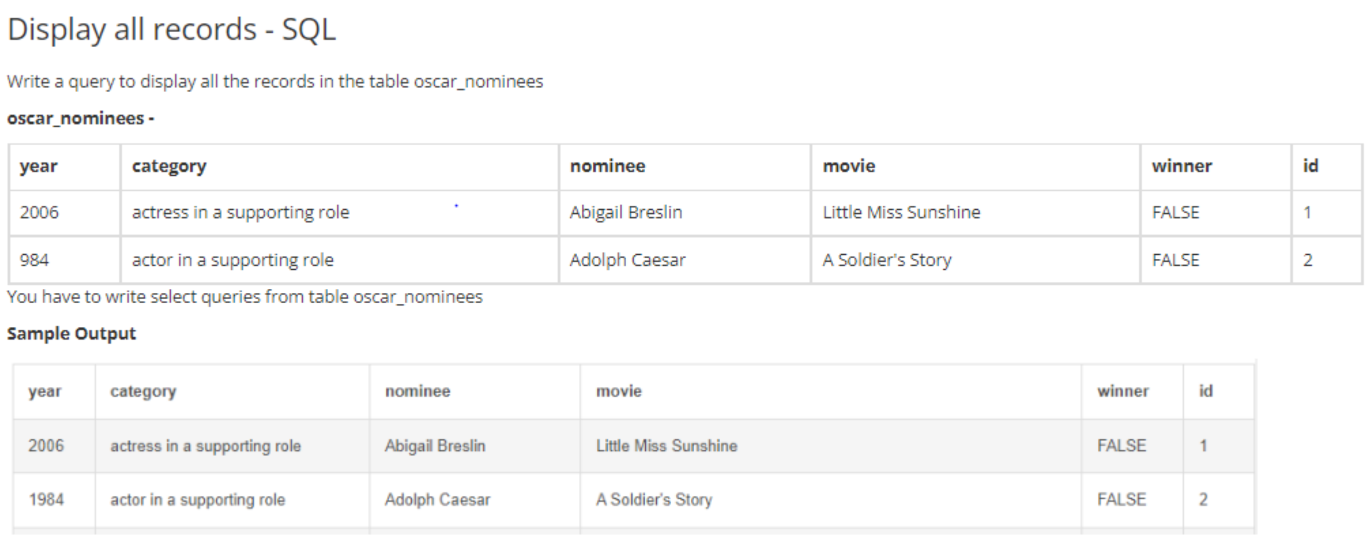
# SQL Logic Building - 1

## 1. Display all records - SQL

Write a query to display all the records in the table oscar_nominees



Display all records - SQL

Write a query to display all the records in the table oscar_nominees

**oscar_nominees -**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 984 | actor in a supporting role | Adolph Caesar | A Soldier's Story | FALSE | 2 |

You have to write select queries from table oscar_nominees

**Sample Output**

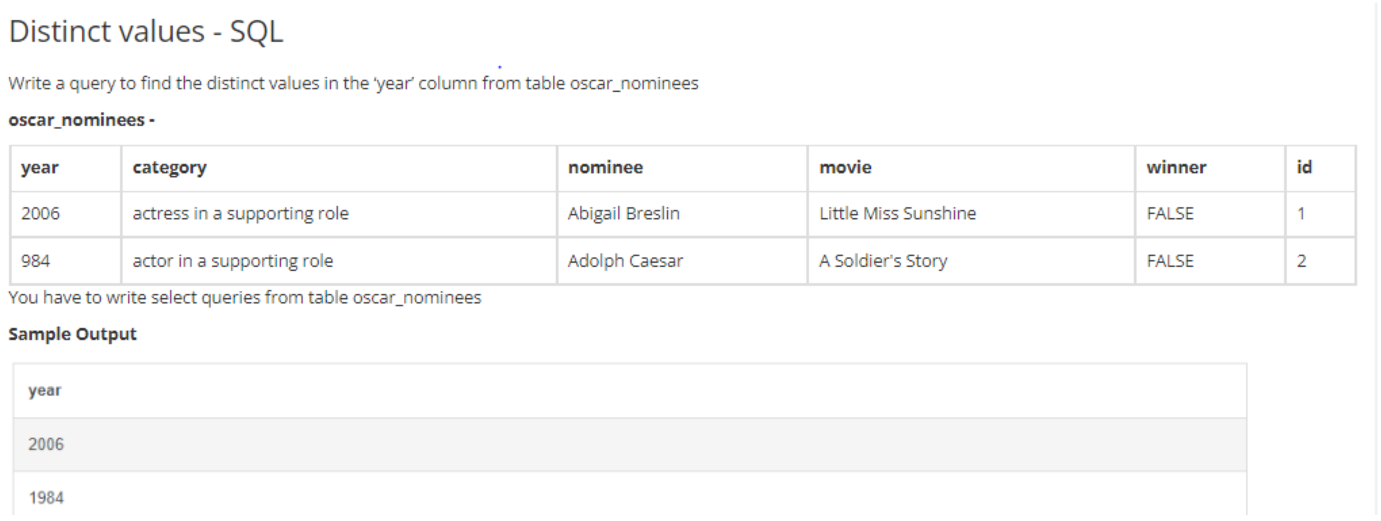
| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 1984 | actor in a supporting role | Adolph Caesar | A Soldier's Story | FALSE | 2 |

Solution:

select * from oscar_nominees;

## 2.Distinct values - SQL

Write a query to find the distinct values in the 'year' column from table oscar_nominees



Distinct values - SQL

Write a query to find the distinct values in the 'year' column from table oscar_nominees

**oscar_nominees -**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 984 | actor in a supporting role | Adolph Caesar | A Soldier's Story | FALSE | 2 |

You have to write select queries from table oscar_nominees

**Sample Output**

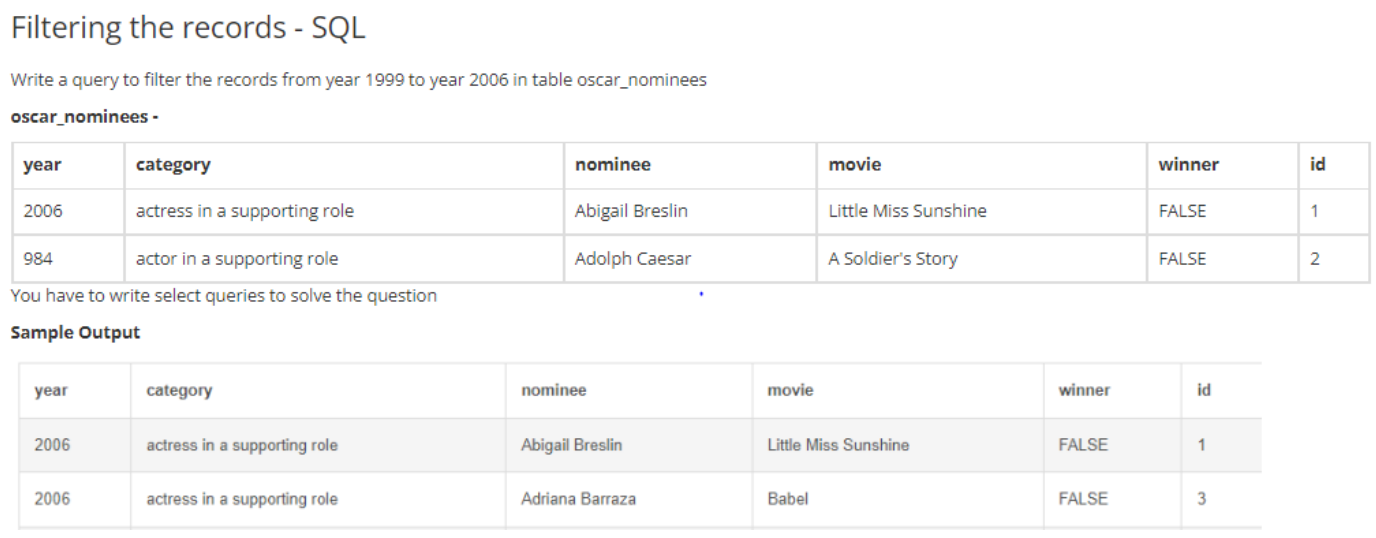
| year |
|---|
| 2006 |
| 1984 |

Solution:

select distinct year from oscar_nominees;

## 3.Filtering the records - SQL

Write a query to filter the records from year 1999 to year 2006 in table oscar_nominees



### Filtering the records - SQL

Write a query to filter the records from year 1999 to year 2006 in table oscar_nominees

**oscar_nominees -**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 984 | actor in a supporting role | Adolph Caesar | A Soldier's Story | FALSE | 2 |

You have to write select queries to solve the question

**Sample Output**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 2006 | actress in a supporting role | Adriana Barraza | Babel | FALSE | 3 |

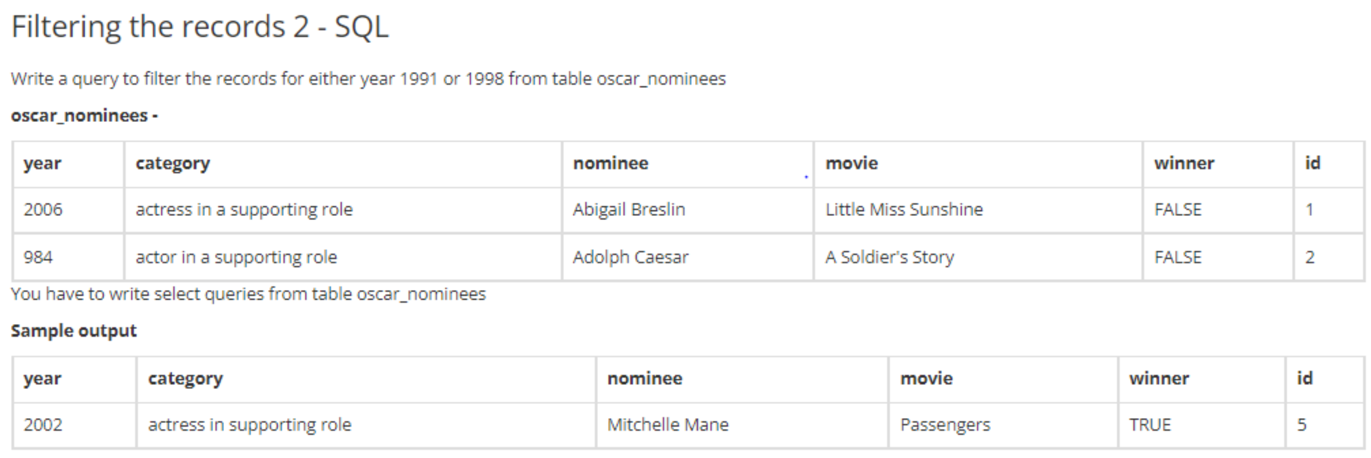
Solution:

select * from oscar_nominees

where year between 1999 and 2006;

## 4. Filtering the records 2 - SQL

Write a query to filter the records for either year 1991 or 1998 from table oscar_nominees



### Filtering the records 2 - SQL

Write a query to filter the records for either year 1991 or 1998 from table oscar_nominees

**oscar_nominees -**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 984 | actor in a supporting role | Adolph Caesar | A Soldier's Story | FALSE | 2 |

You have to write select queries from table oscar_nominees

**Sample output**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2002 | actress in supporting role | Mitchelle Mane | Passengers | TRUE | 5 |

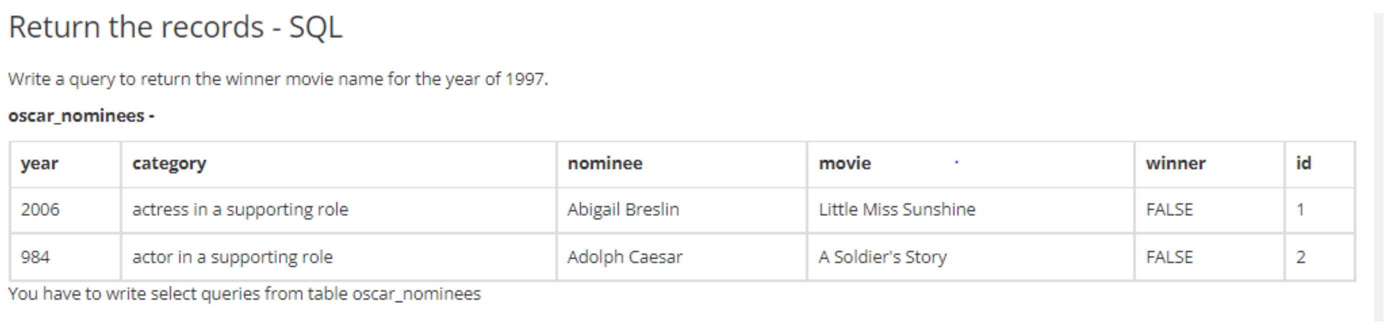
Solution:

select * from oscar_nominees

where year = 1991 or year = 1998;

## 5. Return the records - SQL

Write a query to return the winner movie name for the year of 1997.



Return the records - SQL

Write a query to return the winner movie name for the year of 1997.

**oscar_nominees -**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 984 | actor in a supporting role | Adolph Caesar | A Soldier's Story | FALSE | 2 |

You have to write select queries from table oscar_nominees

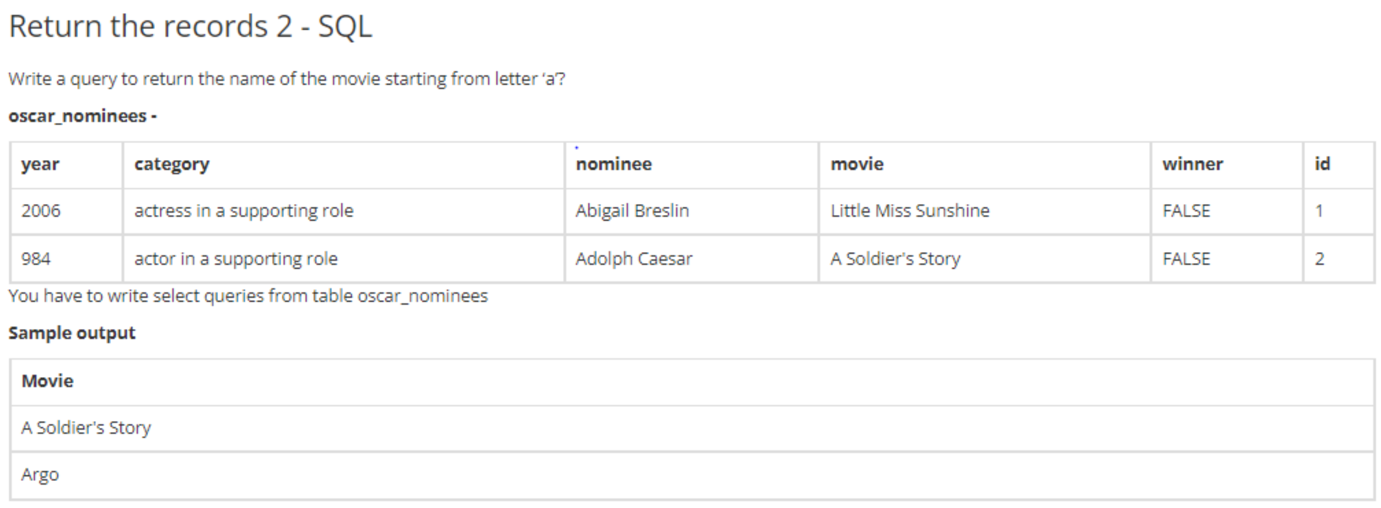
Solution:

select movie from oscar_nominees

where year = 1997 and winner = True;

## 6. Return the records 2 - SQL

Write a query to return the name of the movie starting from letter ‘a’?



Return the records 2 - SQL

Write a query to return the name of the movie starting from letter ‘a’?

**oscar_nominees -**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 984 | actor in a supporting role | Adolph Caesar | A Soldier's Story | FALSE | 2 |

You have to write select queries from table oscar_nominees

**Sample output**

| Movie |
|---|
| A Soldier's Story |
| Argo |

Solution:

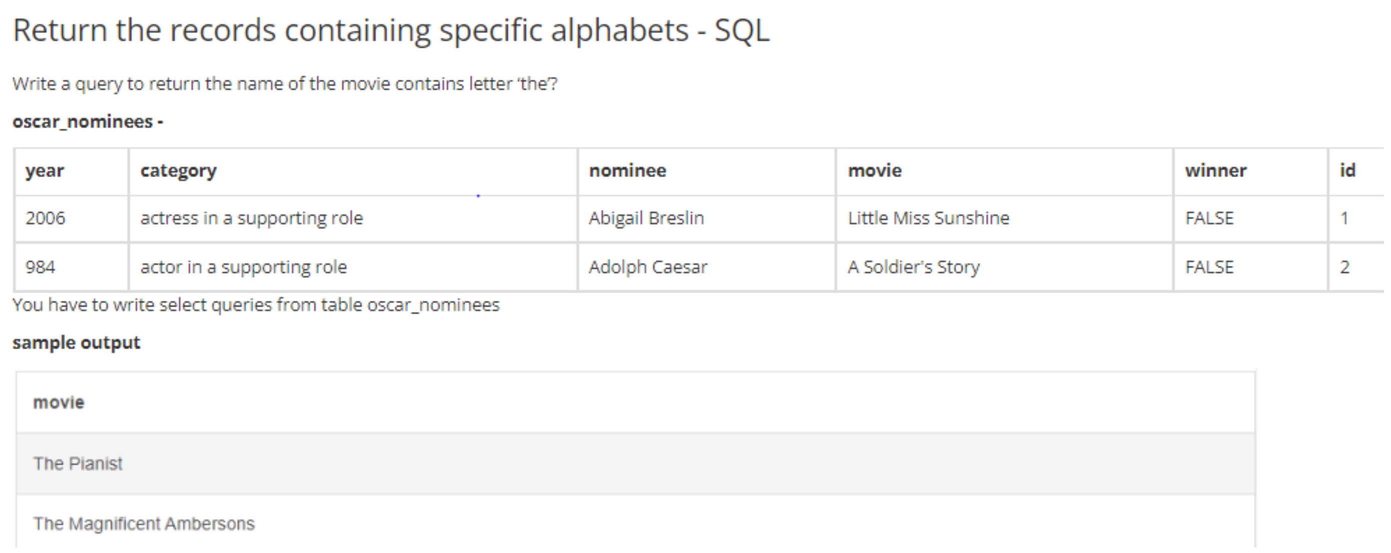
select movie

from oscar_nominees

where movie like "a%";

## 7. Return the records containing specific alphabets - SQL

Write a query to return the name of the movie contains letter ‘the’?


Return the records containing specific alphabets - SQL

Write a query to return the name of the movie contains letter 'the'?

**oscar_nominees -**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 2006 | actress in a supporting role | Abigail Breslin | Little Miss Sunshine | FALSE | 1 |
| 984 | actor in a supporting role | Adolph Caesar | A Soldier's Story | FALSE | 2 |

You have to write select queries from table oscar_nominees

**sample output**

| movie |
|---|
| The Pianist |
| The Magnificent Ambersons |

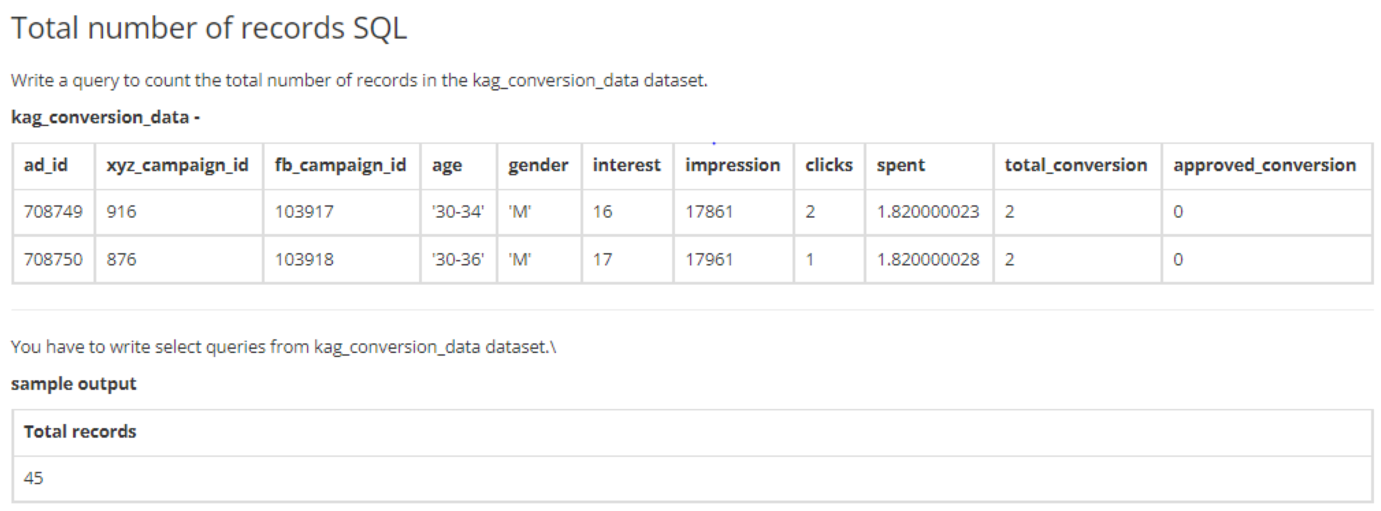
Solution:

select movie from oscar_nominees

where movie like "%the%"

## 8. Total number of records SQL

Write a query to count the total number of records in the kag_conversion_data dataset.


