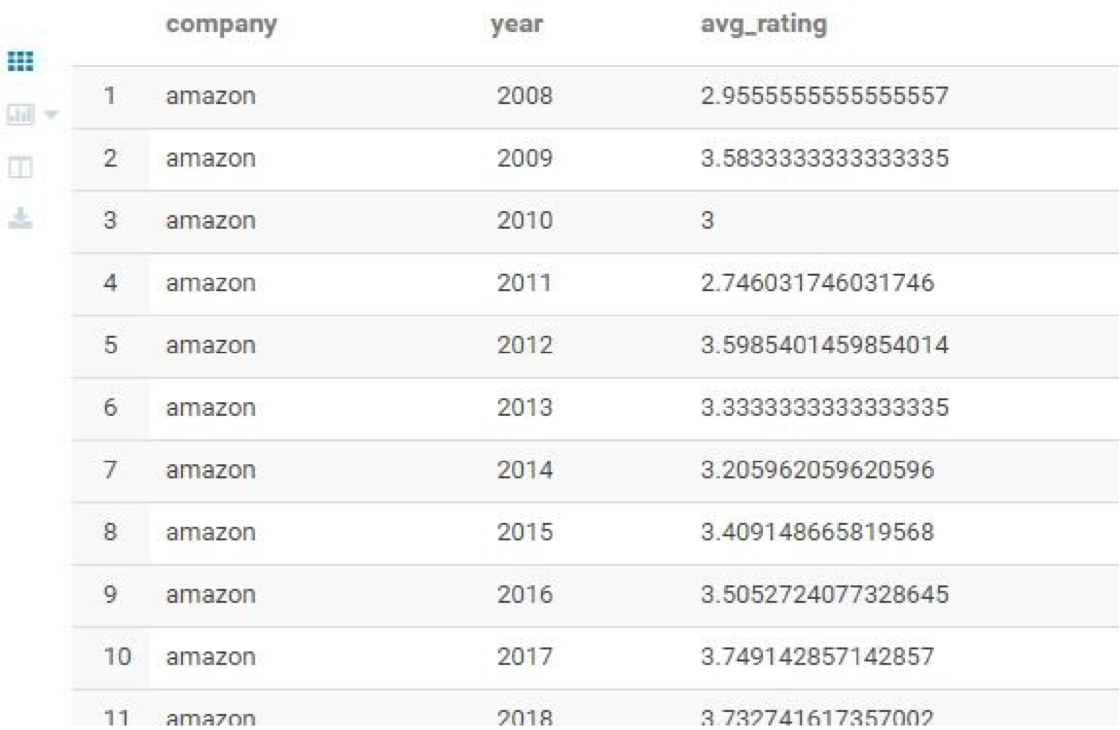
<u>Screenshot:</u>



	company	avg_rating
1	amazon	3.5958029197080292
2	apple	3.931054802592811
3	facebook	4.506596306068602
4	google	4.3598562628336754
5	microsoft	3.8090542357687136
6	netflix	3.4361702127659575

> select company,yr as Year,avg(rating) as Avg_Rating from emp_review_25percent group by company,yr;

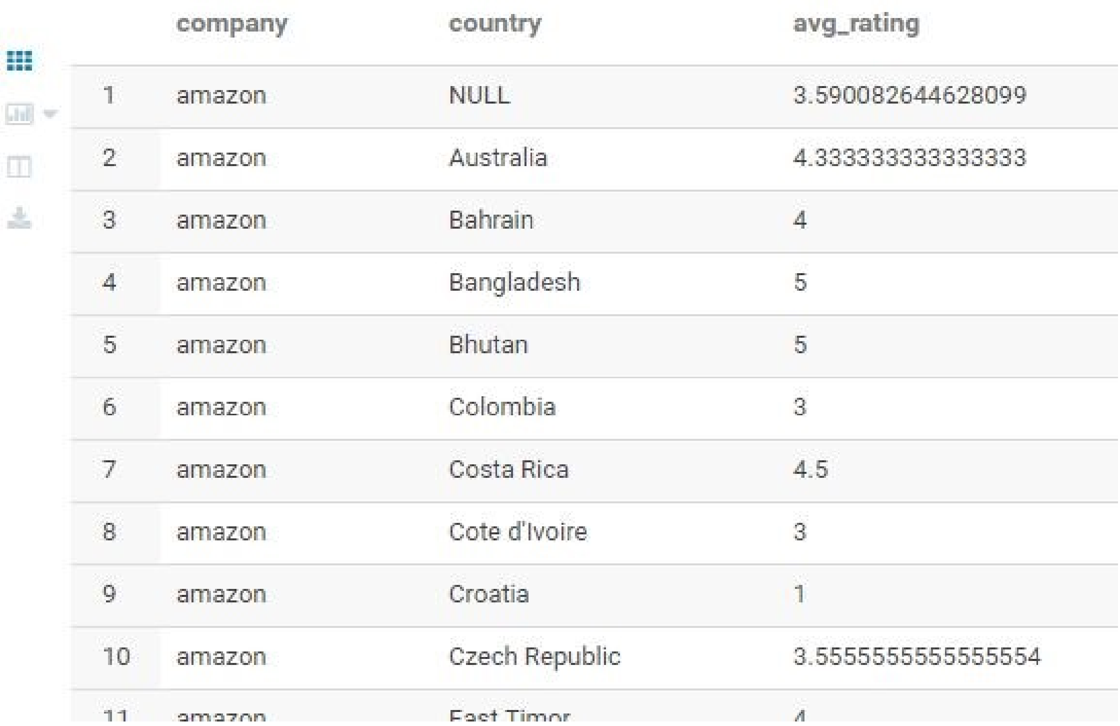
<u>Screenshot:</u>



	company	year	avg_rating
1	amazon	2008	2.9555555555555557
2	amazon	2009	3.5833333333333335
3	amazon	2010	3
4	amazon	2011	2.746031746031746
5	amazon	2012	3.5985401459854014
6	amazon	2013	3.3333333333333335
7	amazon	2014	3.205962059620596
8	amazon	2015	3.409148665819568
9	amazon	2016	3.5052724077328645
10	amazon	2017	3.749142857142857
11	amazon	2018	3.732741617357002