Total number of records SQL

Write a query to count the total number of records in the kag_conversion_data dataset.

**kag_conversion_data -**

| ad_id | xyz_campaign_id | fb_campaign_id | age | gender | interest | impression | clicks | spent | total_conversion | approved_conversion |
|---|---|---|---|---|---|---|---|---|---|---|
| 708749 | 916 | 103917 | '30-34' | 'M' | 16 | 17861 | 2 | 1.820000023 | 2 | 0 |
| 708750 | 876 | 103918 | '30-36' | 'M' | 17 | 17961 | 1 | 1.820000028 | 2 | 0 |

You have to write select queries from kag_conversion_data dataset.\

**sample output**

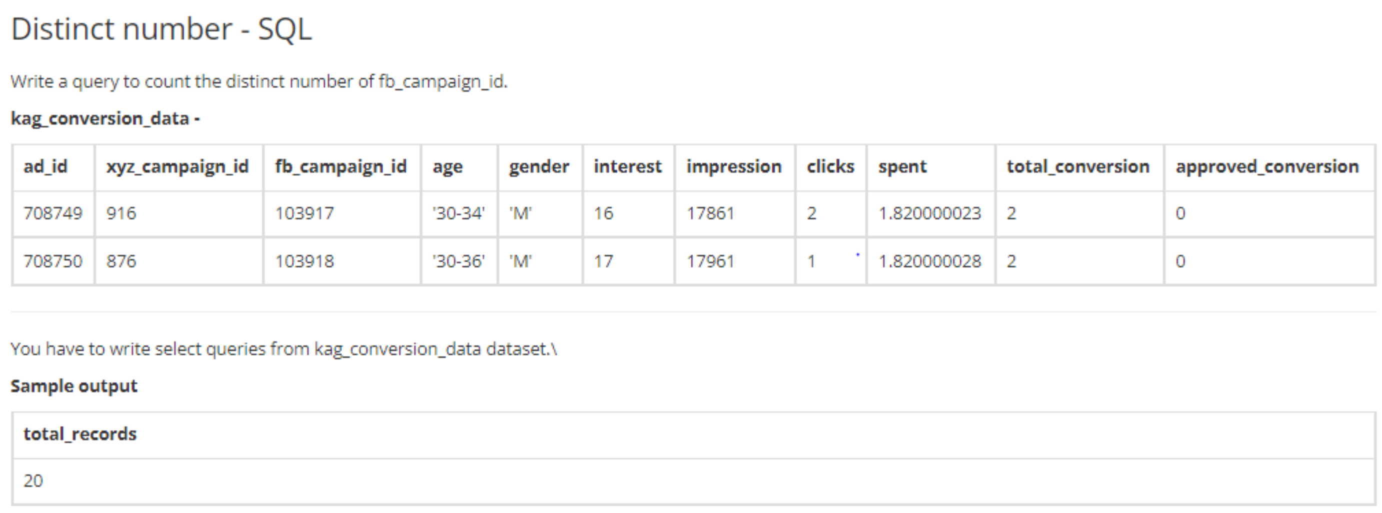
| Total records |
|---|
| 45 |

Solution:

select count(*) from kag_conversion_data

## 9. Distinct number - SQL

Write a query to count the distinct number of fb_campaign_id.



Distinct number - SQL

Write a query to count the distinct number of fb_campaign_id.

**kag_conversion_data -**

| ad_id | xyz_campaign_id | fb_campaign_id | age | gender | interest | impression | clicks | spent | total_conversion | approved_conversion |
|---|---|---|---|---|---|---|---|---|---|---|
| 708749 | 916 | 103917 | '30-34' | 'M' | 16 | 17861 | 2 | 1.820000023 | 2 | 0 |
| 708750 | 876 | 103918 | '30-36' | 'M' | 17 | 17961 | 1 | 1.820000028 | 2 | 0 |

You have to write select queries from kag_conversion_data dataset.\

**Sample output**

| total_records |
|---|
| 20 |

Solution:

select count(distinct fb_campaign_id)

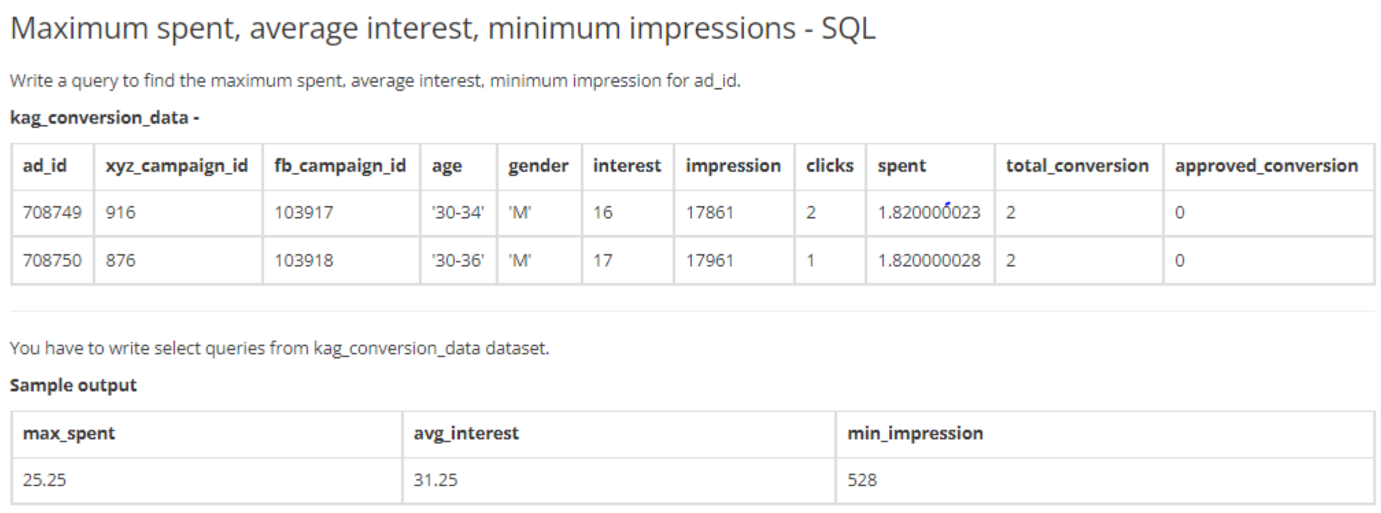
from kag_conversion_data

```
In [ ]:
```

# SQL Logic Building - 2

### 1. Maximum spent, average interest, minimum impressions - SQL

Write a query to find the maximum spent, average interest, minimum impression for ad_id.



Maximum spent, average interest, minimum impressions - SQL

Write a query to find the maximum spent, average interest, minimum impression for ad_id.

**kag_conversion_data -**

| ad_id | xyz_campaign_id | fb_campaign_id | age | gender | interest | impression | clicks | spent | total_conversion | approved_conversion |
|---|---|---|---|---|---|---|---|---|---|---|
| 708749 | 916 | 103917 | '30-34' | 'M' | 16 | 17861 | 2 | 1.820000023 | 2 | 0 |
| 708750 | 876 | 103918 | '30-36' | 'M' | 17 | 17961 | 1 | 1.820000028 | 2 | 0 |

You have to write select queries from kag_conversion_data dataset.

**Sample output**

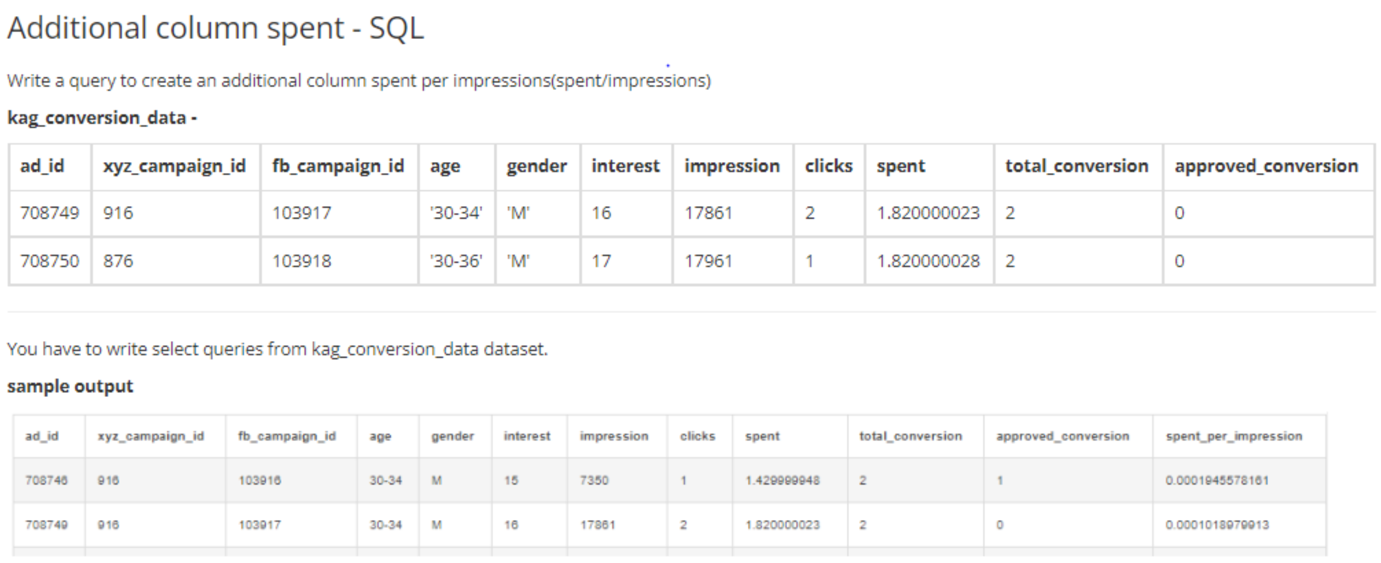
| max_spent | avg_interest | min_impression |
|---|---|---|
| 25.25 | 31.25 | 528 |

Solution:

select max(spent) as max_spent,avg(interest) as avg_interest,min(impression) as min_impression from kag_conversion_data

### 2. Additional column spent - SQL

Write a query to create an additional column spent per impressions(spent/impressions)



Additional column spent - SQL

Write a query to create an additional column spent per impressions(spent/impressions)

**kag_conversion_data -**

| ad_id | xyz_campaign_id | fb_campaign_id | age | gender | interest | impression | clicks | spent | total_conversion | approved_conversion |
|---|---|---|---|---|---|---|---|---|---|---|
| 708749 | 916 | 103917 | '30-34' | 'M' | 16 | 17861 | 2 | 1.820000023 | 2 | 0 |
| 708750 | 876 | 103918 | '30-36' | 'M' | 17 | 17961 | 1 | 1.820000028 | 2 | 0 |

You have to write select queries from kag_conversion_data dataset.

**sample output**

| ad_id | xyz_campaign_id | fb_campaign_id | age | gender | interest | impression | clicks | spent | total_conversion | approved_conversion | spent_per_impression |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 708746 | 916 | 103916 | 30-34 | M | 15 | 7350 | 1 | 1.429999948 | 2 | 1 | 0.0001945578161 |
| 708749 | 916 | 103917 | 30-34 | M | 16 | 17861 | 2 | 1.820000023 | 2 | 0 | 0.0001018979913 |

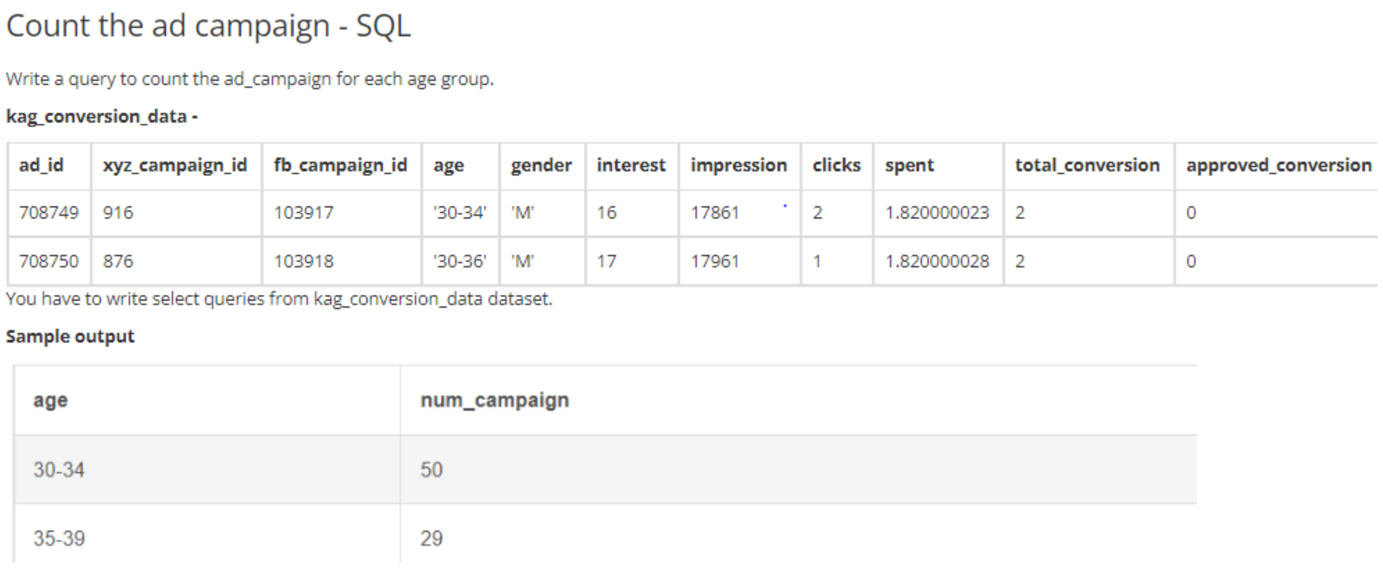
Solution:

SELECT*, spent/impression AS spent_per_impression

FROM kag_conversion_data

### 3. Count the ad campaign - SQL

Write a query to count the ad_campaign for each age group. ##...........(GROUP BY).....



Count the ad campaign - SQL

Write a query to count the ad_campaign for each age group.

**kag_conversion_data -**

| ad_id | xyz_campaign_id | fb_campaign_id | age | gender | interest | impression | clicks | spent | total_conversion | approved_conversion |
|---|---|---|---|---|---|---|---|---|---|---|
| 708749 | 916 | 103917 | '30-34' | 'M' | 16 | 17861 | 2 | 1.820000023 | 2 | 0 |
| 708750 | 876 | 103918 | '30-36' | 'M' | 17 | 17961 | 1 | 1.820000028 | 2 | 0 |

You have to write select queries from kag_conversion_data dataset.

**Sample output**

| age | num_campaign |
|---|---|
| 30-34 | 50 |
| 35-39 | 29 |

Solution:

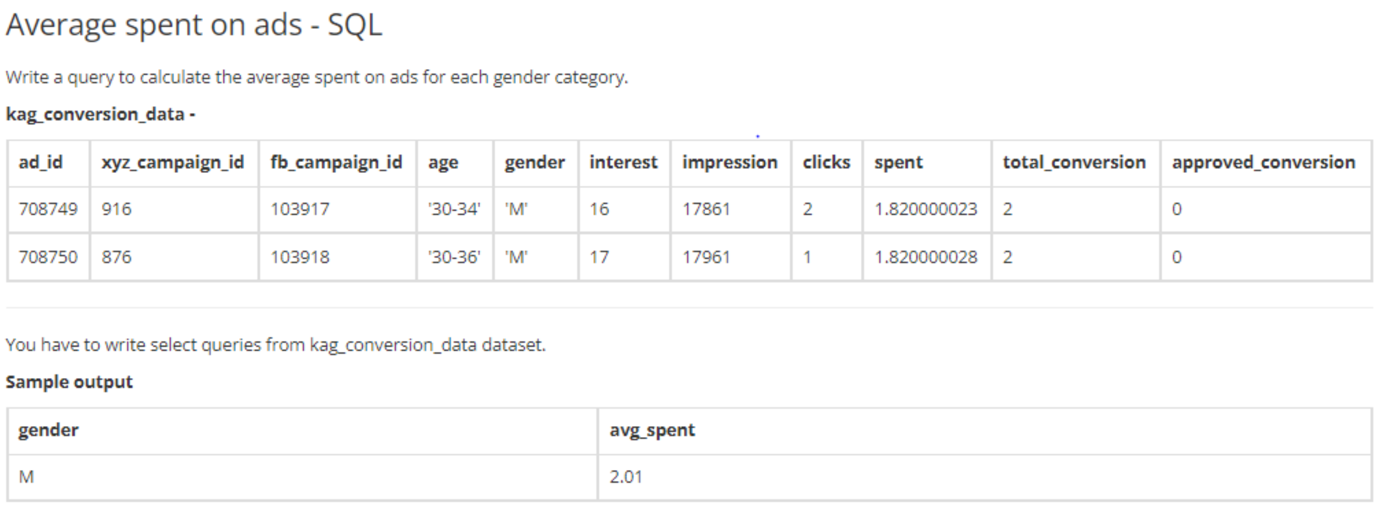
select age,count(ad_id) as num_campaign

from kag_conversion_data

group by age

## 4. Average spent on ads - SQL

Write a query to calculate the average spent on ads for each gender category. ##..........(GROUP BY)..........



Average spent on ads - SQL

Write a query to calculate the average spent on ads for each gender category.

**kag_conversion_data -**

| ad_id | xyz_campaign_id | fb_campaign_id | age | gender | interest | impression | clicks | spent | total_conversion | approved_conversion |
|---|---|---|---|---|---|---|---|---|---|---|
| 708749 | 916 | 103917 | '30-34' | 'M' | 16 | 17861 | 2 | 1.820000023 | 2 | 0 |
| 708750 | 876 | 103918 | '30-36' | 'M' | 17 | 17961 | 1 | 1.820000028 | 2 | 0 |

You have to write select queries from kag_conversion_data dataset.