> select company,country,avg(rating) as Avg_Rating from emp_review_25percent group by company,country;

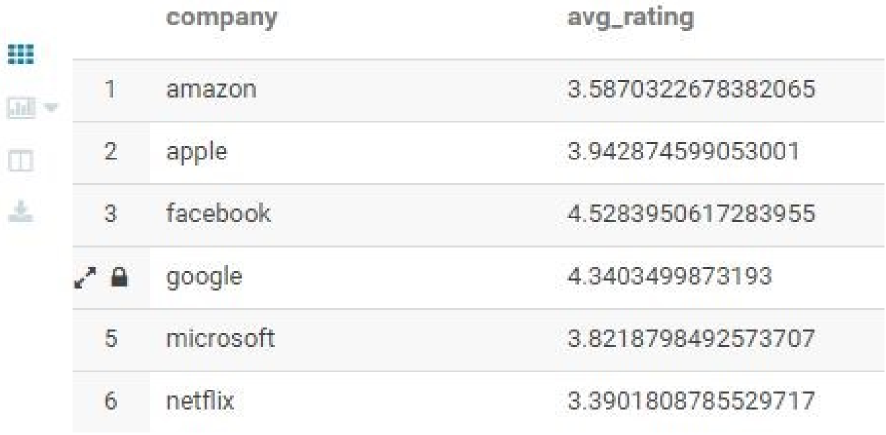
Screenshot:



	company	country	avg_rating
1	amazon	NULL	3.590082644628099
2	amazon	Australia	4.333333333333333
3	amazon	Bahrain	4
4	amazon	Bangladesh	5
5	amazon	Bhutan	5
6	amazon	Colombia	3
7	amazon	Costa Rica	4.5
8	amazon	Cote d'Ivoire	3
9	amazon	Croatia	1
10	amazon	Czech Republic	3.5555555555555554
11	amazon	East Timor	4

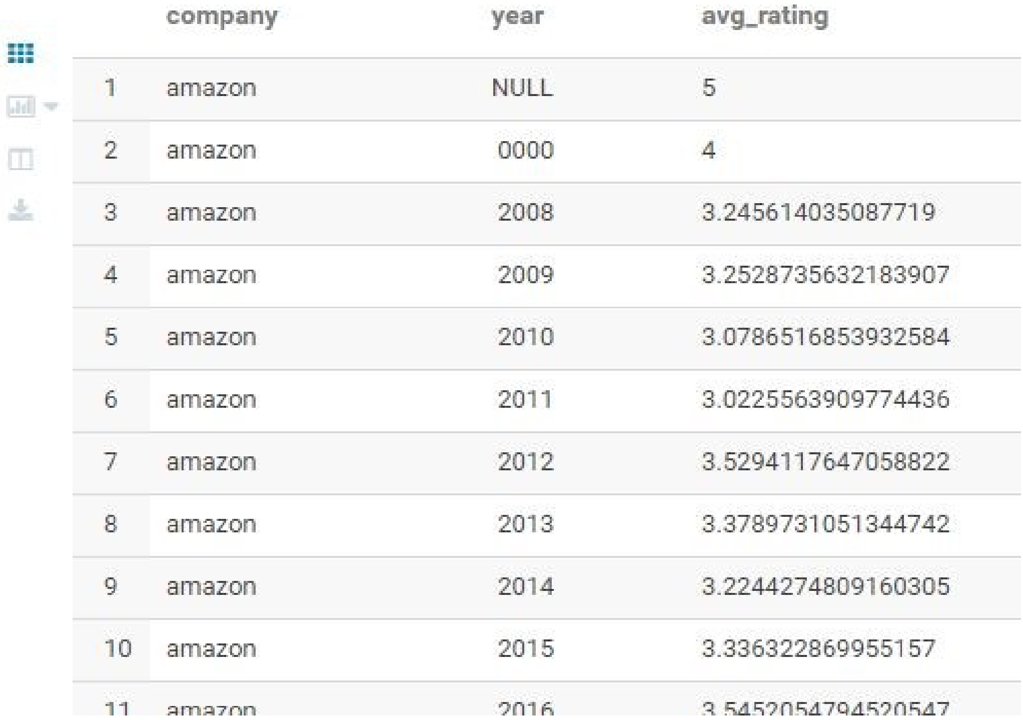
For 50 percent:

```
> 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;
```



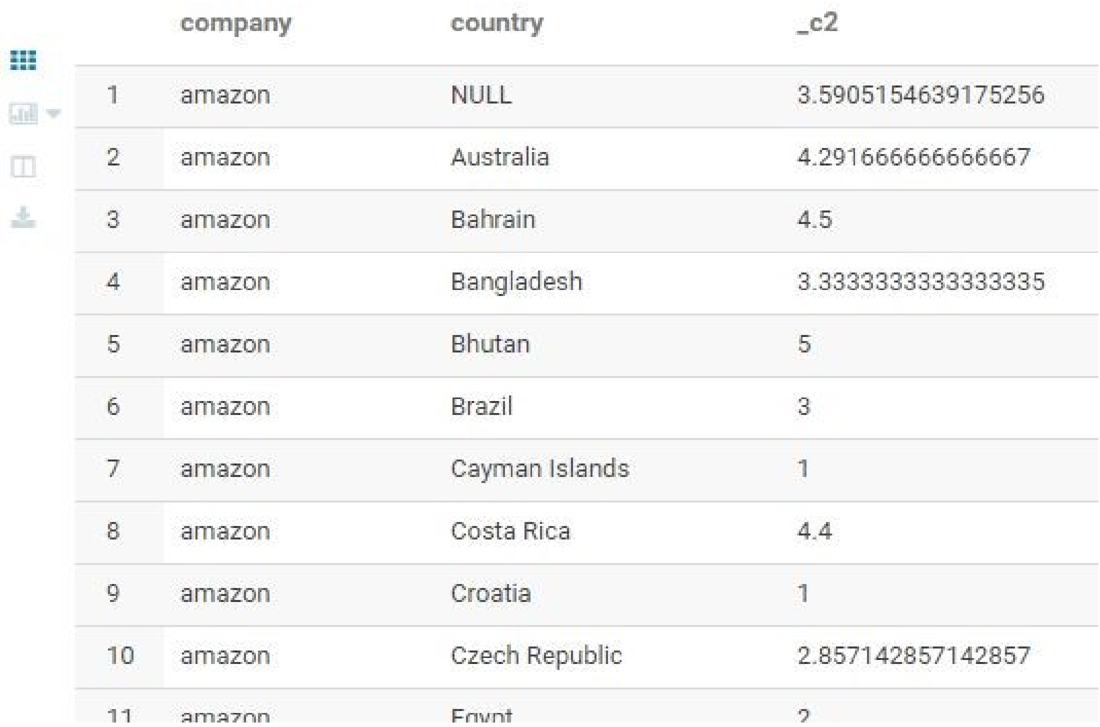
	company	avg_rating
1	amazon	3.5870322678382065
2	apple	3.942874599053001
3	facebook	4.5283950617283955
	google	4.3403499873193
5	microsoft	3.8218798492573707
6	netflix	3.3901808785529717

```
> select company,yr as year,avg(rating) as Avg_Rating from emp_review_50percent group by company,yr
```



	company	year	avg_rating
1	amazon	NULL	5
2	amazon	0000	4
3	amazon	2008	3.245614035087719
4	amazon	2009	3.2528735632183907
5	amazon	2010	3.0786516853932584
6	amazon	2011	3.0225563909774436
7	amazon	2012	3.5294117647058822
8	amazon	2013	3.3789731051344742
9	amazon	2014	3.2244274809160305
10	amazon	2015	3.336322869955157
11	amazon	2016	3.545205479452054**7**

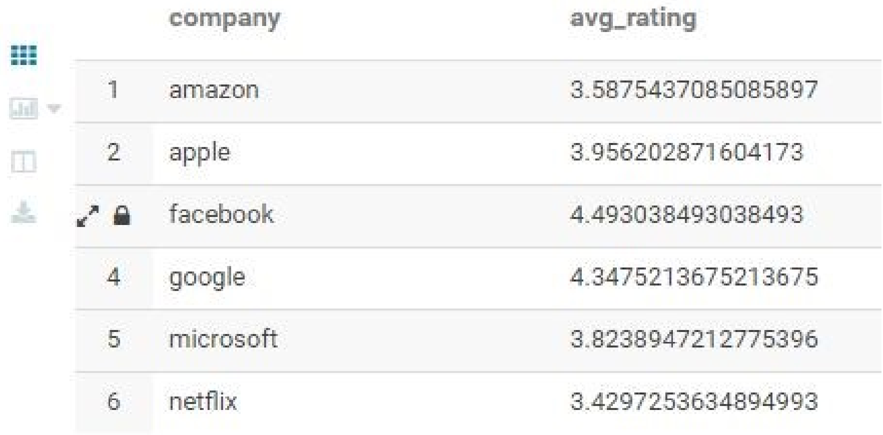
```
> select company,country,avg(rating)  from emp_review_50percent group by company,country;
```



	company	country	_c2
1	amazon	NULL	3.5905154639175256
2	amazon	Australia	4.291666666666667
3	amazon	Bahrain	4.5
4	amazon	Bangladesh	3.3333333333333335
5	amazon	Bhutan	5
6	amazon	Brazil	3
7	amazon	Cayman Islands	1
8	amazon	Costa Rica	4.4
9	amazon	Croatia	1
10	amazon	Czech Republic	2.857142857142857
11	amazon	Egypt	2

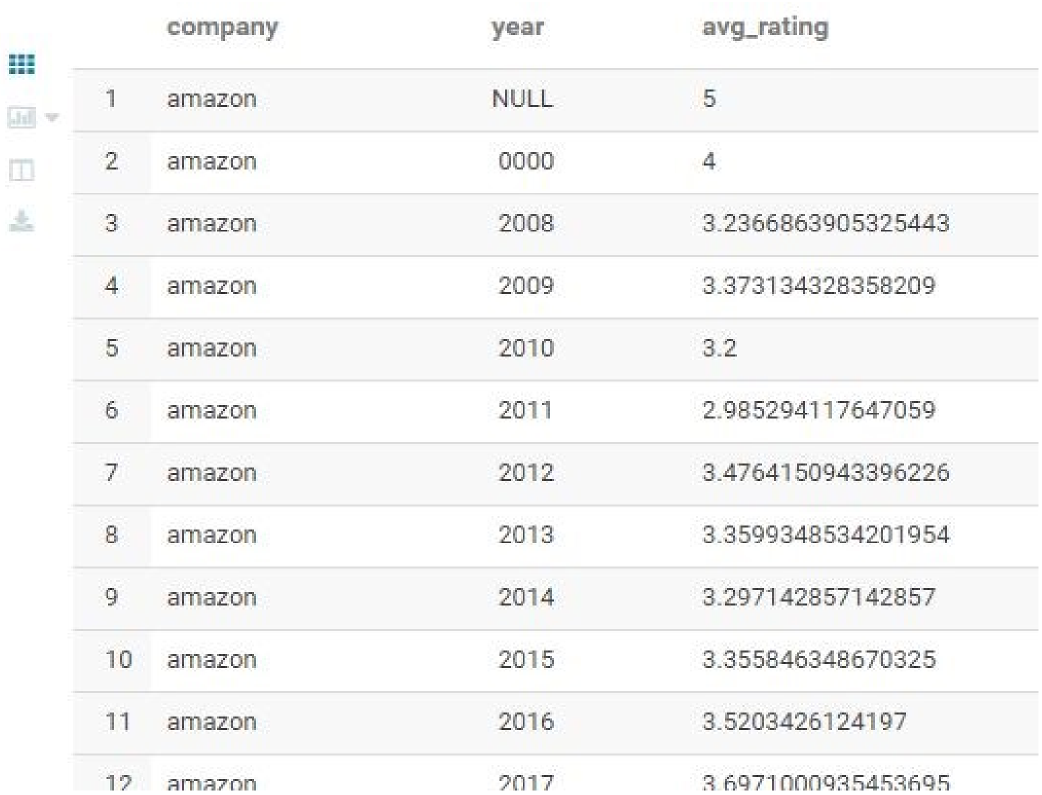
For 75 percent:

```
> 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;
```