**Sample output**

| gender | avg_spent |
|---|---|
| M | 2.01 |

Solution:

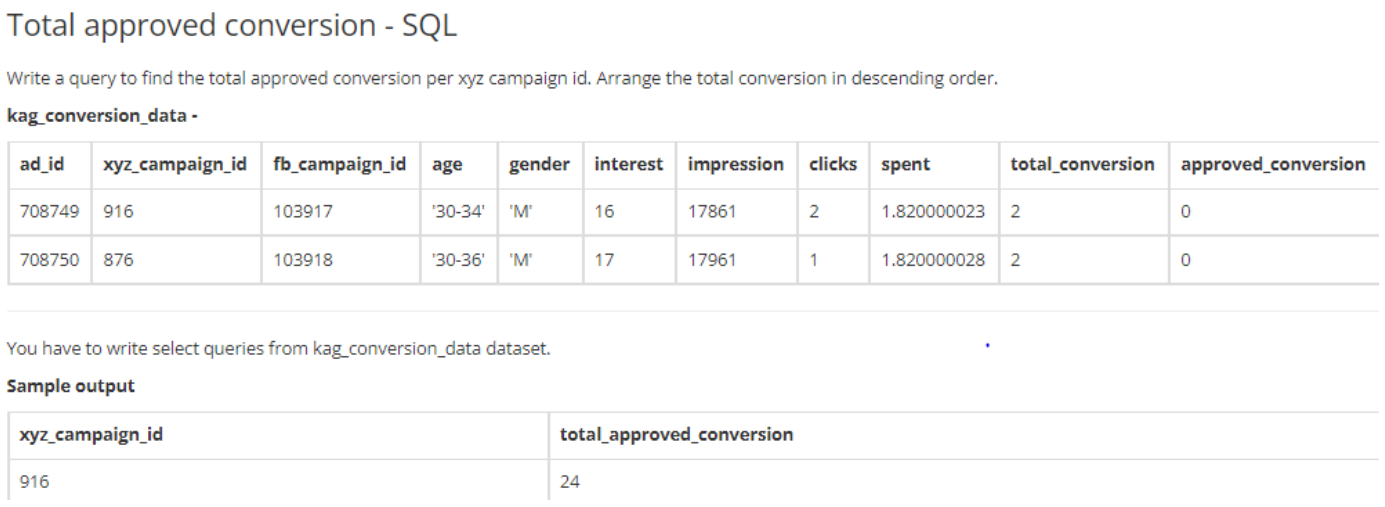
SELECT gender,AVG(spent) AS avg_spent

FROM kag_conversion_data

GROUP BY Gender

## 5. Total approved conversion - SQL

Write a query to find the total approved conversion per xyz campaign id. Arrange the total conversion in descending order.



Total approved conversion - SQL

Write a query to find the total approved conversion per xyz campaign id. Arrange the total conversion in descending order.

**kag_conversion_data -**

| ad_id | xyz_campaign_id | fb_campaign_id | age | gender | interest | impression | clicks | spent | total_conversion | approved_conversion |
|---|---|---|---|---|---|---|---|---|---|---|
| 708749 | 916 | 103917 | '30-34' | 'M' | 16 | 17861 | 2 | 1.820000023 | 2 | 0 |
| 708750 | 876 | 103918 | '30-36' | 'M' | 17 | 17961 | 1 | 1.820000028 | 2 | 0 |

You have to write select queries from kag_conversion_data dataset.

**Sample output**

| xyz_campaign_id | total_approved_conversion |
|---|---|
| 916 | 24 |

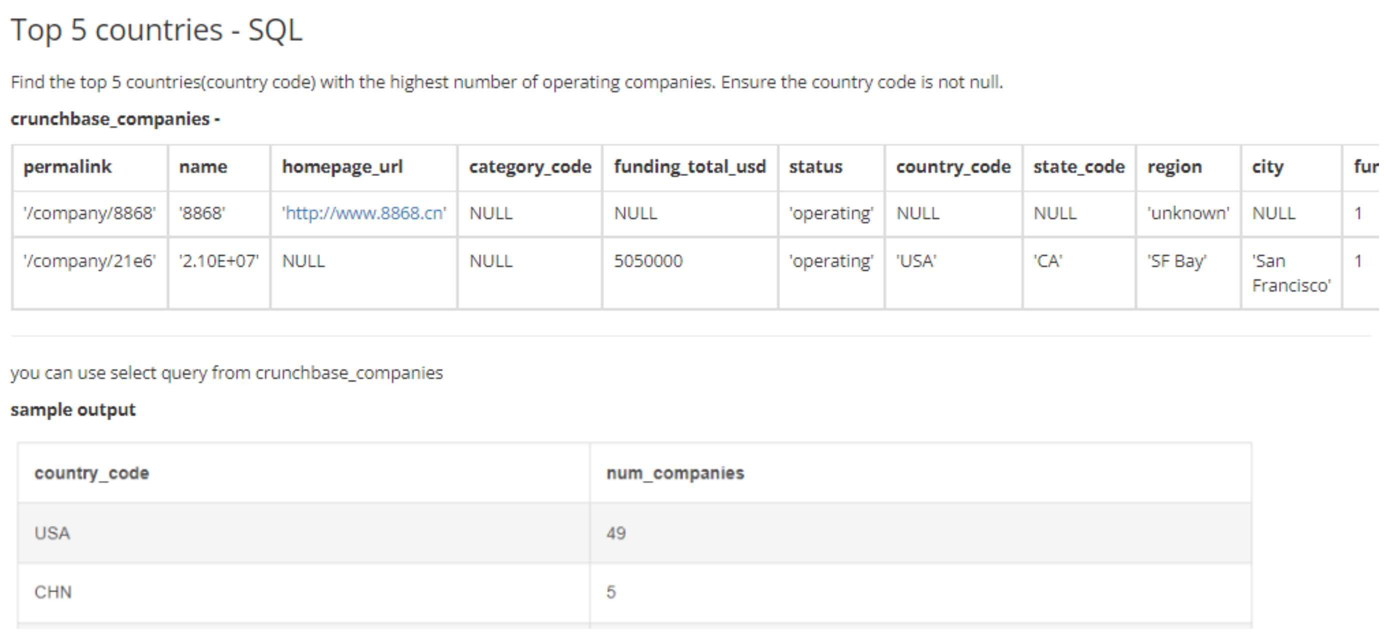
Solution:

```
SELECT xyz_campaign_id, SUM(approved_conversion) AS total_approved_conversion

FROM kag_conversion_data

GROUP BY xyz_campaign_id

ORDER BY total_approved_conversion DESC
```

**6. Top 5 countries - SQL**

Find the top 5 countries(country code) with the highest number of operating companies. Ensure the country code is not null.

crunchbase_companies -



Top 5 countries - SQL

Find the top 5 countries(country code) with the highest number of operating companies. Ensure the country code is not null.

**crunchbase_companies -**

| permalink | name | homepage_url | category_code | funding_total_usd | status | country_code | state_code | region | city | fur |
|---|---|---|---|---|---|---|---|---|---|---|
| '/company/8868' | '8868' | 'http://www.8868.cn' | NULL | NULL | 'operating' | NULL | NULL | 'unknown' | NULL | 1 |
| '/company/21e6' | '2.10E+07' | NULL | NULL | 5050000 | 'operating' | 'USA' | 'CA' | 'SF Bay' | 'San Francisco' | 1 |

you can use select query from crunchbase_companies

**sample output**

| country_code | num_companies |
|---|---|
| USA | 49 |
| CHN | 5 |

Solution:

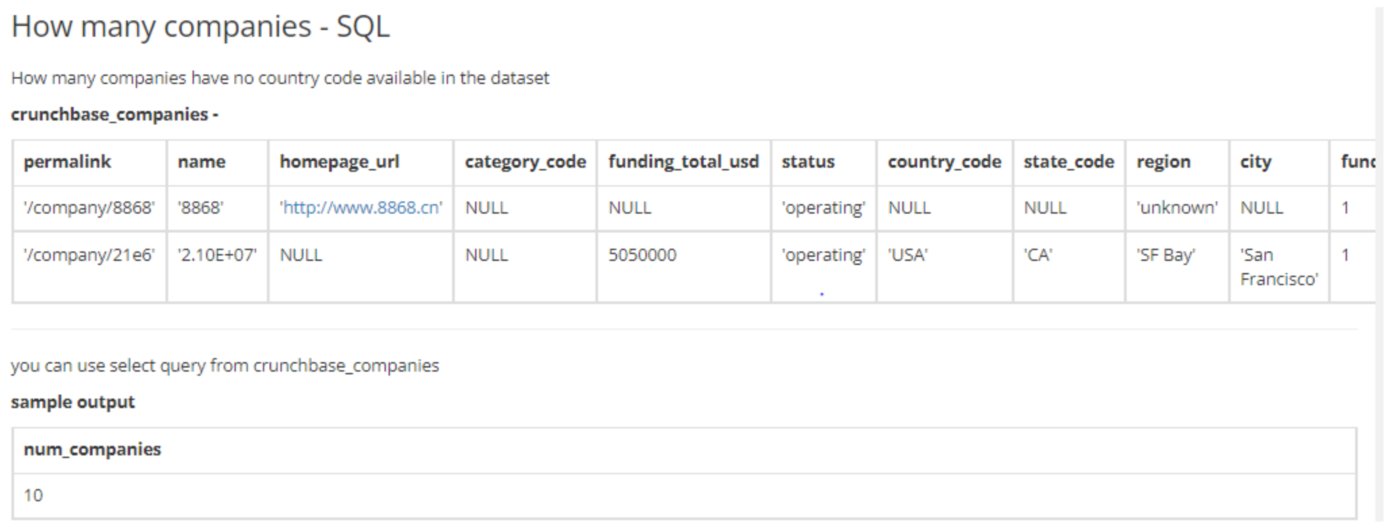
```
SELECT country_code, COUNT(name) as num_companies

FROM crunchbase_companies

WHERE country_code IS NOT NULL

GROUP BY country_code

ORDER BY num_companies DESC

LIMIT 5
```

**7. How many companies - SQL**

## 7. How many companies - SQL

How many companies have no country code available in the dataset

crunchbase_companies -



How many companies - SQL

How many companies have no country code available in the dataset

**crunchbase_companies -**

| permalink | name | homepage_url | category_code | funding_total_usd | status | country_code | state_code | region | city | func |
|---|---|---|---|---|---|---|---|---|---|---|
| '/company/8868' | '8868' | 'http://www.8868.cn' | NULL | NULL | 'operating' | NULL | NULL | 'unknown' | NULL | 1 |
| '/company/21e6' | '2.10E+07' | NULL | NULL | 5050000 | 'operating' | 'USA' | 'CA' | 'SF Bay' | 'San Francisco' | 1 |

you can use select query from crunchbase_companies

**sample output**

| num_companies |
|---|
| 10 |

Solution:

SELECT COUNT(name) as num_companies

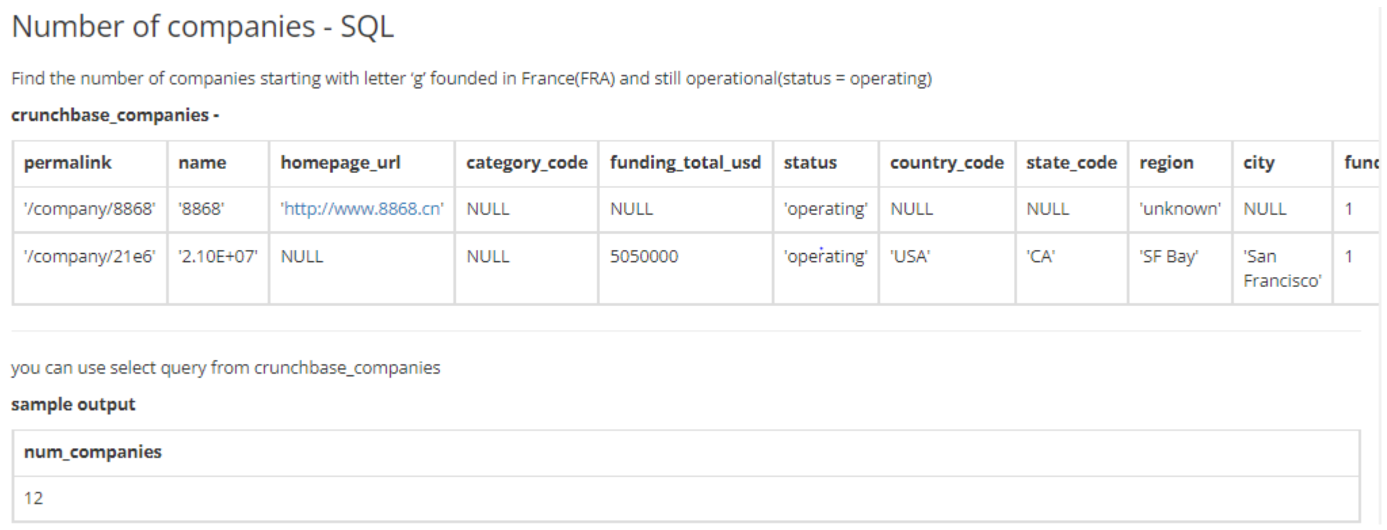
FROM crunchbase_companies

WHERE country_code IS NULL

## 8. Number of companies - SQL

Find the number of companies starting with letter ‘g’ founded in France(FRA) and still operational(status = operating)

crunchbase_companies -



Number of companies - SQL

Find the number of companies starting with letter 'g' founded in France(FRA) and still operational(status = operating)

**crunchbase_companies -**

| permalink | name | homepage_url | category_code | funding_total_usd | status | country_code | state_code | region | city | func |
|---|---|---|---|---|---|---|---|---|---|---|
| '/company/8868' | '8868' | 'http://www.8868.cn' | NULL | NULL | 'operating' | NULL | NULL | 'unknown' | NULL | 1 |
| '/company/21e6' | '2.10E+07' | NULL | NULL | 5050000 | 'operating' | 'USA' | 'CA' | 'SF Bay' | 'San Francisco' | 1 |

you can use select query from crunchbase_companies

**sample output**

| num_companies |
|---|
| 12 |

Solution:

SELECT COUNT(name) AS num_companies

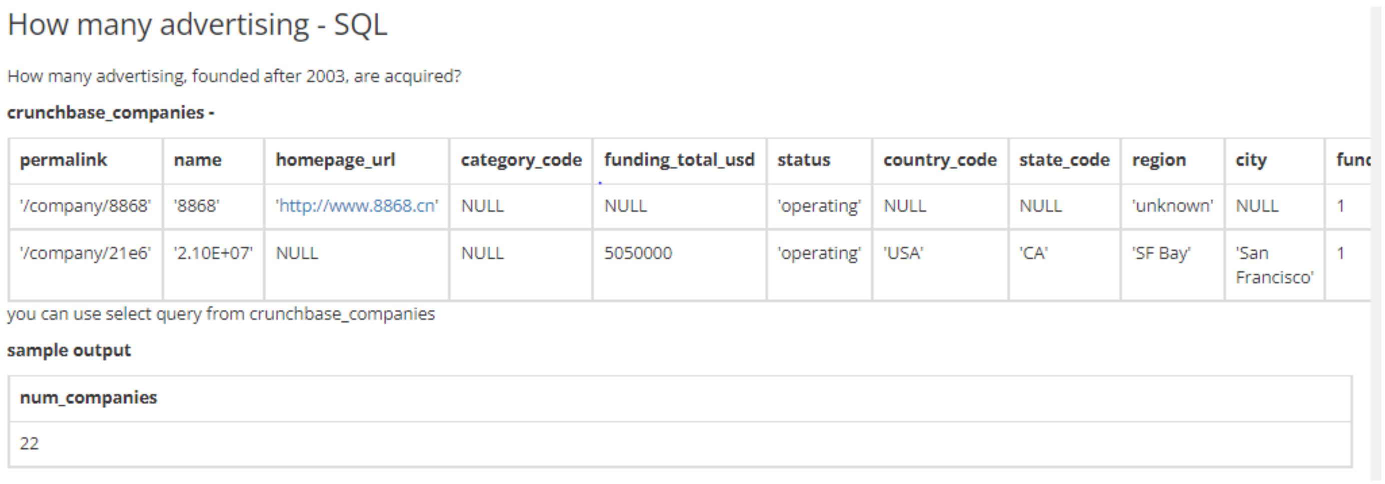
FROM crunchbase_companies

WHERE name LIKE 'g%' AND country_code = 'FRA' AND status ='operating';

### 9. How many advertising - SQL

How many advertising, founded after 2003, are acquired?

crunchbase_companies -



How many advertising - SQL

How many advertising, founded after 2003, are acquired?

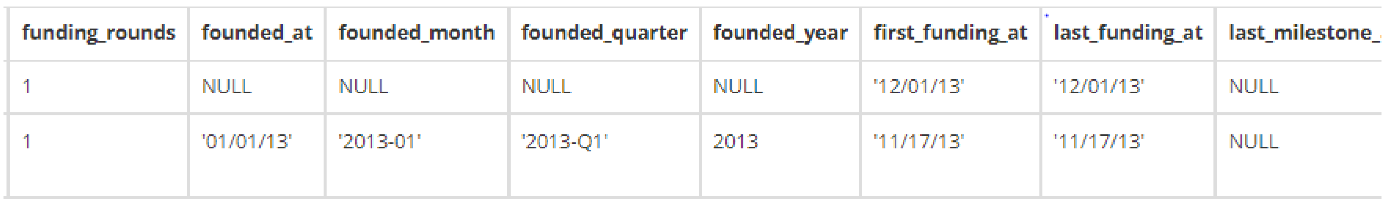
**crunchbase_companies -**

| permalink | name | homepage_url | category_code | funding_total_usd | status | country_code | state_code | region | city | func |
|---|---|---|---|---|---|---|---|---|---|---|
| '/company/8868' | '8868' | 'http://www.8868.cn' | NULL | NULL | 'operating' | NULL | NULL | 'unknown' | NULL | 1 |
| '/company/21e6' | '2.10E+07' | NULL | NULL | 5050000 | 'operating' | 'USA' | 'CA' | 'SF Bay' | 'San Francisco' | 1 |

you can use select query from crunchbase_companies

**sample output**

| num_companies |
|---|
| 22 |



| funding_rounds | founded_at | founded_month | founded_quarter | founded_year | first_funding_at | last_funding_at | last_milestone_ |
|---|---|---|---|---|---|---|---|
| 1 | NULL | NULL | NULL | NULL | '12/01/13' | '12/01/13' | NULL |
| 1 | '01/01/13' | '2013-01' | '2013-Q1' | 2013 | '11/17/13' | '11/17/13' | NULL |

solution:

SELECT COUNT(name) AS num_companies

FROM crunchbase_companies

WHERE founded_year > 2003 AND status = 'acquired' AND category_code = 'advertising';

```
In [ ]:
```

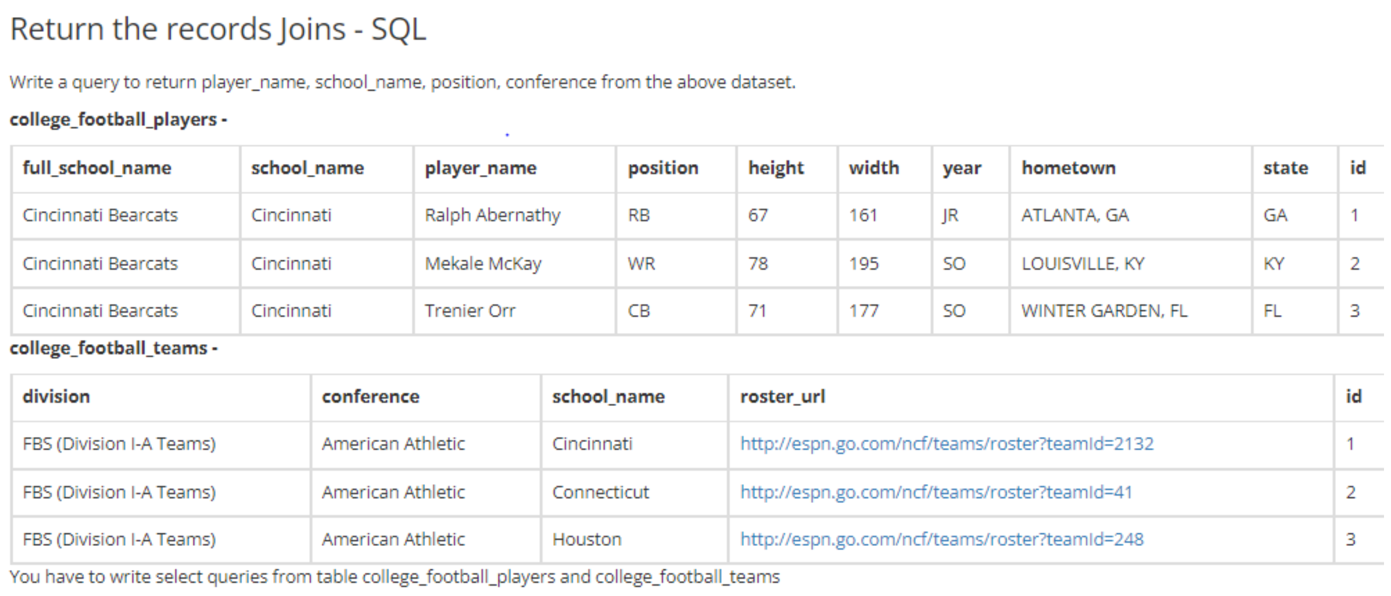
# SQL Logic Building - 3

### 1. Return the records Joins - SQL

Write a query to return player_name, school_name, position, conference from the above dataset.

college_football_players -

college_football_teams -



Return the records Joins - SQL

Write a query to return player_name, school_name, position, conference from the above dataset.