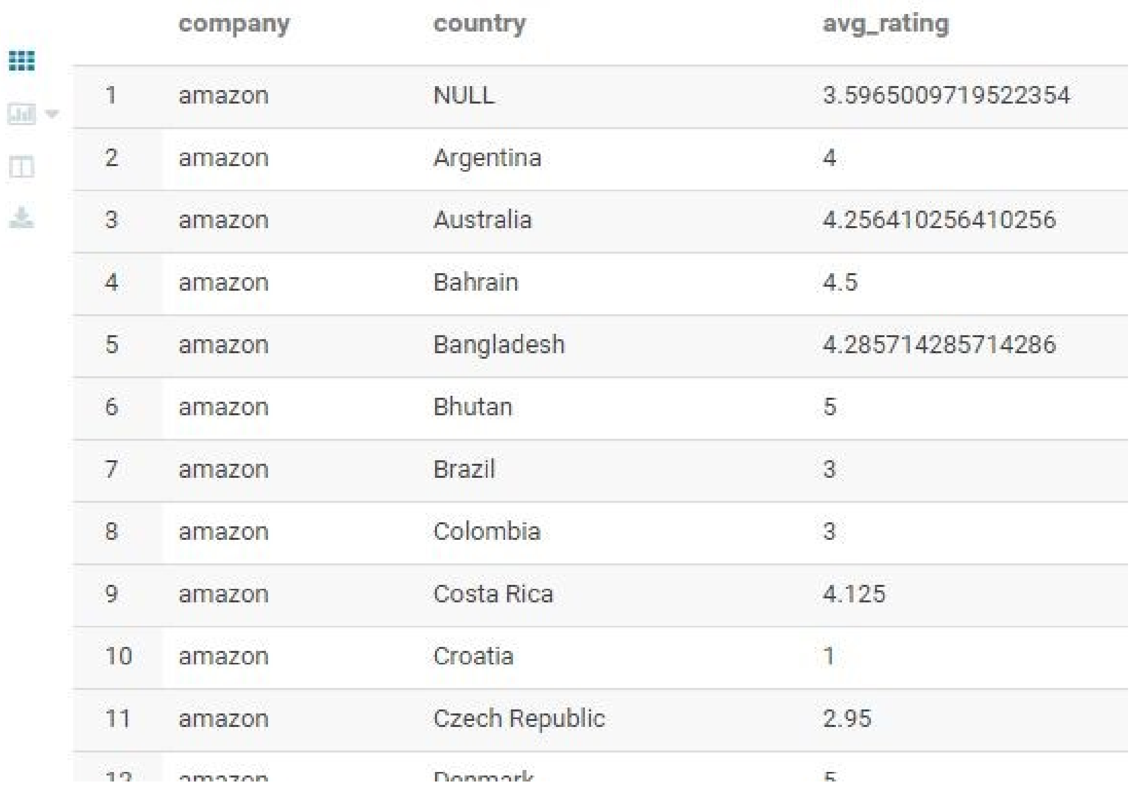
	company	avg_rating
1	amazon	3.5875437085085897
2	apple	3.956202871604173
	facebook	4.493038493038493
4	google	4.3475213675213675
5	microsoft	3.8238947212775396
6	netflix	3.4297253634894993

```
> select company,yr as year,avg(rating) as Avg_Rating from emp_review_75percent group by company,yr;
```



	company	year	avg_rating
1	amazon	NULL	5
2	amazon	0000	4
3	amazon	2008	3.2366863905325443
4	amazon	2009	3.373134328358209
5	amazon	2010	3.2
6	amazon	2011	2.985294117647059
7	amazon	2012	3.4764150943396226
8	amazon	2013	3.3599348534201954
9	amazon	2014	3.297142857142857
10	amazon	2015	3.355846348670325
11	amazon	2016	3.5203426124197
12	amazon	2017	3.6971000935453695

```
> select company,country,avg(rating) as Avg_Rating from emp_review_75percent group by company,country;
```

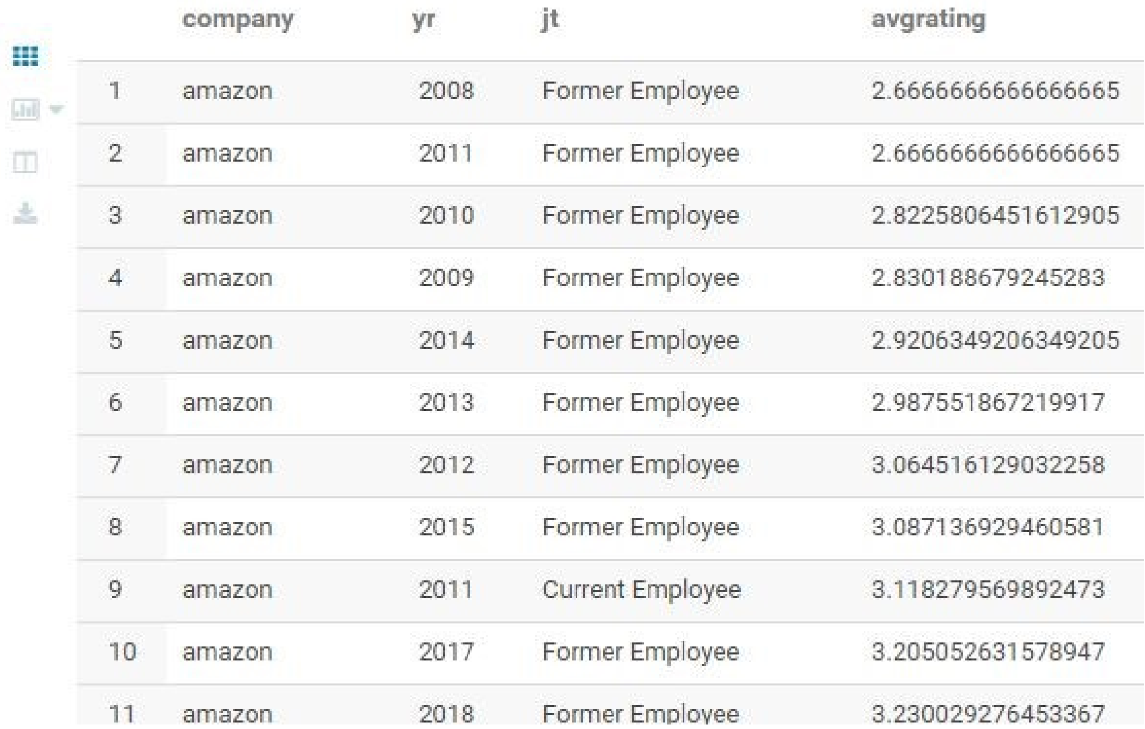


	company	country	avg_rating
1	amazon	NULL	3.5965009719522354
2	amazon	Argentina	4
3	amazon	Australia	4.256410256410256
4	amazon	Bahrain	4.5
5	amazon	Bangladesh	4.285714285714286
6	amazon	Bhutan	5
7	amazon	Brazil	3
8	amazon	Colombia	3
9	amazon	Costa Rica	4.125
10	amazon	Croatia	1
11	amazon	Czech Republic	2.95
12	amazon	Denmark	5

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

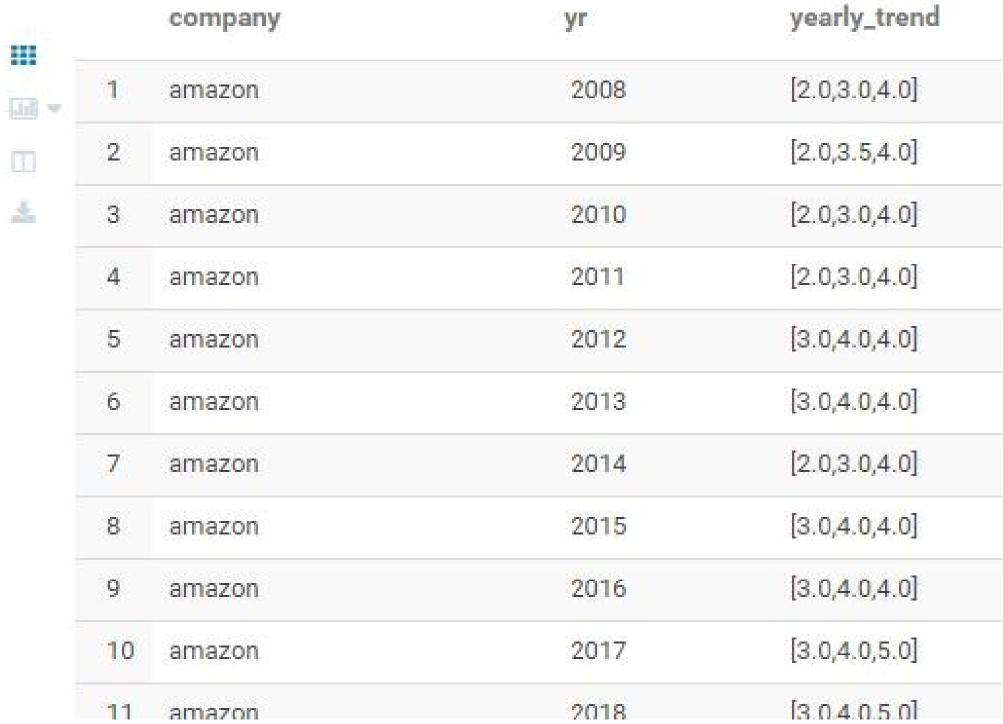
> SELECT company, yr, SPLIT(job_title, '-')[0] as jt, avg(rating) as avgrating from employeereview GROUP BY company, yr, SPLIT(job_title, '-')[0] ORDER BY company, avgrating



	company	yr	jt	avgrating
1	amazon	2008	Former Employee	2.6666666666666665
2	amazon	2011	Former Employee	2.6666666666666665
3	amazon	2010	Former Employee	2.8225806451612905
4	amazon	2009	Former Employee	2.830188679245283
5	amazon	2014	Former Employee	2.9206349206349205
6	amazon	2013	Former Employee	2.987551867219917
7	amazon	2012	Former Employee	3.064516129032258
8	amazon	2015	Former Employee	3.087136929460581
9	amazon	2011	Current Employee	3.118279569892473
10	amazon	2017	Former Employee	3.205052631578947
11	amazon	2018	Former Employee	3.230029276453367

The overall rating field trend for each company over the years

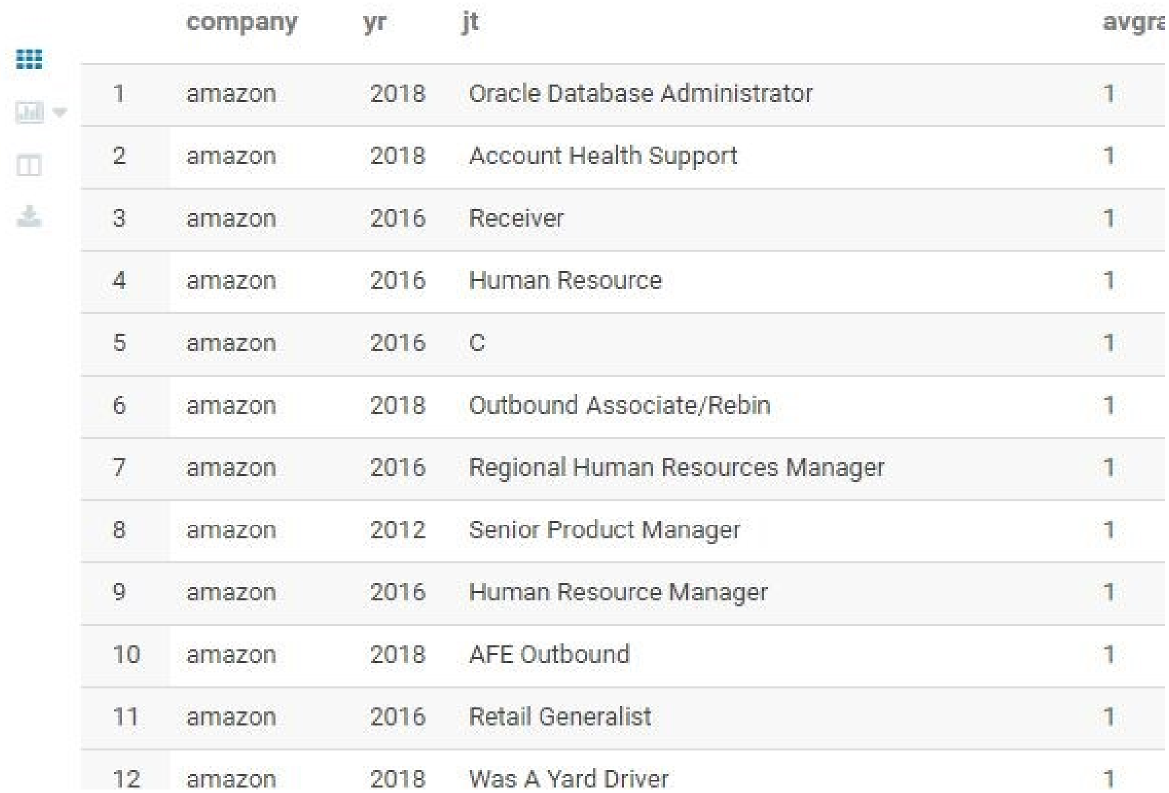
> select company,yr, percentile(rating,array(0.25,0.5,0.75)) as yearly_trend from employeereview where yr > 0 group by company,yr order by company,yr