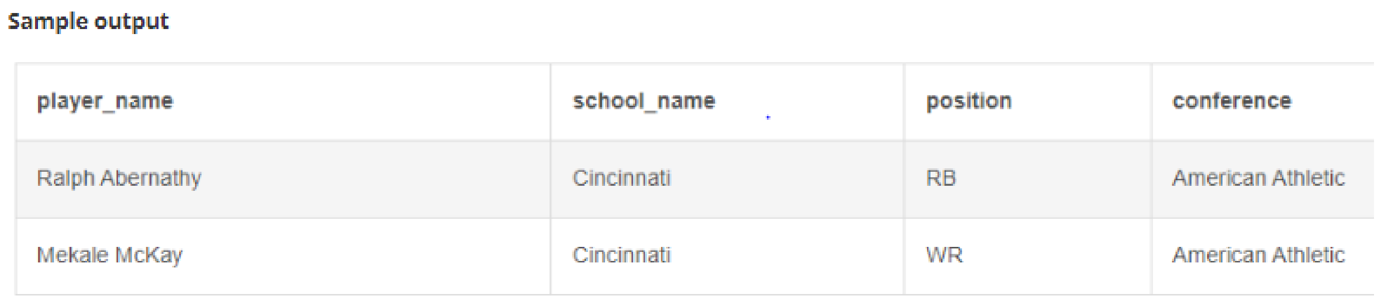
**college_football_players -**

| full_school_name | school_name | player_name | position | height | width | year | hometown | state | id |
|---|---|---|---|---|---|---|---|---|---|
| Cincinnati Bearcats | Cincinnati | Ralph Abernathy | RB | 67 | 161 | JR | ATLANTA, GA | GA | 1 |
| Cincinnati Bearcats | Cincinnati | Mekale McKay | WR | 78 | 195 | SO | LOUISVILLE, KY | KY | 2 |
| Cincinnati Bearcats | Cincinnati | Trenier Orr | CB | 71 | 177 | SO | WINTER GARDEN, FL | FL | 3 |

**college_football_teams -**

| division | conference | school_name | roster_url | id |
|---|---|---|---|---|
| FBS (Division I-A Teams) | American Athletic | Cincinnati | http://espn.go.com/ncf/teams/roster?teamId=2132 | 1 |
| FBS (Division I-A Teams) | American Athletic | Connecticut | http://espn.go.com/ncf/teams/roster?teamId=41 | 2 |
| FBS (Division I-A Teams) | American Athletic | Houston | http://espn.go.com/ncf/teams/roster?teamId=248 | 3 |

You have to write select queries from table college_football_players and college_football_teams



**Sample output**

| player_name | school_name | position | conference |
|---|---|---|---|
| Ralph Abernathy | Cincinnati | RB | American Athletic |
| Mekale McKay | Cincinnati | WR | American Athletic |

Solution:

select a.player_name,b.school_name,a.position,b.conference

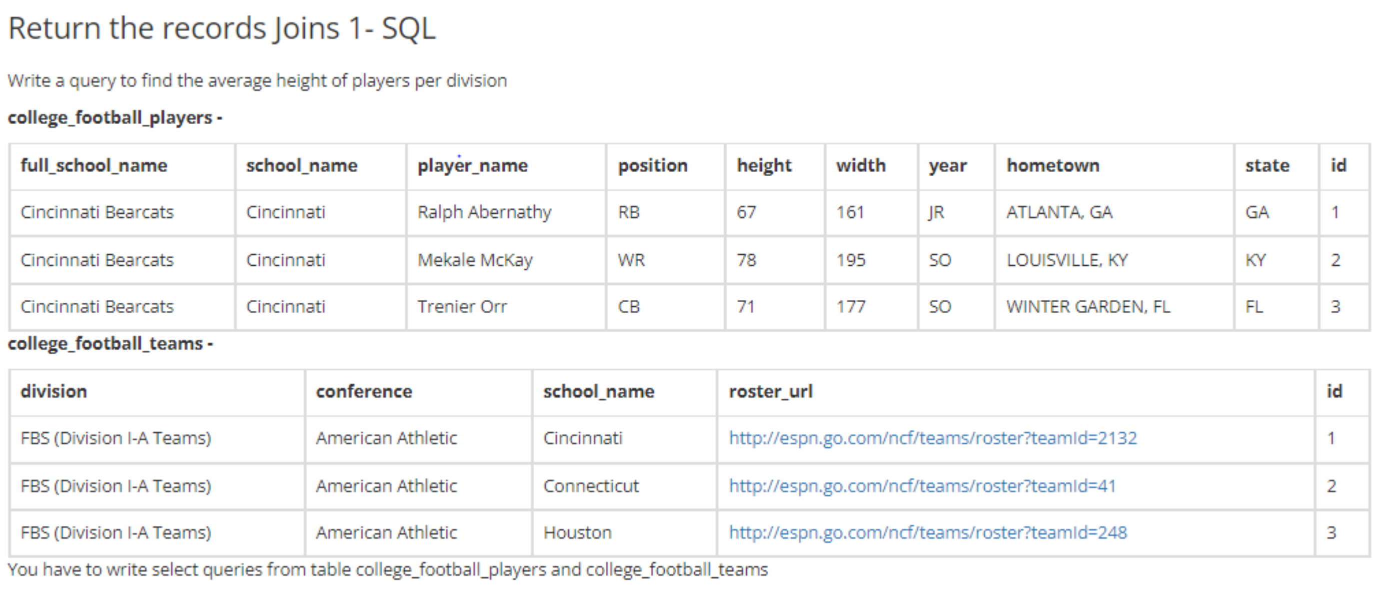
from college_football_players as a

inner join college_football_teams as b

on a.school_name = b.school_name;

## 2. Return the records Joins 1- SQL

Write a query to find the average height of players per division


Return the records Joins 1- SQL

Write a query to find the average height of players per division

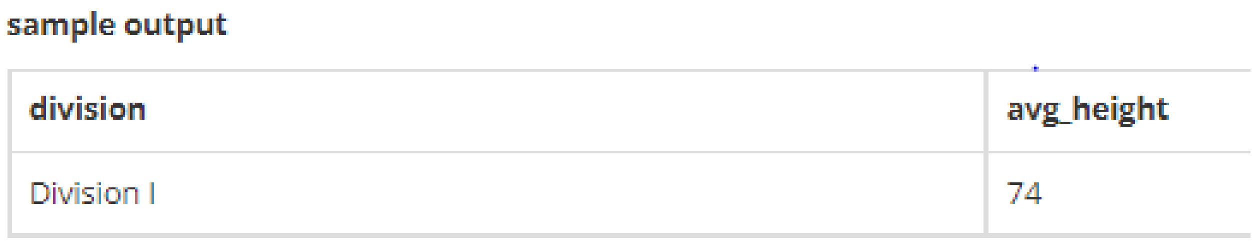
**college_football_players -**

| full_school_name | school_name | player_name | position | height | width | year | hometown | state | id |
|---|---|---|---|---|---|---|---|---|---|
| Cincinnati Bearcats | Cincinnati | Ralph Abernathy | RB | 67 | 161 | JR | ATLANTA, GA | GA | 1 |
| Cincinnati Bearcats | Cincinnati | Mekale McKay | WR | 78 | 195 | SO | LOUISVILLE, KY | KY | 2 |
| Cincinnati Bearcats | Cincinnati | Trenier Orr | CB | 71 | 177 | SO | WINTER GARDEN, FL | FL | 3 |

**college_football_teams -**

| division | conference | school_name | roster_url | id |
|---|---|---|---|---|
| FBS (Division I-A Teams) | American Athletic | Cincinnati | http://espn.go.com/ncf/teams/roster?teamId=2132 | 1 |
| FBS (Division I-A Teams) | American Athletic | Connecticut | http://espn.go.com/ncf/teams/roster?teamId=41 | 2 |
| FBS (Division I-A Teams) | American Athletic | Houston | http://espn.go.com/ncf/teams/roster?teamId=248 | 3 |

You have to write select queries from table college_football_players and college_football_teams


**sample output**

| division | avg_height |
|---|---|
| Division I | 74 |

Solution:

select b.division, avg(a.height) as avg_height

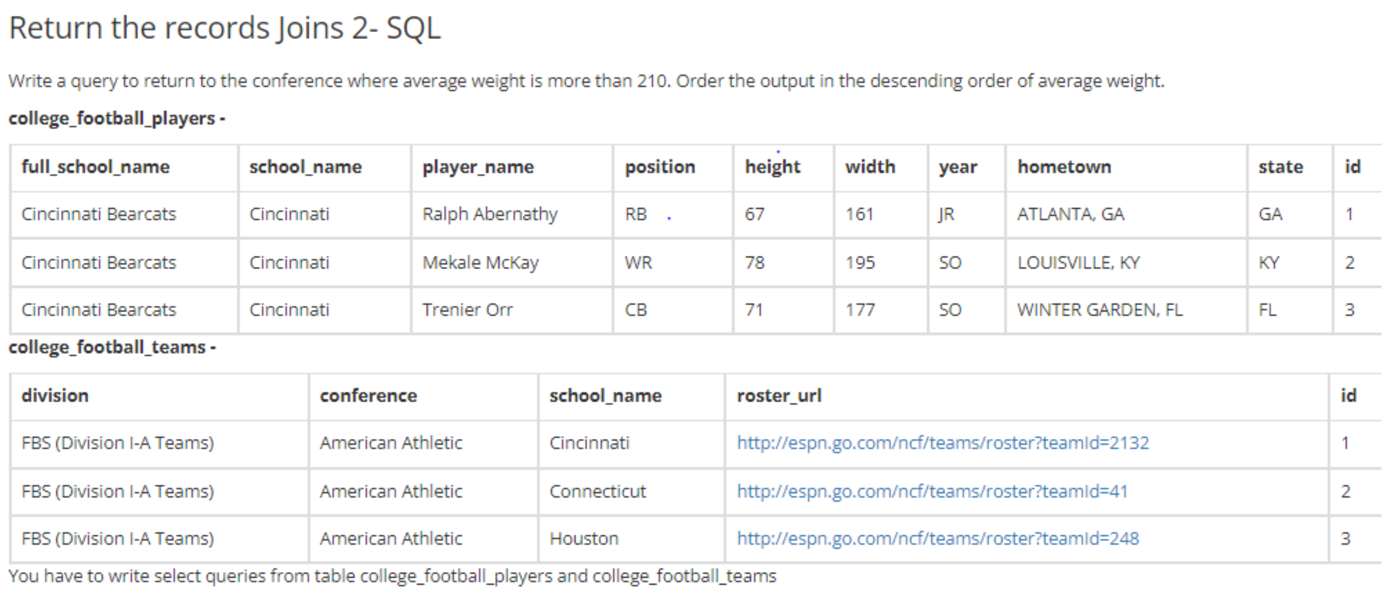
from college_football_players as a

join college_football_teams as b

on a.school_name = b.school_name;

### 3. Return the records Joins 2- SQL

Write a query to return to the conference where average weight is more than 210. Order the output in the descending order of average weight.



## Return the records Joins 2- SQL

Write a query to return to the conference where average weight is more than 210. Order the output in the descending order of average weight.

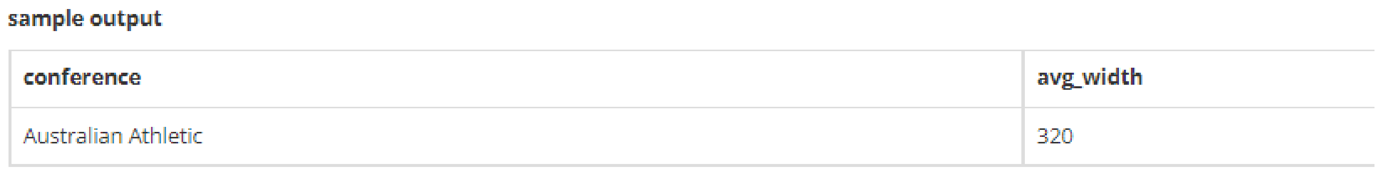
**college_football_players -**

| full_school_name | school_name | player_name | position | height | width | year | hometown | state | id |
|---|---|---|---|---|---|---|---|---|---|
| Cincinnati Bearcats | Cincinnati | Ralph Abernathy | RB | 67 | 161 | JR | ATLANTA, GA | GA | 1 |
| Cincinnati Bearcats | Cincinnati | Mekale McKay | WR | 78 | 195 | SO | LOUISVILLE, KY | KY | 2 |
| Cincinnati Bearcats | Cincinnati | Trenier Orr | CB | 71 | 177 | SO | WINTER GARDEN, FL | FL | 3 |

**college_football_teams -**

| division | conference | school_name | roster_url | id |
|---|---|---|---|---|
| FBS (Division I-A Teams) | American Athletic | Cincinnati | http://espn.go.com/ncf/teams/roster?teamId=2132 | 1 |
| FBS (Division I-A Teams) | American Athletic | Connecticut | http://espn.go.com/ncf/teams/roster?teamId=41 | 2 |
| FBS (Division I-A Teams) | American Athletic | Houston | http://espn.go.com/ncf/teams/roster?teamId=248 | 3 |

You have to write select queries from table college_football_players and college_football_teams



**sample output**

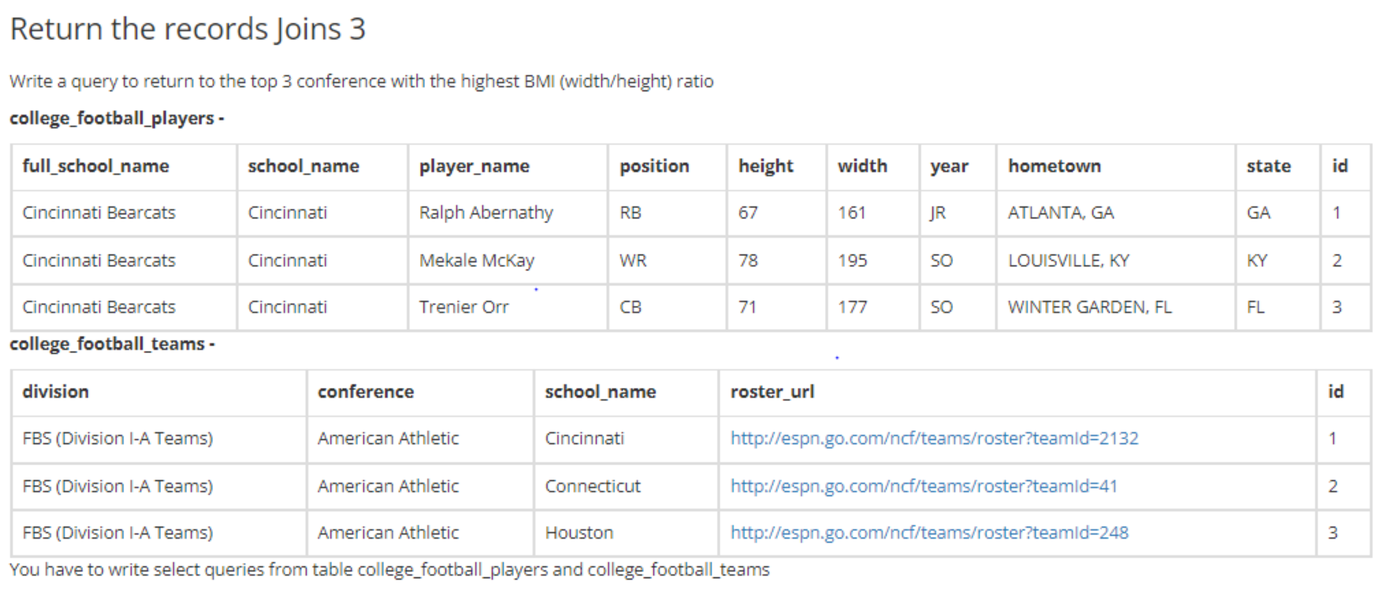
| conference | avg_width |
|---|---|
| Australian Athletic | 320 |

Solution:

```
select b.conference,avg(a.width) as avg_width

from college_football_players as a

join college_football_teams as b

on a.school_name = b.school_name

group by 1

having avg(a.width) > 210

order by 2 desc;
```

### 4. Return the records Joins 3

Write a query to return to the **top 3 conference with the highest BMI (width/height) ratio**



## Return the records Joins 3

Write a query to return to the top 3 conference with the highest BMI (width/height) ratio

**college_football_players -**

| full_school_name | school_name | player_name | position | height | width | year | hometown | state | id |
|---|---|---|---|---|---|---|---|---|---|
| Cincinnati Bearcats | Cincinnati | Ralph Abernathy | RB | 67 | 161 | JR | ATLANTA, GA | GA | 1 |
| Cincinnati Bearcats | Cincinnati | Mekale McKay | WR | 78 | 195 | SO | LOUISVILLE, KY | KY | 2 |
| Cincinnati Bearcats | Cincinnati | Trenier Orr | CB | 71 | 177 | SO | WINTER GARDEN, FL | FL | 3 |

**college_football_teams -**

| division | conference | school_name | roster_url | id |
|---|---|---|---|---|
| FBS (Division I-A Teams) | American Athletic | Cincinnati | http://espn.go.com/ncf/teams/roster?teamId=2132 | 1 |
| FBS (Division I-A Teams) | American Athletic | Connecticut | http://espn.go.com/ncf/teams/roster?teamId=41 | 2 |
| FBS (Division I-A Teams) | American Athletic | Houston | http://espn.go.com/ncf/teams/roster?teamId=248 | 3 |

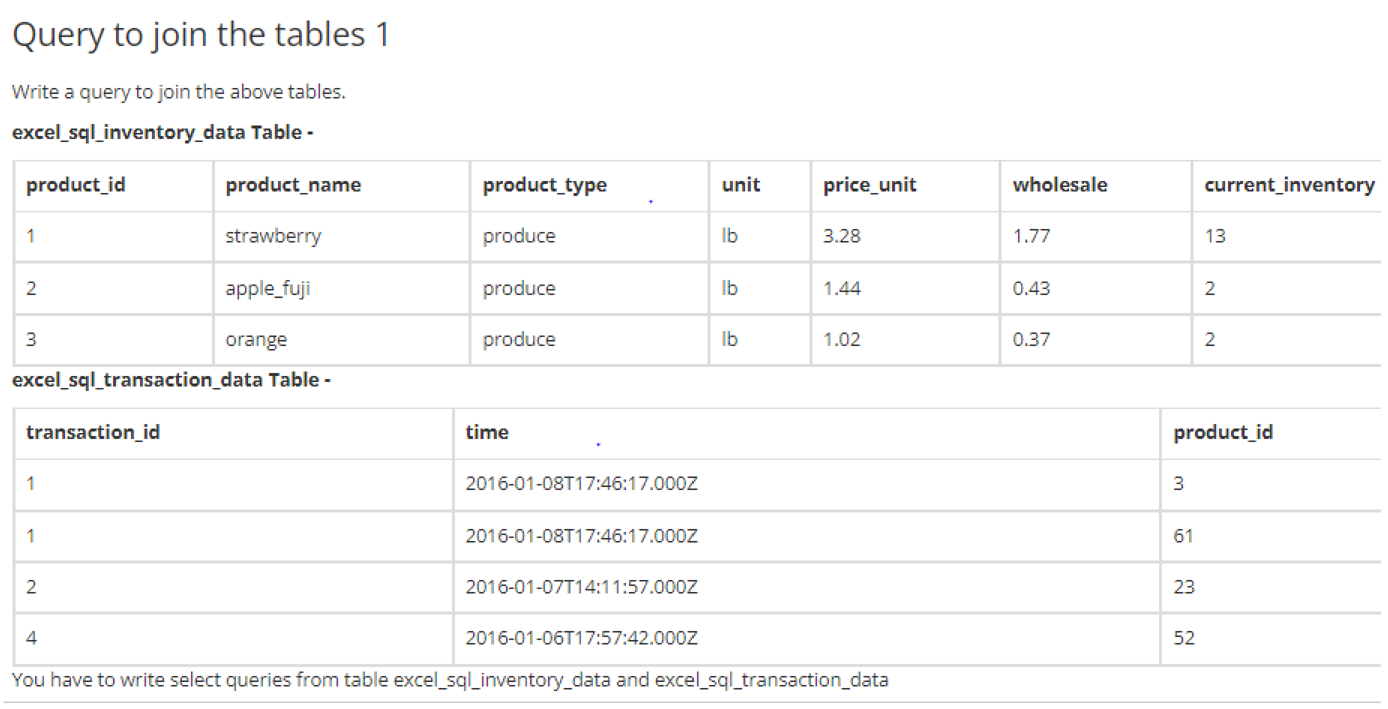
You have to write select queries from table college_football_players and college_football_teams

Solution:

```
select b.conference,sum(a.width) / sum(a.height) as bmi

from college_football_players as a

join college_football_teams as b

on a.school_name = b.school_name

group by 1

order by bmi desc

limit 3;
```

**5. Query to join the tables 1**

Write a query to join the above tables.



## Query to join the tables 1

Write a query to join the above tables.

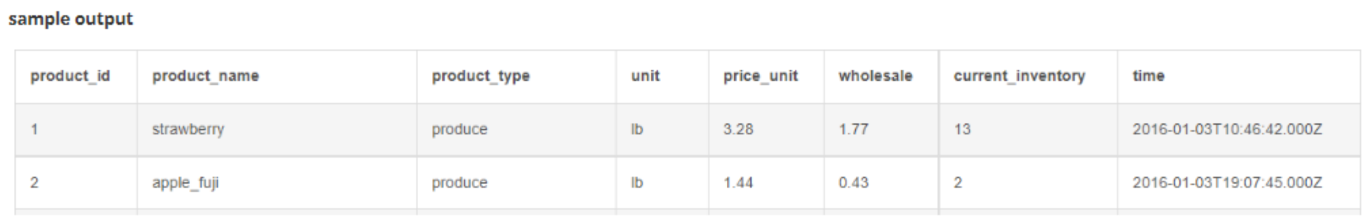
**excel_sql_inventory_data Table -**

| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory |
|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 |
| 3 | orange | produce | lb | 1.02 | 0.37 | 2 |

**excel_sql_transaction_data Table -**

| transaction_id | time | product_id |
|---|---|---|
| 1 | 2016-01-08T17:46:17.000Z | 3 |
| 1 | 2016-01-08T17:46:17.000Z | 61 |
| 2 | 2016-01-07T14:11:57.000Z | 23 |
| 4 | 2016-01-06T17:57:42.000Z | 52 |

You have to write select queries from table excel_sql_inventory_data and excel_sql_transaction_data



**sample output**

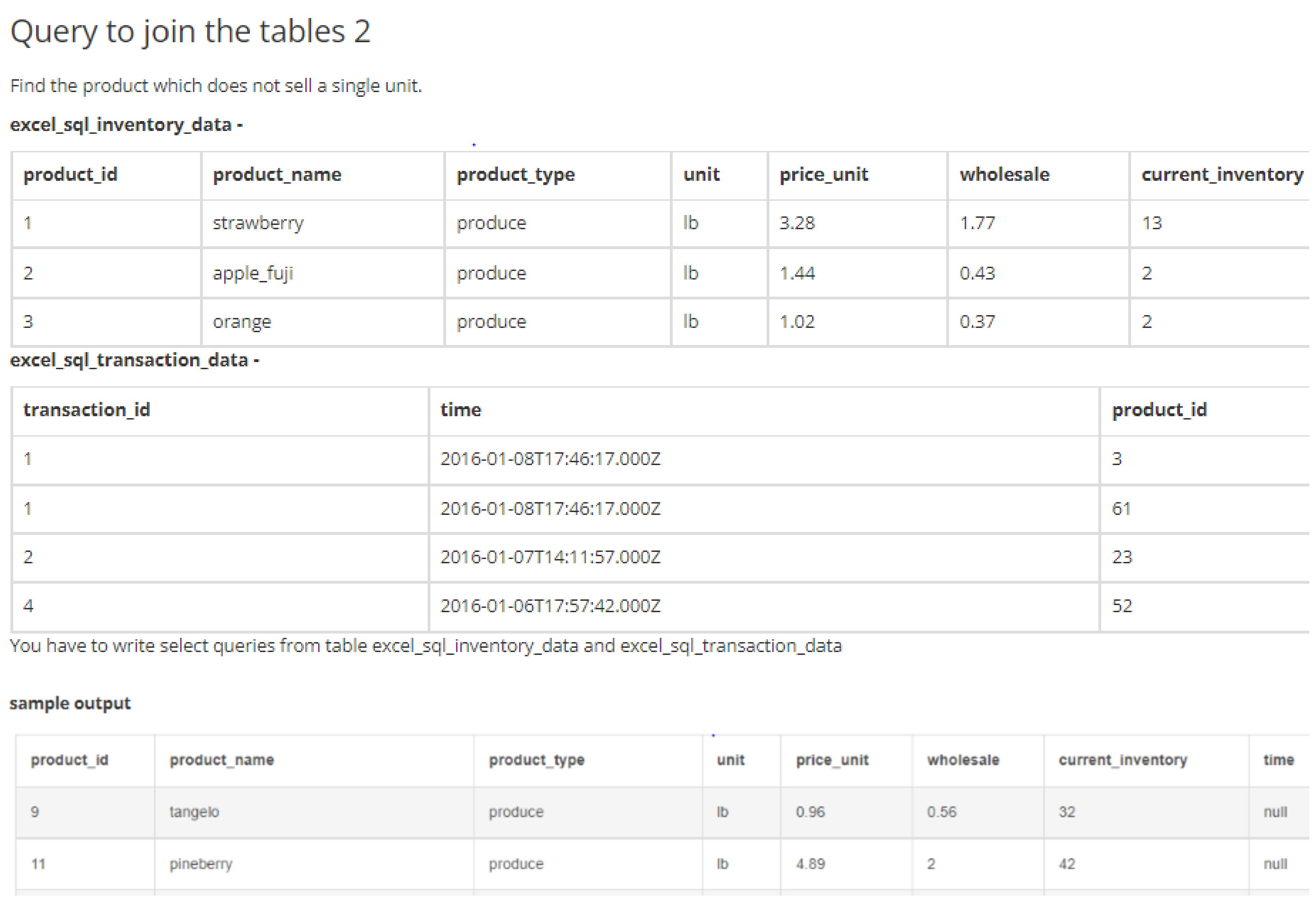
| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory | time |
|---|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 | 2016-01-03T10:46:42.000Z |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 | 2016-01-03T19:07:45.000Z |

Solution:

```
select
a.product_id,a.product_name,a.product_type,a.unit,a.price_unit,a.wholesale,a.current_inventc

from excel_sql_inventory_data as a

left join excel_sql_transaction_data as b

on a.product_id = b.product_id;
```

### 6. Query to join the tables 2

Find the product which **does not sell a single unit.**



## Query to join the tables 2

Find the product which does not sell a single unit.

**excel_sql_inventory_data -**

| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory |
|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 |
| 3 | orange | produce | lb | 1.02 | 0.37 | 2 |

**excel_sql_transaction_data -**

| transaction_id | time | product_id |
|---|---|---|
| 1 | 2016-01-08T17:46:17.000Z | 3 |
| 1 | 2016-01-08T17:46:17.000Z | 61 |
| 2 | 2016-01-07T14:11:57.000Z | 23 |
| 4 | 2016-01-06T17:57:42.000Z | 52 |

You have to write select queries from table excel_sql_inventory_data and excel_sql_transaction_data

**sample output**

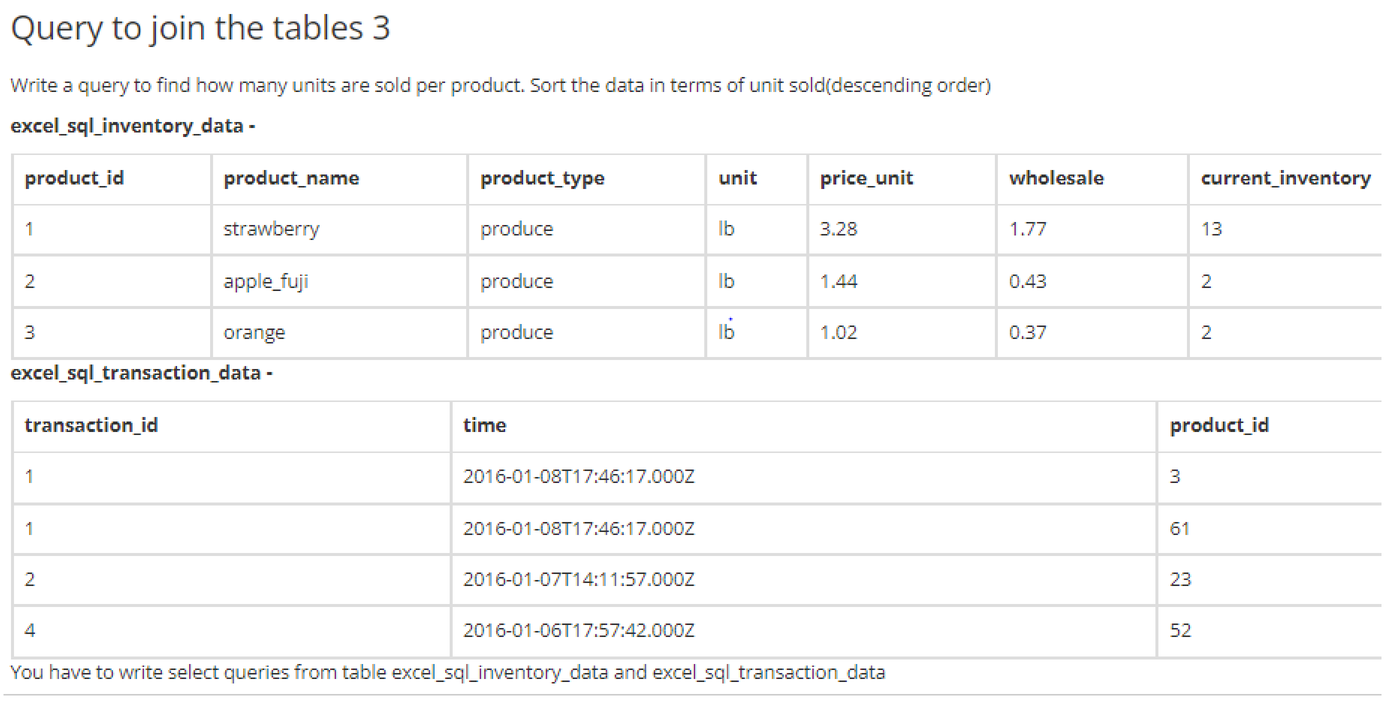
| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory | time |
|---|---|---|---|---|---|---|---|
| 9 | tangelo | produce | lb | 0.96 | 0.56 | 32 | null |
| 11 | pineberry | produce | lb | 4.89 | 2 | 42 | null |

solution:

```
select
a.product_id,a.product_name,a.product_type,a.unit,a.price_unit,a.wholesale,a.current_inventc

from excel_sql_inventory_data as a

left join excel_sql_transaction_data as b

on a.product_id = b.product_id

where b.time is null;
```

### 7. Query to join the tables 3

Write a query to find how many units are sold per product. Sort the data in terms of unit sold(descending order)



## Query to join the tables 3

Write a query to find how many units are sold per product. Sort the data in terms of unit sold(descending order)

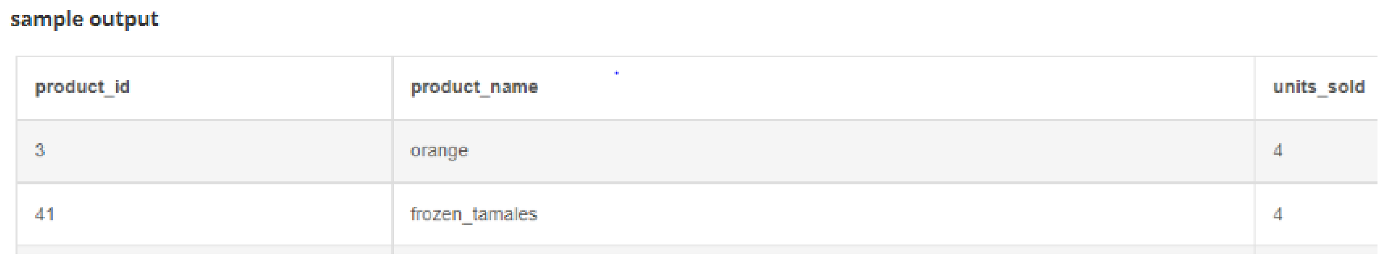
**excel_sql_inventory_data -**

| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory |
|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 |
| 3 | orange | produce | lb | 1.02 | 0.37 | 2 |

**excel_sql_transaction_data -**

| transaction_id | time | product_id |
|---|---|---|
| 1 | 2016-01-08T17:46:17.000Z | 3 |
| 1 | 2016-01-08T17:46:17.000Z | 61 |
| 2 | 2016-01-07T14:11:57.000Z | 23 |
| 4 | 2016-01-06T17:57:42.000Z | 52 |

You have to write select queries from table excel_sql_inventory_data and excel_sql_transaction_data



**sample output**

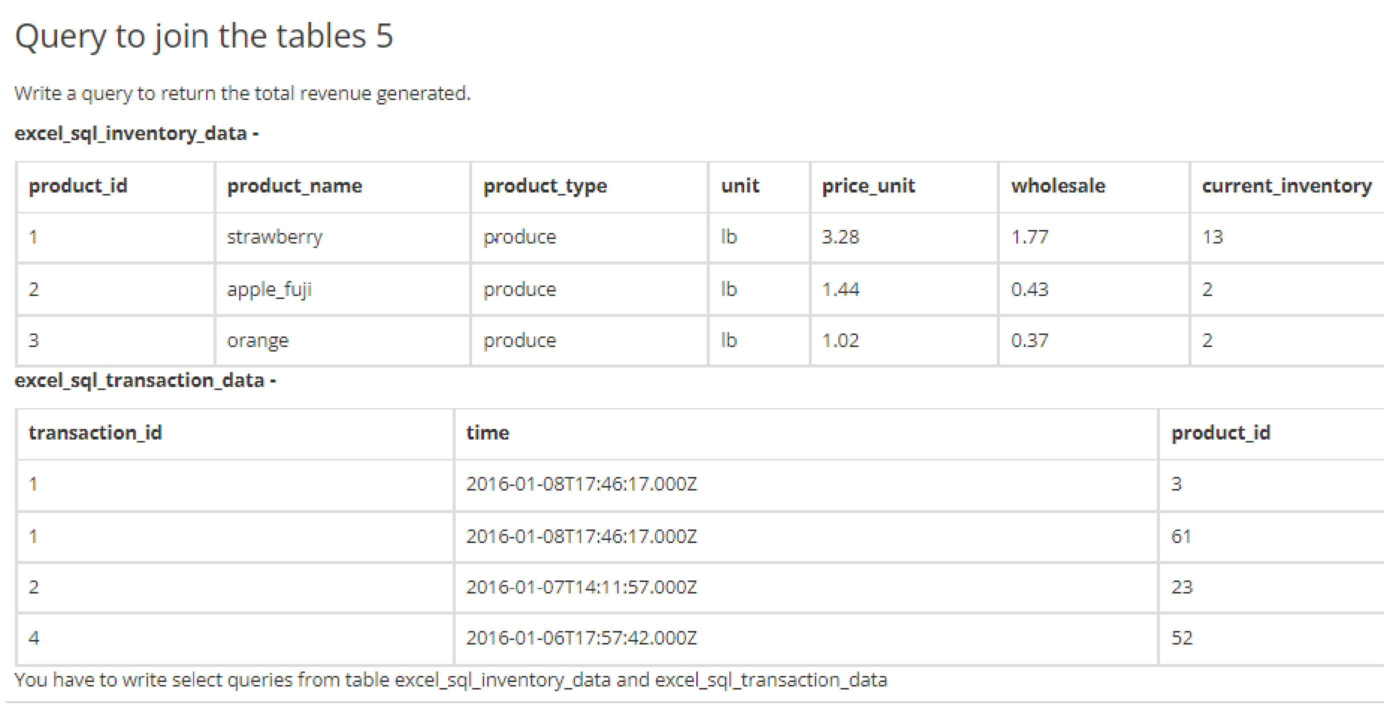
| product_id | product_name | units_sold |
|---|---|---|
| 3 | orange | 4 |
| 41 | frozen_tamales | 4 |

solution:

```
select a.product_id,a.product_name,count(b.time) as unit_sold

from excel_sql_inventory_data as a

left join excel_sql_transaction_data as b

on a.product_id = b.product_id

group by a.product_id, a.product_name

order by unit_sold desc;
```

### 8. Query to join the tables 5

Write a query to return the total revenue generated.



## Query to join the tables 5

Write a query to return the total revenue generated.

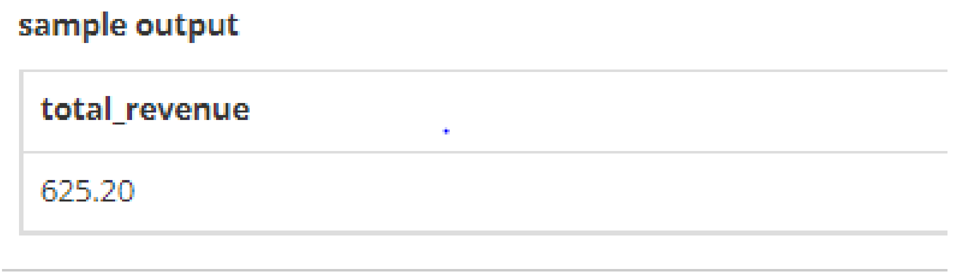
**excel_sql_inventory_data -**

| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory |
|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 |
| 3 | orange | produce | lb | 1.02 | 0.37 | 2 |

**excel_sql_transaction_data -**

| transaction_id | time | product_id |
|---|---|---|
| 1 | 2016-01-08T17:46:17.000Z | 3 |
| 1 | 2016-01-08T17:46:17.000Z | 61 |
| 2 | 2016-01-07T14:11:57.000Z | 23 |
| 4 | 2016-01-06T17:57:42.000Z | 52 |

You have to write select queries from table excel_sql_inventory_data and excel_sql_transaction_data



**sample output**

| total_revenue |
|---|
| 625.20 |

solution:

select sum(a.price_unit) as total_revenue

from excel_sql_inventory_data as a

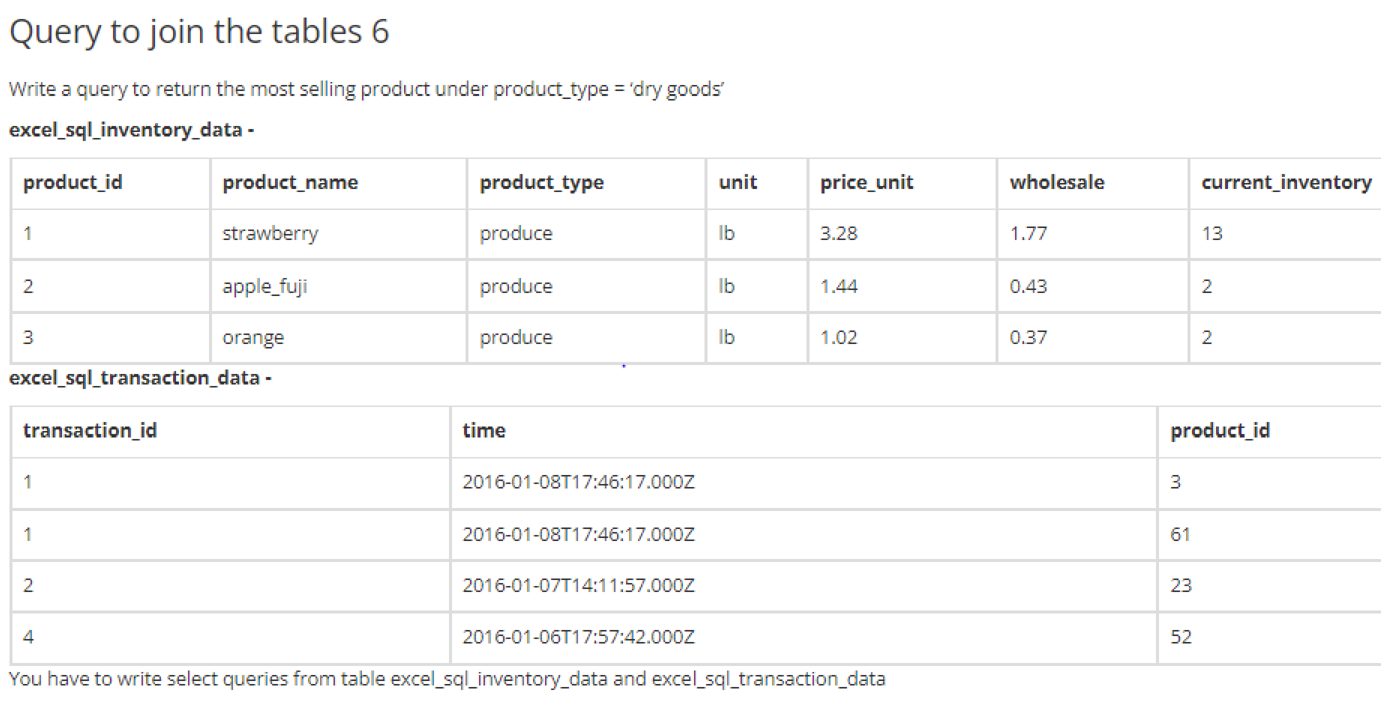
left join excel_sql_transaction_data as b

on a.product_id = b.product_id

where b.time is not null;