	company	yr	yearly_trend
1	amazon	2008	[2.0,3.0,4.0]
2	amazon	2009	[2.0,3.5,4.0]
3	amazon	2010	[2.0,3.0,4.0]
4	amazon	2011	[2.0,3.0,4.0]
5	amazon	2012	[3.0,4.0,4.0]
6	amazon	2013	[3.0,4.0,4.0]
7	amazon	2014	[2.0,3.0,4.0]
8	amazon	2015	[3.0,4.0,4.0]
9	amazon	2016	[3.0,4.0,4.0]
10	amazon	2017	[3.0,4.0,5.0]
11	amazon	2018	[3.0,4.0,5.0]

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

> SELECT company, yr, SPLIT(job_title, '-')[1] as jt, round(avg(rating),2) as avgrating from employeereview GROUP BY company, yr, SPLIT(job_title, '-')[1] ORDER BY company, avgrating

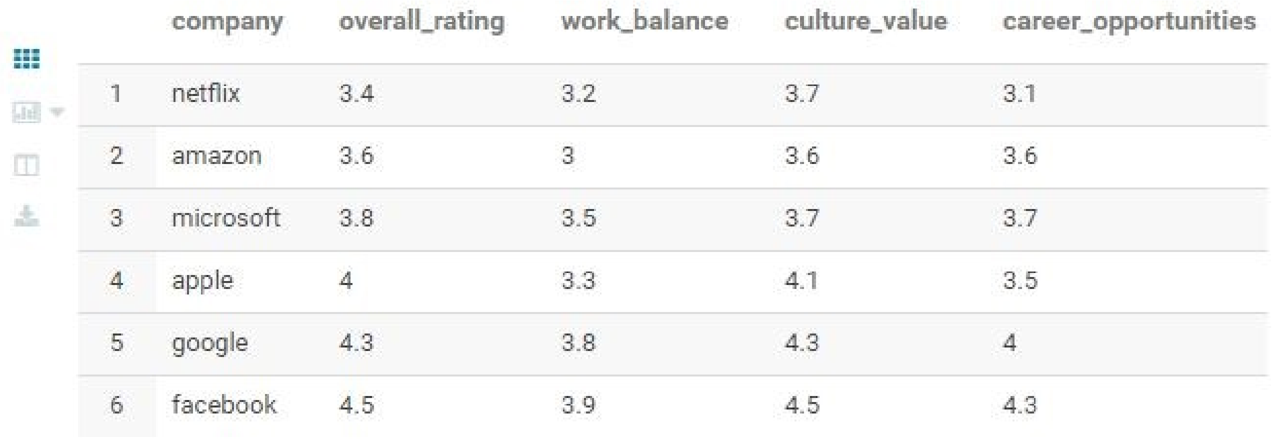


	company	yr	jt	avgra
1	amazon	2018	Oracle Database Administrator	1
2	amazon	2018	Account Health Support	1
3	amazon	2016	Receiver	1
4	amazon	2016	Human Resource	1
5	amazon	2016	C	1
6	amazon	2018	Outbound Associate/Rebin	1
7	amazon	2016	Regional Human Resources Manager	1
8	amazon	2012	Senior Product Manager	1
9	amazon	2016	Human Resource Manager	1
10	amazon	2018	AFE Outbound	1
11	amazon	2016	Retail Generalist	1
12	amazon	2018	Was A Yard Driver	1

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

Complete ratings

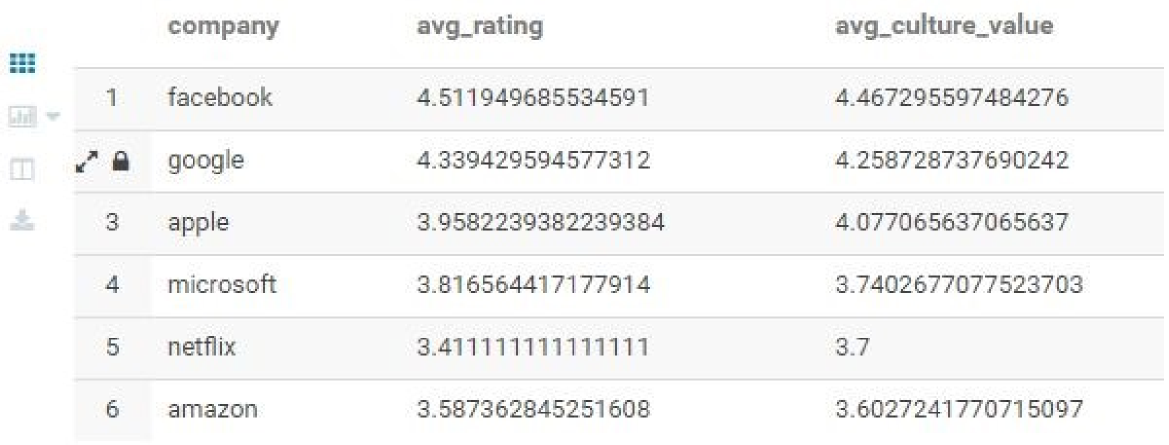
> select company,round(avg(Rating),1) as Overall_Rating, round(avg(wb),1) Work_Balance,round(avg(cv),1) Culture_Value,round(avg(co),1) Career_Opportunities,round(avg(cb),1) Comp_Benefit,round(avg(sm),1) Senior_Management from employeereview group by company order by Overall_Rating



	company	overall_rating	work_balance	culture_value	career_opportunities
1	netflix	3.4	3.2	3.7	3.1
2	amazon	3.6	3	3.6	3.6
3	microsoft	3.8	3.5	3.7	3.7
4	apple	4	3.3	4.1	3.5
5	google	4.3	3.8	4.3	4
6	facebook	4.5	3.9	4.5	4.3