**9. Query to join the tables 6**

Write a query to return the most selling product under product_type = ‘dry goods’



Query to join the tables 6

Write a query to return the most selling product under product_type = 'dry goods'

**excel_sql_inventory_data -**

| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory |
|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 |
| 3 | orange | produce | lb | 1.02 | 0.37 | 2 |

**excel_sql_transaction_data -**

| transaction_id | time | product_id |
|---|---|---|
| 1 | 2016-01-08T17:46:17.000Z | 3 |
| 1 | 2016-01-08T17:46:17.000Z | 61 |
| 2 | 2016-01-07T14:11:57.000Z | 23 |
| 4 | 2016-01-06T17:57:42.000Z | 52 |

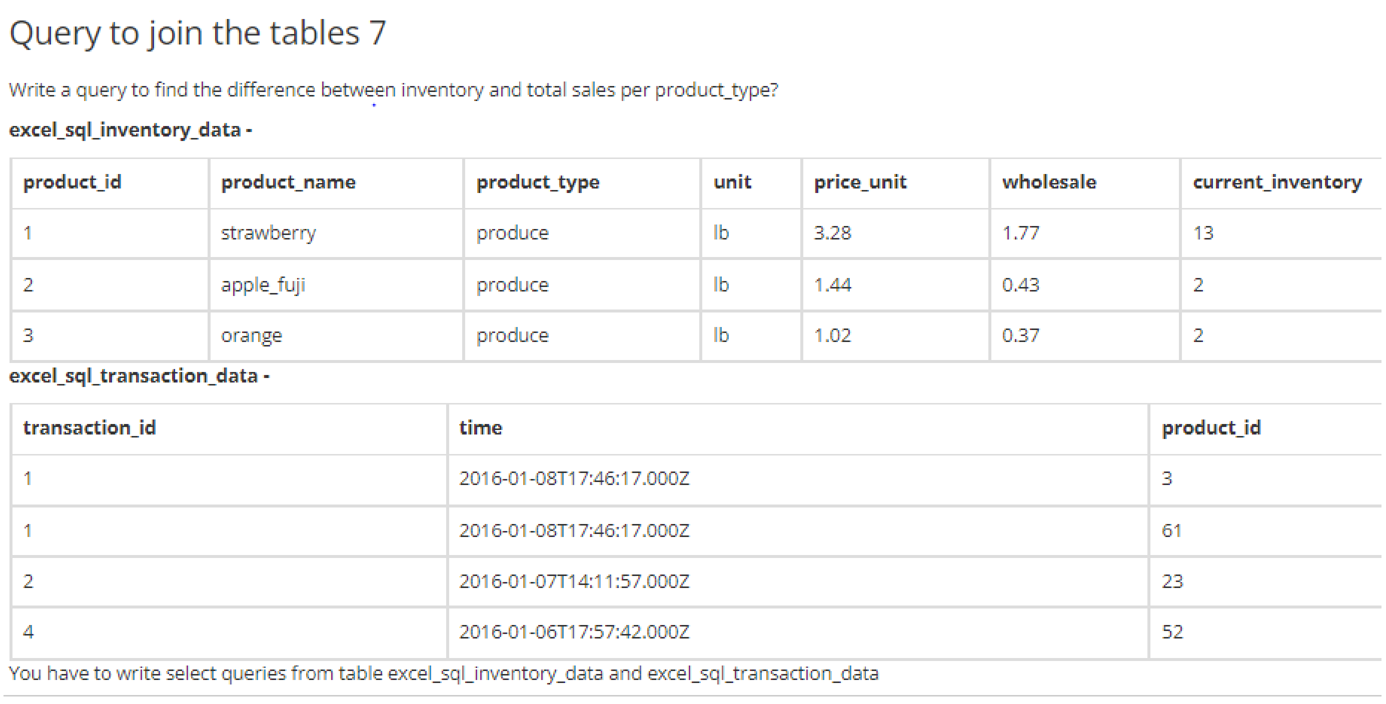
You have to write select queries from table excel_sql_inventory_data and excel_sql_transaction_data

solution:

```
SELECT a.product_name, COUNT(b.time) AS unit_sold

FROM excel_sql_inventory_data a

LEFT JOIN excel_sql_transaction_data b

ON a.product_id = b.product_id

GROUP BY product_name

ORDER BY unit_sold DESC

LIMIT 1;
```

**10. Query to join the tables 7**

Write a query to **find the difference between inventory and total sales per product_type?**



## Query to join the tables 7

Write a query to find the difference between inventory and total sales per product_type?

**excel_sql_inventory_data -**

| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory |
|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 |
| 3 | orange | produce | lb | 1.02 | 0.37 | 2 |

**excel_sql_transaction_data -**

| transaction_id | time | product_id |
|---|---|---|
| 1 | 2016-01-08T17:46:17.000Z | 3 |
| 1 | 2016-01-08T17:46:17.000Z | 61 |
| 2 | 2016-01-07T14:11:57.000Z | 23 |
| 4 | 2016-01-06T17:57:42.000Z | 52 |

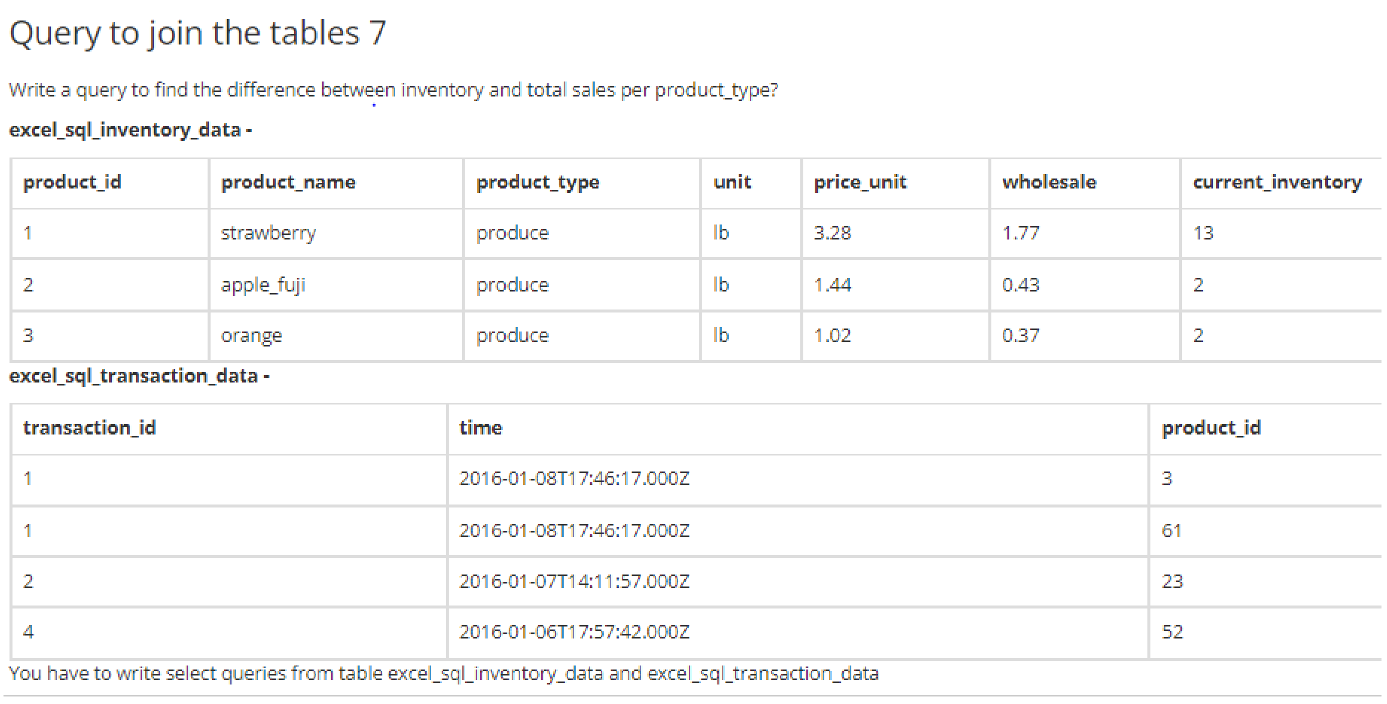
You have to write select queries from table excel_sql_inventory_data and excel_sql_transaction_data

solution:

```
SELECT a.product_type,SUM(current_inventory) - COUNT(b.time) AS delta

FROM excel_sql_inventory_data a

LEFT JOIN excel_sql_transaction_data b

ON a.product_id = b.product_id

GROUP BY product_type

ORDER BY delta DESC;
```

### 11. Query to join the tables 8

Find the product-wise sales for product_type ='dairy'



## Query to join the tables 7

Write a query to find the difference between inventory and total sales per product_type?

**excel_sql_inventory_data -**

| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory |
|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 |
| 3 | orange | produce | lb | 1.02 | 0.37 | 2 |

**excel_sql_transaction_data -**

| transaction_id | time | product_id |
|---|---|---|
| 1 | 2016-01-08T17:46:17.000Z | 3 |
| 1 | 2016-01-08T17:46:17.000Z | 61 |
| 2 | 2016-01-07T14:11:57.000Z | 23 |
| 4 | 2016-01-06T17:57:42.000Z | 52 |

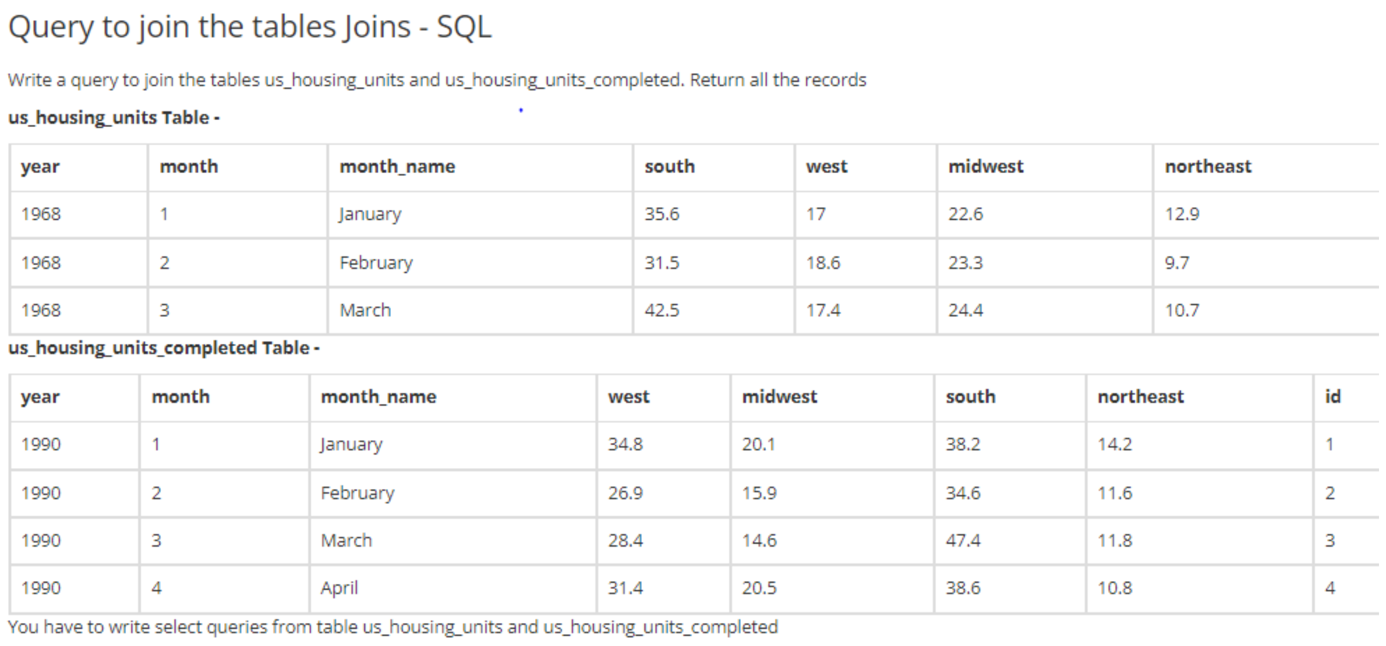
You have to write select queries from table excel_sql_inventory_data and excel_sql_transaction_data

solution:

```
SELECT a.product_name,SUM(a.price_unit)*COUNT(b.time) AS sales

FROM excel_sql_inventory_data a

LEFT JOIN excel_sql_transaction_data b

ON a.product_id = b.product_id

WHERE product_type = 'dairy'

GROUP BY product_name

ORDER BY sales DESC
```

### 12. Query to join the tables Joins - SQL

Write a query to join the tables us_housing_units and us_housing_units_completed. Return all the records



Query to join the tables Joins - SQL

Write a query to join the tables us_housing_units and us_housing_units_completed. Return all the records

**us_housing_units Table -**

| year | month | month_name | south | west | midwest | northeast |
|---|---|---|---|---|---|---|
| 1968 | 1 | January | 35.6 | 17 | 22.6 | 12.9 |
| 1968 | 2 | February | 31.5 | 18.6 | 23.3 | 9.7 |
| 1968 | 3 | March | 42.5 | 17.4 | 24.4 | 10.7 |

**us_housing_units_completed Table -**

| year | month | month_name | west | midwest | south | northeast | id |
|---|---|---|---|---|---|---|---|
| 1990 | 1 | January | 34.8 | 20.1 | 38.2 | 14.2 | 1 |
| 1990 | 2 | February | 26.9 | 15.9 | 34.6 | 11.6 | 2 |
| 1990 | 3 | March | 28.4 | 14.6 | 47.4 | 11.8 | 3 |
| 1990 | 4 | April | 31.4 | 20.5 | 38.6 | 10.8 | 4 |

You have to write select queries from table us_housing_units and us_housing_units_completed

solution:

SELECT a.year, a.month, a.month_name, a.south AS south_unit, a.west AS west_unit, a.midwest AS midwest_unit, a.northeast AS northeast_unit, b.south AS south_completed, a.west AS west_completed, a.midwest AS midwest_completed, a.northeast AS northeast_completed

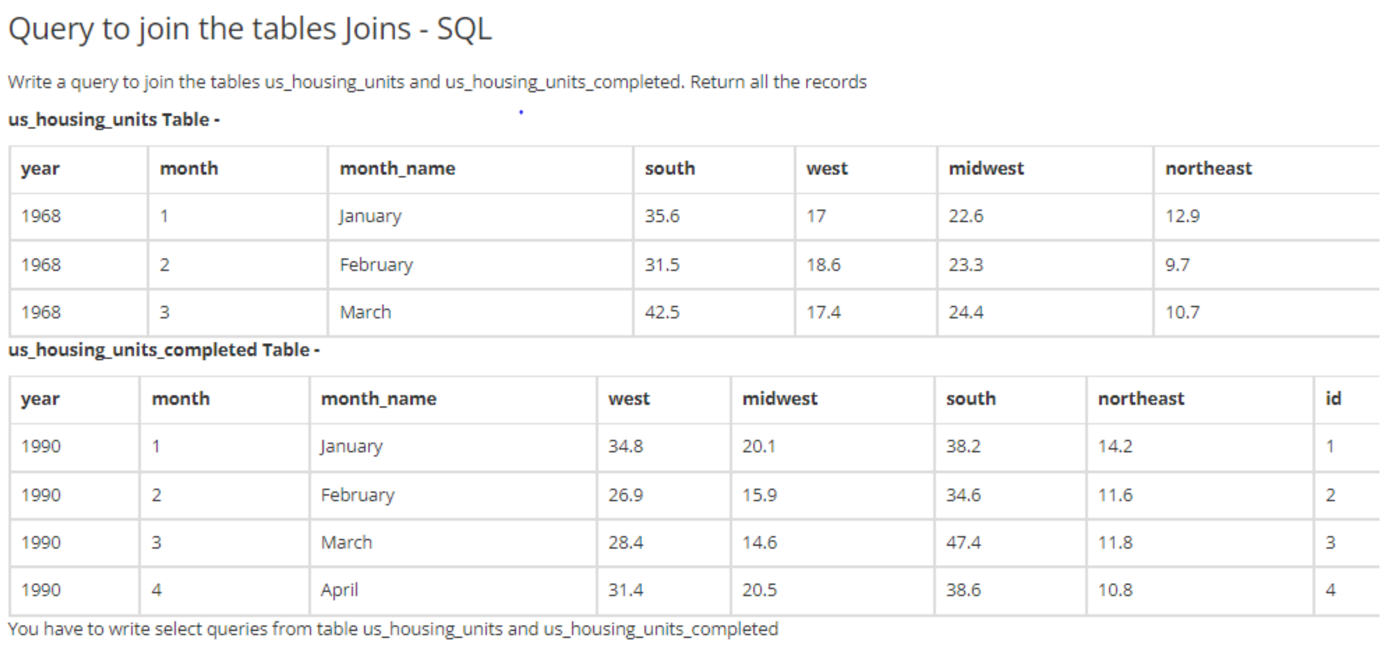
FROM us_housing_units a

LEFT JOIN us_housing_units_completed b

ON a.year = b.year AND a.month = b.month

**13. Query to return the records Joins - SQL**

Write a query to return year, month, month_name and difference between the units and units completed for west from 2000 onwards.