Overall vs Culture Values

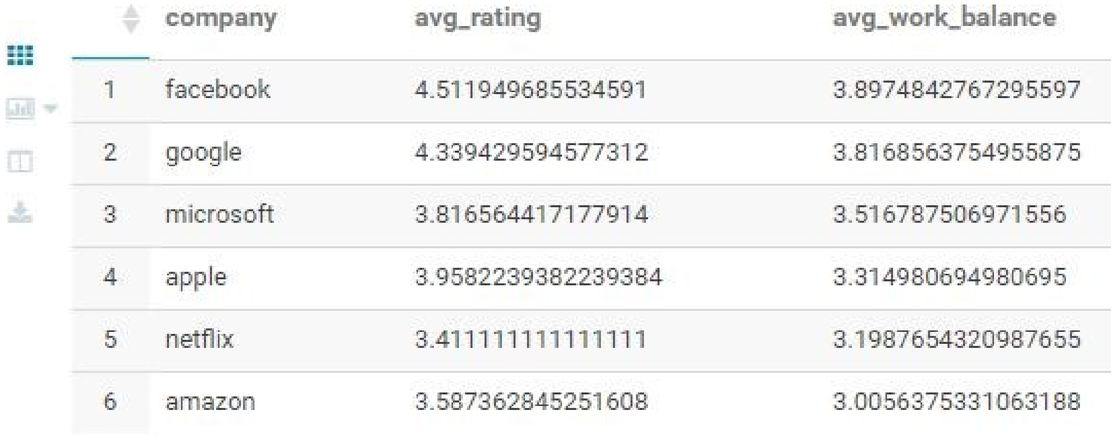
> SELECT company, avg(rating) as Avg_Rating, avg(cv) as Avg_Culture_Value FROM employeereview GROUP BY company ORDER BY Avg_Culture_Value DESC



	company	avg_rating	avg_culture_value
1	facebook	4.511949685534591	4.467295597484276
	google	4.339429594577312	4.258728737690242
3	apple	3.9582239382239384	4.077065637065637
4	microsoft	3.816564417177914	3.7402677077523703
5	netflix	3.411111111111111	3.7
6	amazon	3.587362845251608	3.6027241770715097

Overall vs Work balance

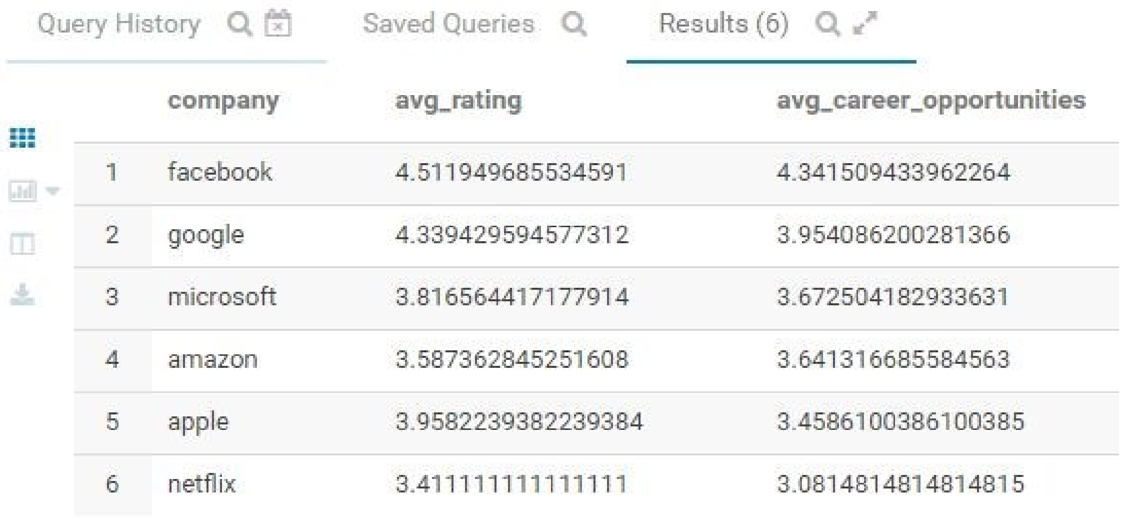
> SELECT company, avg(rating) as Avg_Rating, avg(wb) as Avg_Work_Balance FROM employeereview GROUP BY company ORDER BY Avg_Work_Balance DESC



	company	avg_rating	avg_work_balance
1	facebook	4.511949685534591	3.8974842767295597
2	google	4.339429594577312	3.8168563754955875
3	microsoft	3.816564417177914	3.516787506971556
4	apple	3.9582239382239384	3.314980694980695
5	netflix	3.411111111111111	3.1987654320987655
6	amazon	3.587362845251608	3.0056375331063188

Overall vs Career Opportunities

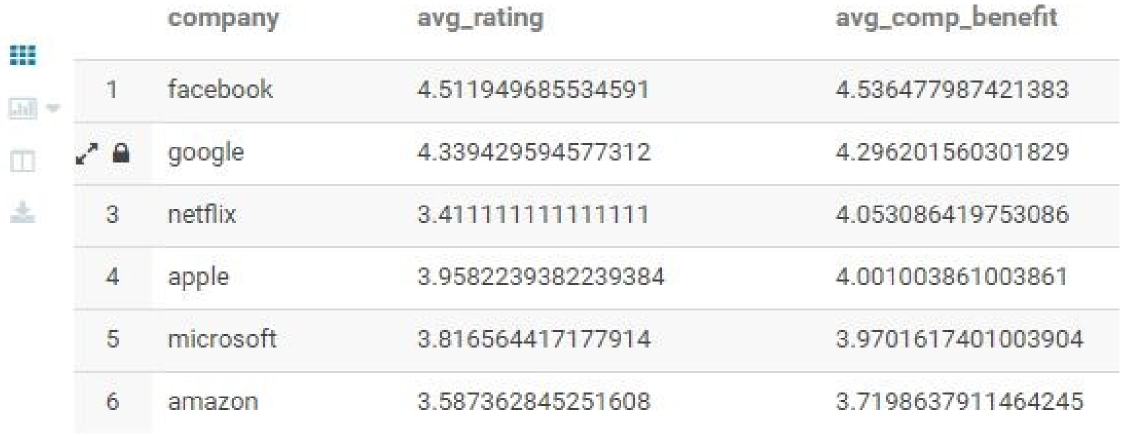
> SELECT company, avg(rating) as Avg_Rating, avg(co) as Avg_Career_Opportunities FROM employeereview GROUP BY company ORDER BY Avg_Career_Opportunities DESC