## Query to join the tables Joins - SQL

Write a query to join the tables us_housing_units and us_housing_units_completed. Return all the records

**us_housing_units Table -**

| year | month | month_name | south | west | midwest | northeast |
|---|---|---|---|---|---|---|
| 1968 | 1 | January | 35.6 | 17 | 22.6 | 12.9 |
| 1968 | 2 | February | 31.5 | 18.6 | 23.3 | 9.7 |
| 1968 | 3 | March | 42.5 | 17.4 | 24.4 | 10.7 |

**us_housing_units_completed Table -**

| year | month | month_name | west | midwest | south | northeast | id |
|---|---|---|---|---|---|---|---|
| 1990 | 1 | January | 34.8 | 20.1 | 38.2 | 14.2 | 1 |
| 1990 | 2 | February | 26.9 | 15.9 | 34.6 | 11.6 | 2 |
| 1990 | 3 | March | 28.4 | 14.6 | 47.4 | 11.8 | 3 |
| 1990 | 4 | April | 31.4 | 20.5 | 38.6 | 10.8 | 4 |

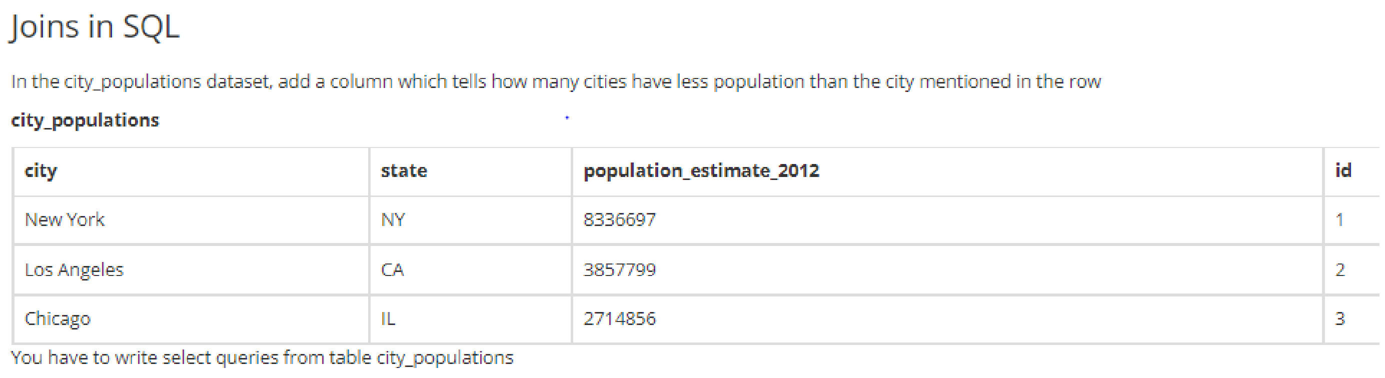
You have to write select queries from table us_housing_units and us_housing_units_completed

solution:

```
SELECT a.year,a.month,a.month_name,a.west,b.west,a.west - b.west AS
difference_in_units

FROM us_housing_units a

LEFT JOIN us_housing_units_completed b

ON a.year = b.year AND a.month = b.month

WHERE a.year > 2000
```

**14. Joins in SQL**

In the city_populations dataset, **add a column which tells how many cities have less population than the city mentioned in the row**



## Joins in SQL

In the city_populations dataset, add a column which tells how many cities have less population than the city mentioned in the row

**city_populations**

| city | state | population_estimate_2012 | id |
|---|---|---|---|
| New York | NY | 8336697 | 1 |
| Los Angeles | CA | 3857799 | 2 |
| Chicago | IL | 2714856 | 3 |

You have to write select queries from table city_populations

solution:

```
SELECT a.city,a.state,a.population_estimate_2012,COUNT(b.city) AS
num_city_with_higher_population
```

FROM city_populations a

JOIN city_populations b

ON a.population_estimate_2012 > b.population_estimate_2012

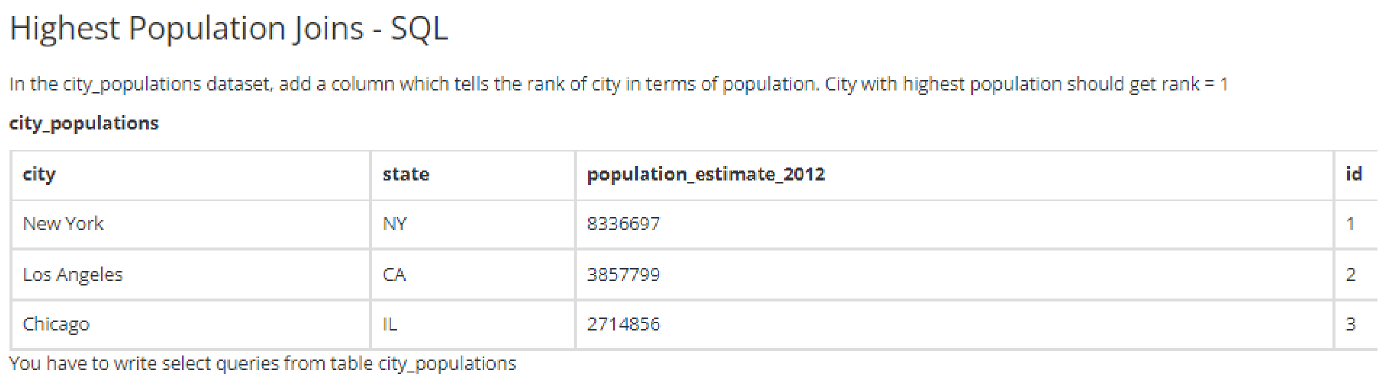
GROUP BY a.city, a.state,A.population_estimate_2012

```
In [ ]:
```

## SQL Logic Building - 4

### 1. Highest Population Joins - SQL

In the city_populations dataset, add a column which tells the **rank of city in terms of population. City with highest population should get rank = 1**



Highest Population Joins - SQL

In the city_populations dataset, add a column which tells the rank of city in terms of population. City with highest population should get rank = 1

**city_populations**

| city | state | population_estimate_2012 | id |
|---|---|---|---|
| New York | NY | 8336697 | 1 |
| Los Angeles | CA | 3857799 | 2 |
| Chicago | IL | 2714856 | 3 |

You have to write select queries from table city_populations

solution:

SELECT a.city,a.state,a.population_estimate_2012,COUNT(b.city) AS rank

FROM city_populations a

JOIN city_populations b

ON a.population_estimate_2012 <= b.population_estimate_2012

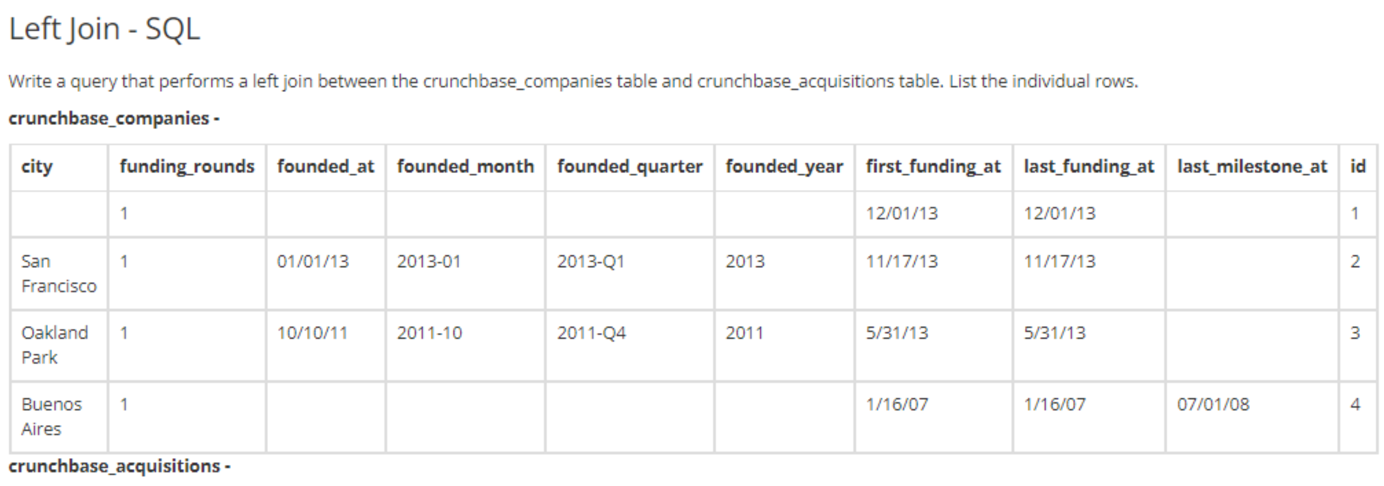
GROUP BY a.city,a.state, a.population_estimate_2012

ORDER BY rank

### 2. Left Join - SQL

Write a query that performs a left join between the crunchbase_companies table and crunchbase_acquisitions table. List the individual rows.

Table No.1 -->crunchbase_companies -



## Left Join - SQL

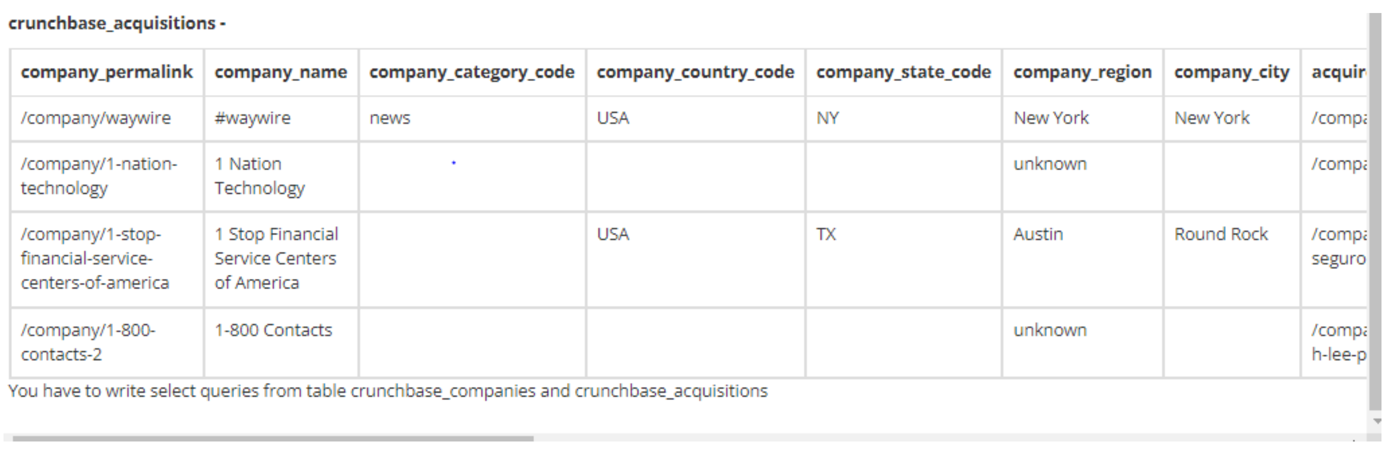
Write a query that performs a left join between the crunchbase_companies table and crunchbase_acquisitions table. List the individual rows.

**crunchbase_companies -**

| city | funding_rounds | founded_at | founded_month | founded_quarter | founded_year | first_funding_at | last_funding_at | last_milestone_at | id |
|---|---|---|---|---|---|---|---|---|---|
|  | 1 |  |  |  |  | 12/01/13 | 12/01/13 |  | 1 |
| San Francisco | 1 | 01/01/13 | 2013-01 | 2013-Q1 | 2013 | 11/17/13 | 11/17/13 |  | 2 |
| Oakland Park | 1 | 10/10/11 | 2011-10 | 2011-Q4 | 2011 | 5/31/13 | 5/31/13 |  | 3 |
| Buenos Aires | 1 |  |  |  |  | 1/16/07 | 1/16/07 | 07/01/08 | 4 |

**crunchbase_acquisitions -**

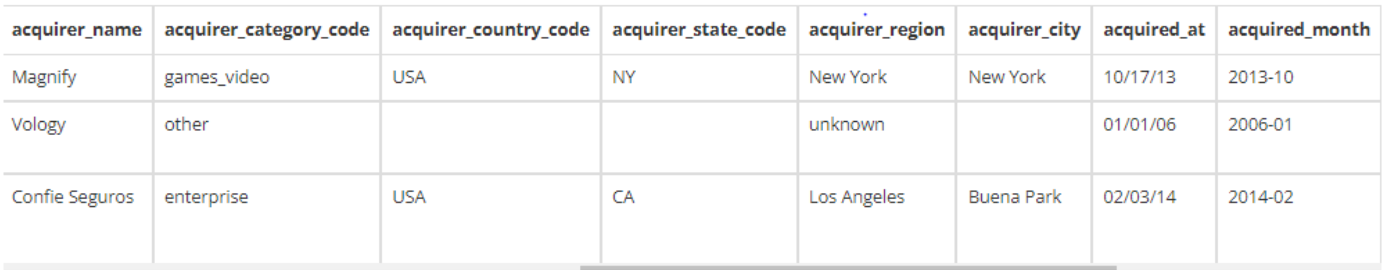
Table No.2 --> crunchbase_acquisitions -



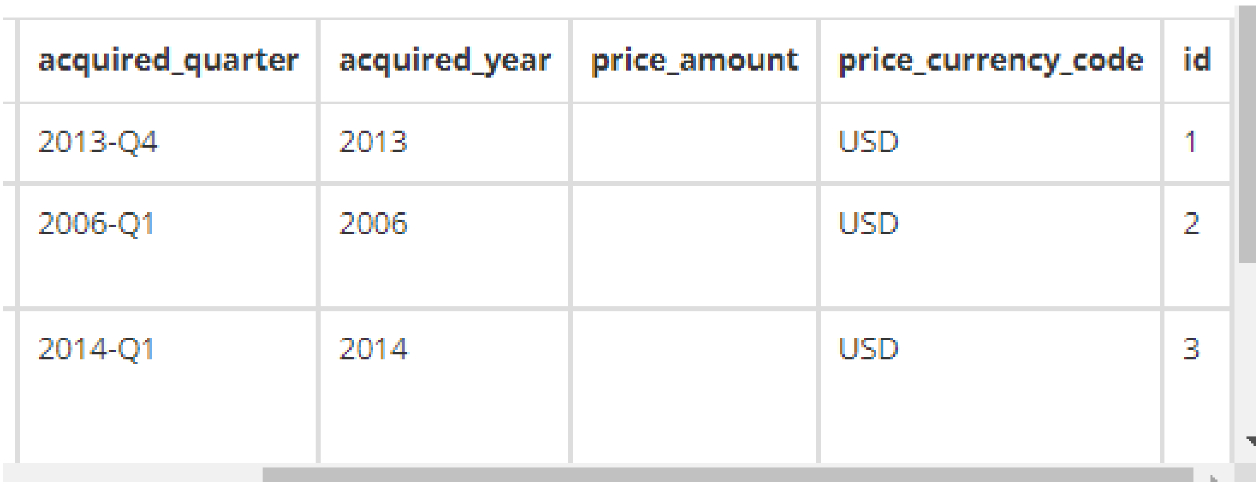
**crunchbase_acquisitions -**

| company_permalink | company_name | company_category_code | company_country_code | company_state_code | company_region | company_city | acquir |
|---|---|---|---|---|---|---|---|
| /company/waywire | #waywire | news | USA | NY | New York | New York | /compa |
| /company/1-nation-technology | 1 Nation Technology |  |  |  | unknown |  | /compa |
| /company/1-stop-financial-service-centers-of-america | 1 Stop Financial Service Centers of America |  | USA | TX | Austin | Round Rock | /compa seguro |
| /company/1-800-contacts-2 | 1-800 Contacts |  |  |  | unknown |  | /compa h-lee-p |

You have to write select queries from table crunchbase_companies and crunchbase_acquisitions



| acquirer_name | acquirer_category_code | acquirer_country_code | acquirer_state_code | acquirer_region | acquirer_city | acquired_at | acquired_month |
|---|---|---|---|---|---|---|---|
| Magnify | games_video | USA | NY | New York | New York | 10/17/13 | 2013-10 |
| Vology | other |  |  | unknown |  | 01/01/06 | 2006-01 |
| Confie Seguros | enterprise | USA | CA | Los Angeles | Buena Park | 02/03/14 | 2014-02 |



| acquired_quarter | acquired_year | price_amount | price_currency_code | id |
|---|---|---|---|---|
| 2013-Q4 | 2013 |  | USD | 1 |
| 2006-Q1 | 2006 |  | USD | 2 |
| 2014-Q1 | 2014 |  | USD | 3 |

solution:

SELECT *

FROM crunchbase_companies as a

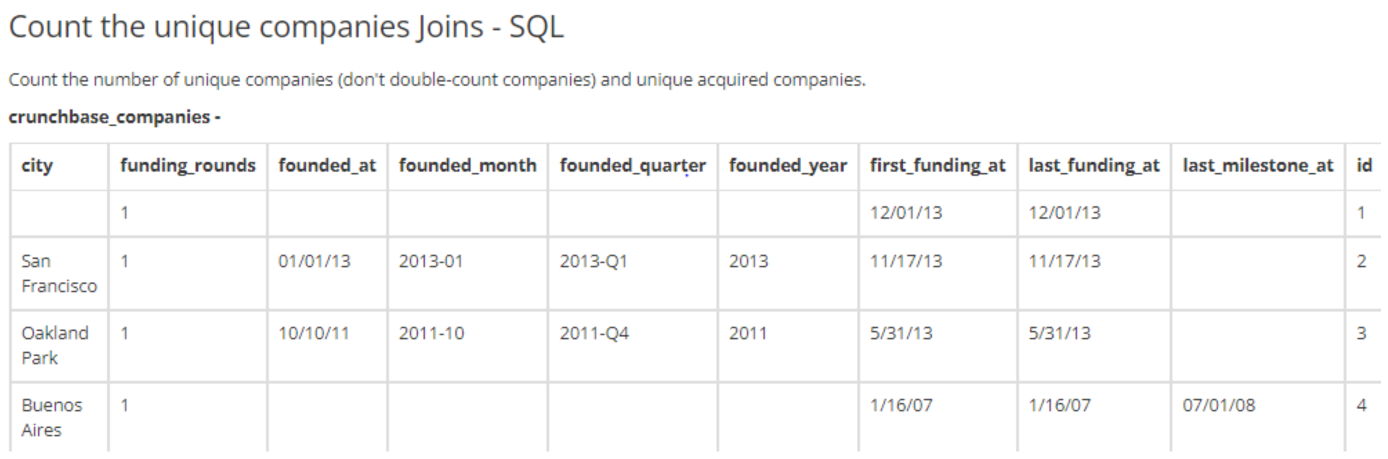
LEFT JOIN crunchbase_acquisitions as b

ON a.permalink = b.company_permalink

### 3. Count the unique companies Joins - SQL

**Count the number of unique companies** (don't double-count companies) and unique acquired companies.

Table No.1 -->crunchbase_companies -



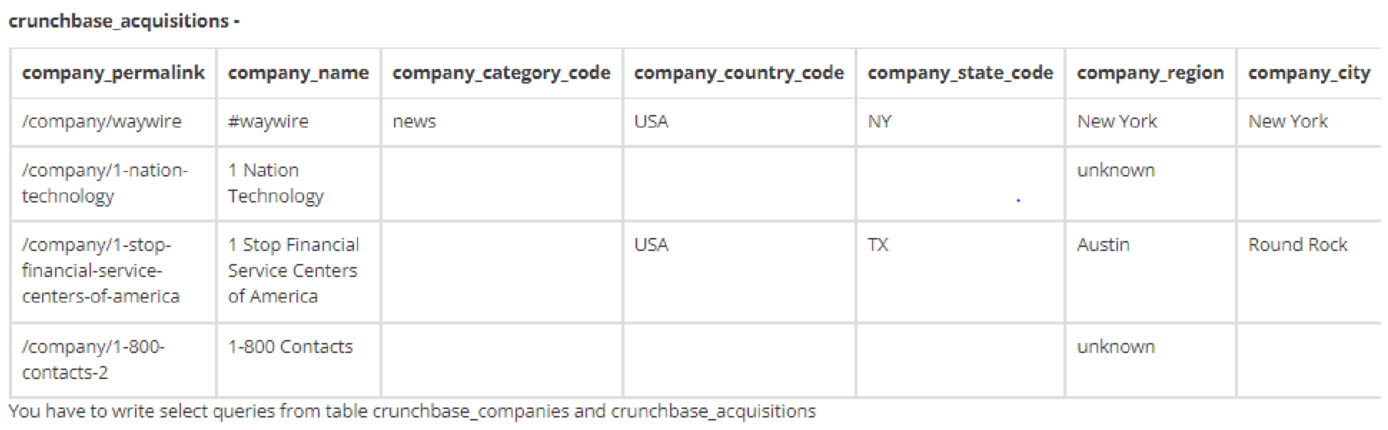
Count the unique companies Joins - SQL

Count the number of unique companies (don't double-count companies) and unique acquired companies.