	company	avg_rating	avg_career_opportunities
1	facebook	4.511949685534591	4.341509433962264
2	google	4.339429594577312	3.954086200281366
3	microsoft	3.816564417177914	3.672504182933631
4	amazon	3.587362845251608	3.641316685584563
5	apple	3.9582239382239384	3.4586100386100385
6	netflix	3.411111111111111	3.0814814814814815

Overall vs Compensation benefit

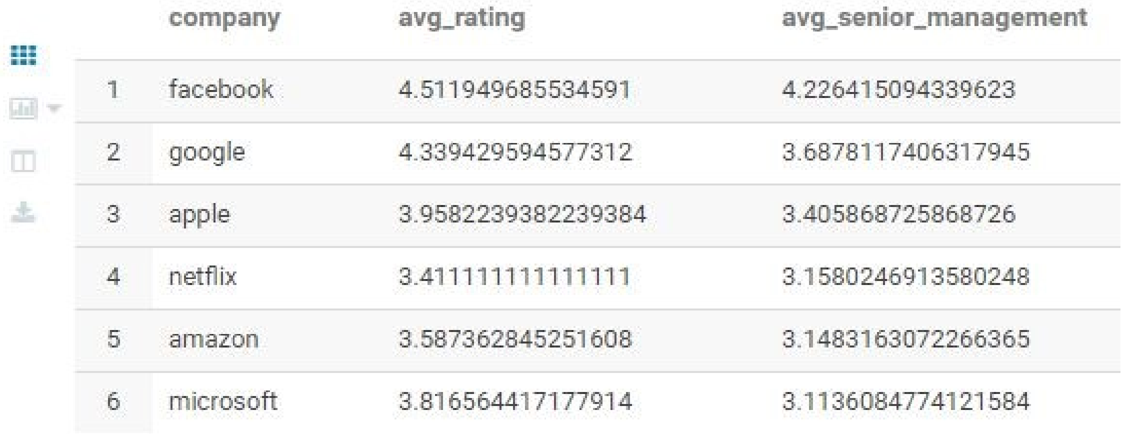
```
> SELECT company, avg(rating) as Avg_Rating, avg(cb) as Avg_Comp_Benefit FROM employeereview GROUP BY company ORDER BY Avg_Comp_Benefit  DESC
```



	company	avg_rating	avg_comp_benefit
1	facebook	4.511949685534591	4.536477987421383
	google	4.339429594577312	4.296201560301829
3	netflix	3.411111111111111	4.053086419753086
4	apple	3.9582239382239384	4.001003861003861
5	microsoft	3.816564417177914	3.9701617401003904
6	amazon	3.587362845251608	3.7198637911464245

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
```



	company	avg_rating	avg_senior_management
1	facebook	4.511949685534591	4.226415094339623
2	google	4.339429594577312	3.6878117406317945
3	apple	3.9582239382239384	3.405868725868726
4	netflix	3.411111111111111	3.1580246913580248
5	amazon	3.587362845251608	3.1483163072266365
6	microsoft	3.816564417177914	3.1136084774121584

Document your findings for the following:

a) Which corporation is worth working for

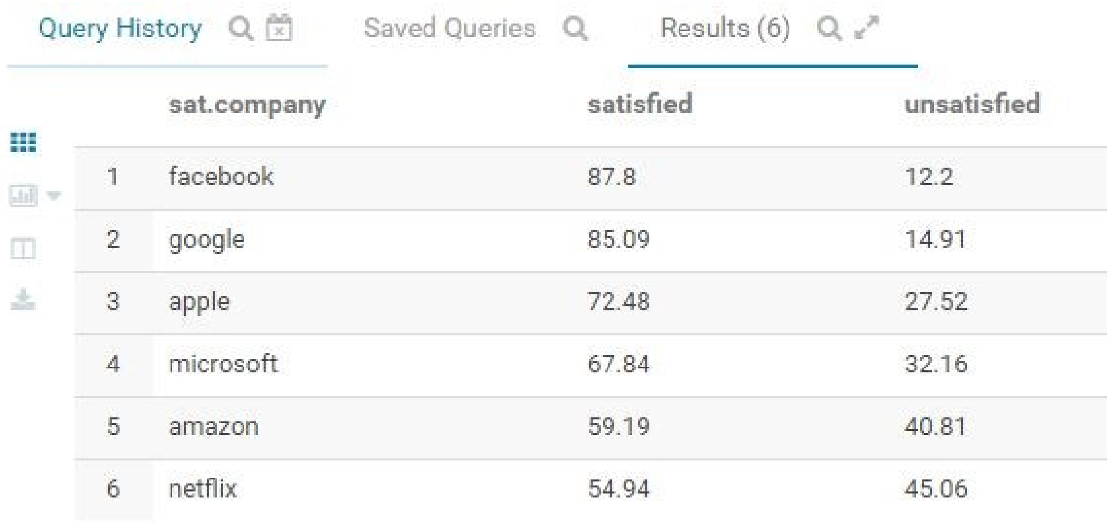
From the above queries we can confirm that facebook has best ratings overall and is the corporation worth working for.

b) 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;
```



	sat.company	satisfied	unsatisfied
1	facebook	87.8	12.2
2	google	85.09	14.91
3	apple	72.48	27.52
4	microsoft	67.84	32.16
5	amazon	59.19	40.81
6	netflix	54.94	45.06

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