**crunchbase_companies -**

| city | funding_rounds | founded_at | founded_month | founded_quarter | founded_year | first_funding_at | last_funding_at | last_milestone_at | id |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | | | 12/01/13 | 12/01/13 | | 1 |
| San Francisco | 1 | 01/01/13 | 2013-01 | 2013-Q1 | 2013 | 11/17/13 | 11/17/13 | | 2 |
| Oakland Park | 1 | 10/10/11 | 2011-10 | 2011-Q4 | 2011 | 5/31/13 | 5/31/13 | | 3 |
| Buenos Aires | 1 | | | | | 1/16/07 | 1/16/07 | 07/01/08 | 4 |

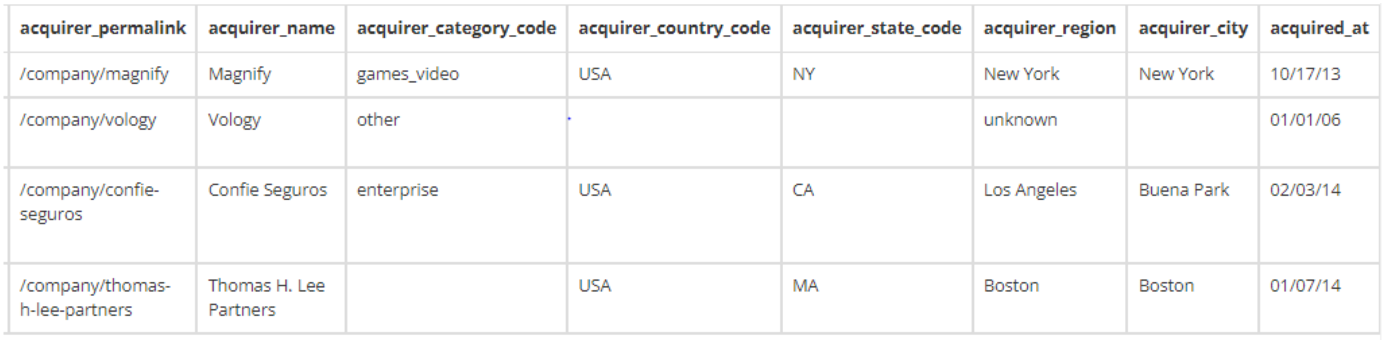
Table No.2 --> crunchbase_acquisitions -



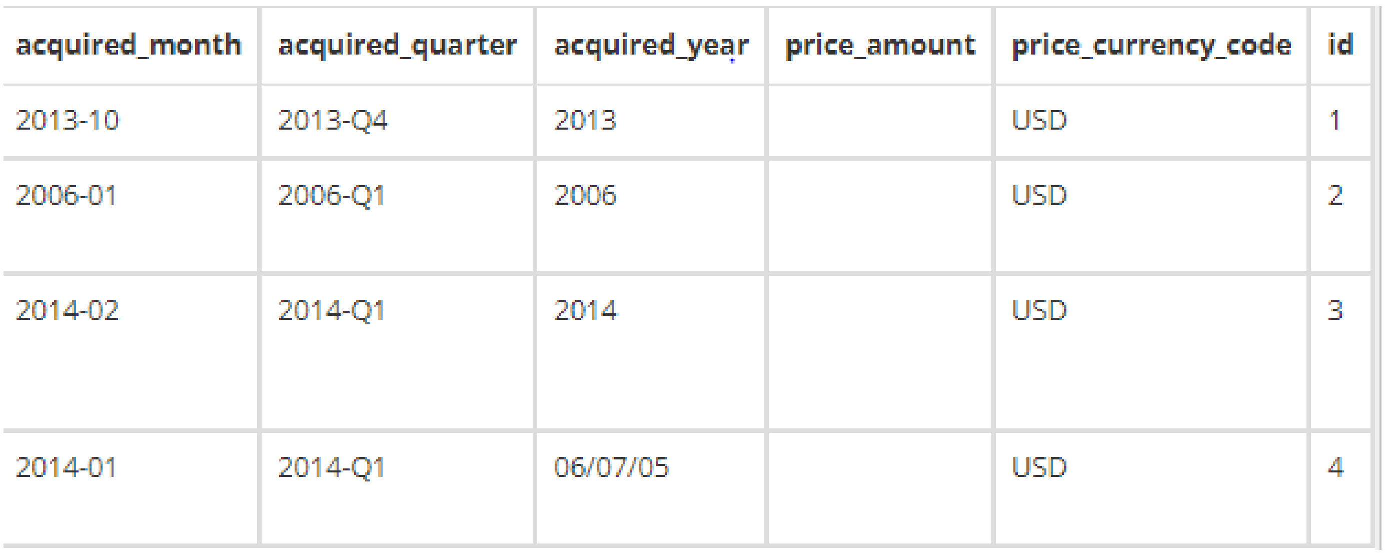
**crunchbase_acquisitions -**

| company_permalink | company_name | company_category_code | company_country_code | company_state_code | company_region | company_city |
|---|---|---|---|---|---|---|
| /company/waywire | #waywire | news | USA | NY | New York | New York |
| /company/1-nation-technology | 1 Nation Technology | | | | unknown | |
| /company/1-stop-financial-service-centers-of-america | 1 Stop Financial Service Centers of America | | USA | TX | Austin | Round Rock |
| /company/1-800-contacts-2 | 1-800 Contacts | | | | unknown | |

You have to write select queries from table crunchbase_companies and crunchbase_acquisitions



| acquirer_permalink | acquirer_name | acquirer_category_code | acquirer_country_code | acquirer_state_code | acquirer_region | acquirer_city | acquired_at |
|---|---|---|---|---|---|---|---|
| /company/magnify | Magnify | games_video | USA | NY | New York | New York | 10/17/13 |
| /company/vology | Vology | other | | | unknown | | 01/01/06 |
| /company/confie-seguros | Confie Seguros | enterprise | USA | CA | Los Angeles | Buena Park | 02/03/14 |
| /company/thomas-h-lee-partners | Thomas H. Lee Partners | | USA | MA | Boston | Boston | 01/07/14 |



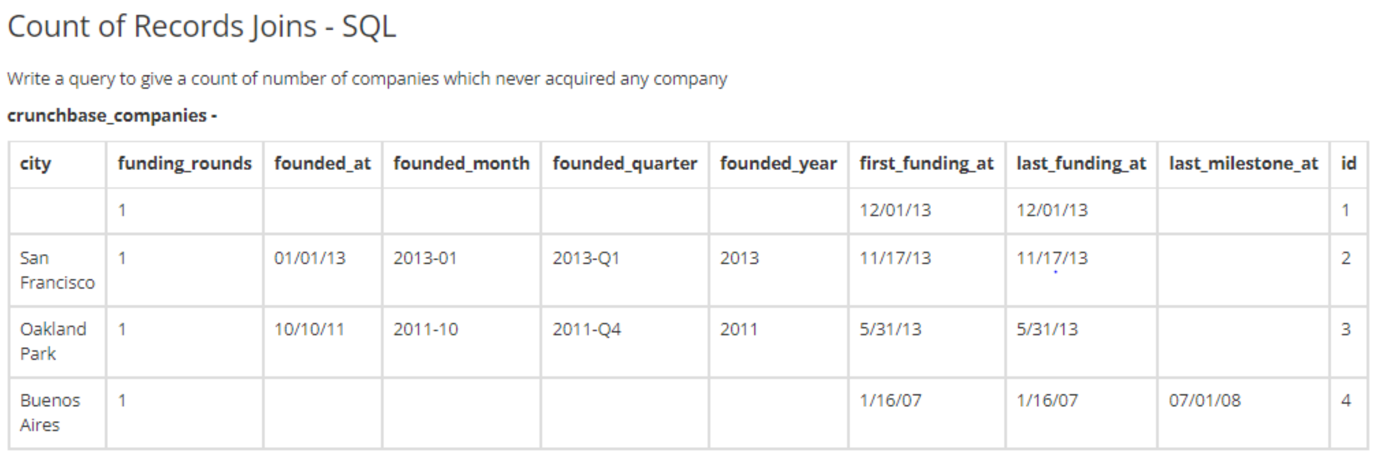
| acquired_month | acquired_quarter | acquired_year | price_amount | price_currency_code | id |
|---|---|---|---|---|---|
| 2013-10 | 2013-Q4 | 2013 | | USD | 1 |
| 2006-01 | 2006-Q1 | 2006 | | USD | 2 |
| 2014-02 | 2014-Q1 | 2014 | | USD | 3 |
| 2014-01 | 2014-Q1 | 06/07/05 | | USD | 4 |

solution:

```
select count(distinct a.permalink) as A1, count(distinct b.company_permalink) as B1

from crunchbase_companies as a

left join crunchbase_acquisitions as b

on a.permalink = b.company_permalink
```

## 4.Count of Records Joins - SQL

Write a query to give a count of number of companies which never acquired any company

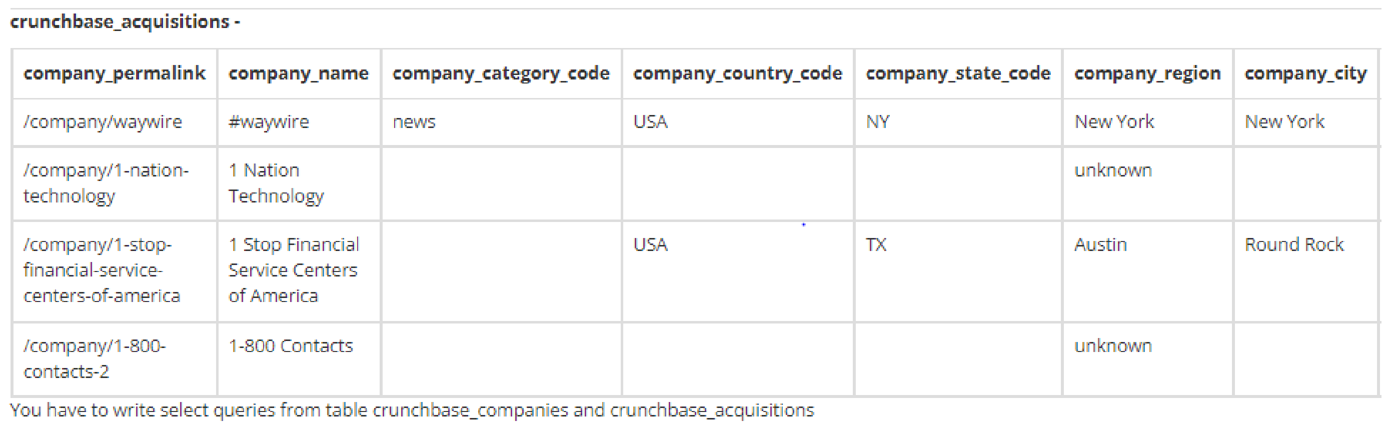


Count of Records Joins - SQL

Write a query to give a count of number of companies which never acquired any company

**crunchbase_companies -**

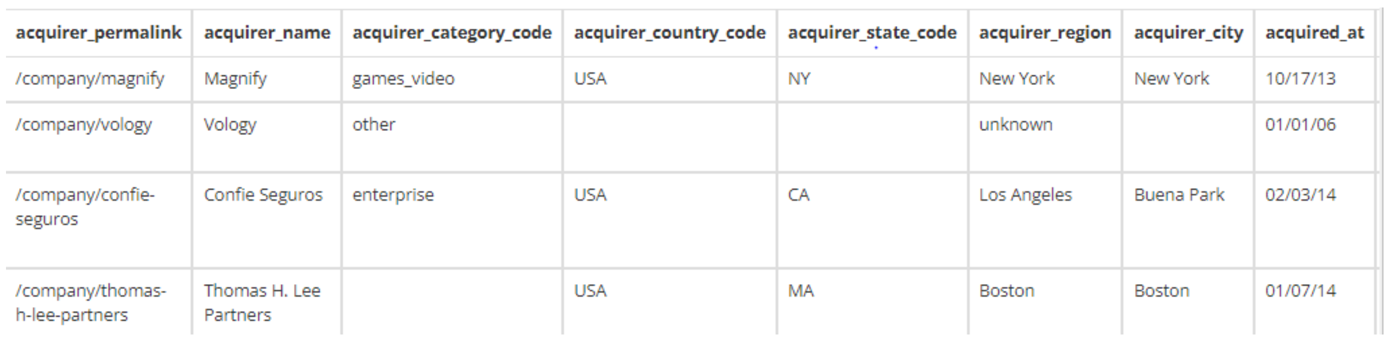
| city | funding_rounds | founded_at | founded_month | founded_quarter | founded_year | first_funding_at | last_funding_at | last_milestone_at | id |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | | | 12/01/13 | 12/01/13 | | 1 |
| San Francisco | 1 | 01/01/13 | 2013-01 | 2013-Q1 | 2013 | 11/17/13 | 11/17/13 | | 2 |
| Oakland Park | 1 | 10/10/11 | 2011-10 | 2011-Q4 | 2011 | 5/31/13 | 5/31/13 | | 3 |
| Buenos Aires | 1 | | | | | 1/16/07 | 1/16/07 | 07/01/08 | 4 |



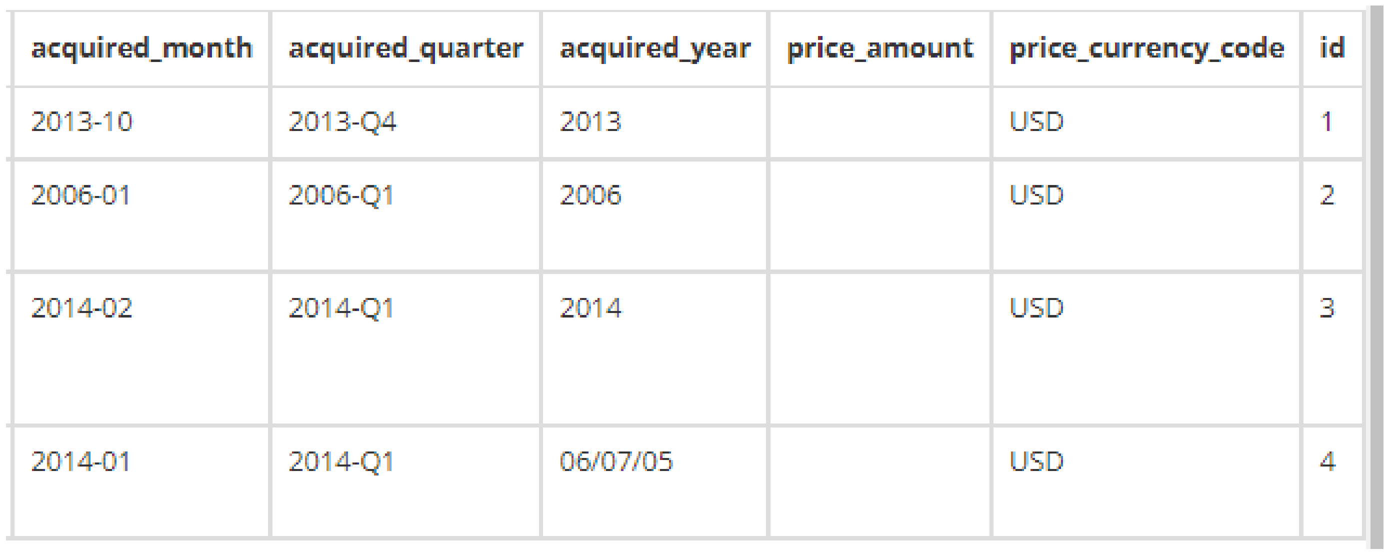
crunchbase_acquisitions -

| company_permalink | company_name | company_category_code | company_country_code | company_state_code | company_region | company_city |
|---|---|---|---|---|---|---|
| /company/waywire | #waywire | news | USA | NY | New York | New York |
| /company/1-nation-technology | 1 Nation Technology | | | | unknown | |
| /company/1-stop-financial-service-centers-of-america | 1 Stop Financial Service Centers of America | | USA | TX | Austin | Round Rock |
| /company/1-800-contacts-2 | 1-800 Contacts | | | | unknown | |

You have to write select queries from table crunchbase_companies and crunchbase_acquisitions



| acquirer_permalink | acquirer_name | acquirer_category_code | acquirer_country_code | acquirer_state_code | acquirer_region | acquirer_city | acquired_at |
|---|---|---|---|---|---|---|---|
| /company/magnify | Magnify | games_video | USA | NY | New York | New York | 10/17/13 |
| /company/vology | Vology | other | | | unknown | | 01/01/06 |
| /company/confie-seguros | Confie Seguros | enterprise | USA | CA | Los Angeles | Buena Park | 02/03/14 |
| /company/thomas-h-lee-partners | Thomas H. Lee Partners | | USA | MA | Boston | Boston | 01/07/14 |



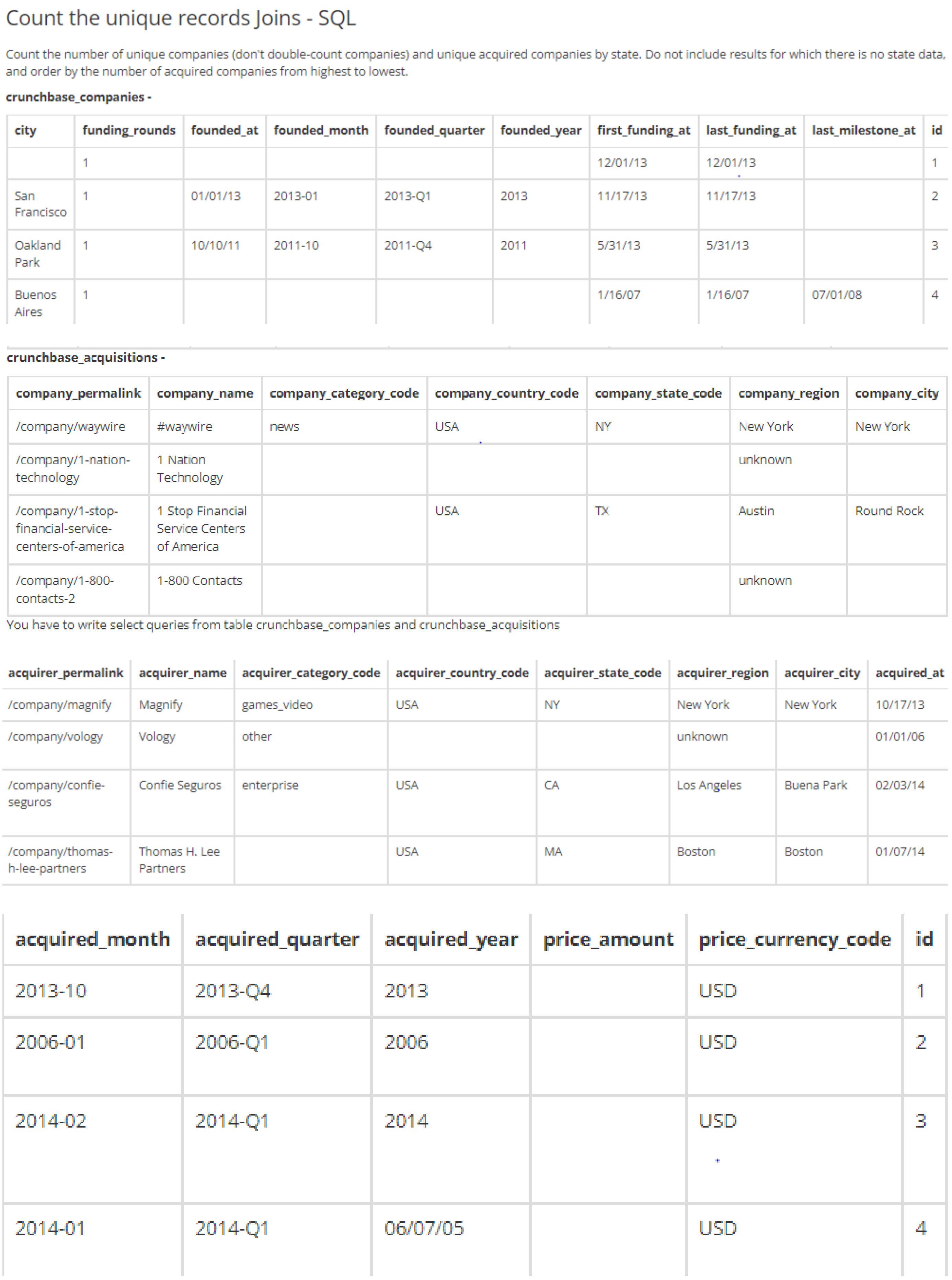
| acquired_month | acquired_quarter | acquired_year | price_amount | price_currency_code | id |
|---|---|---|---|---|---|
| 2013-10 | 2013-Q4 | 2013 | | USD | 1 |
| 2006-01 | 2006-Q1 | 2006 | | USD | 2 |
| 2014-02 | 2014-Q1 | 2014 | | USD | 3 |
| 2014-01 | 2014-Q1 | 06/07/05 | | USD | 4 |

solution:

```
select count(distinct a.permalink) as a1

from crunchbase_companies as a

left join crunchbase_acquisitions as b

on a.permalink = b.company_permalink

where b.company_name is null;
```

**5. Count the unique records Joins - SQL --> (Not Unterstood)**

Count the number of unique companies (don't double-count companies) and unique acquired companies by state. Do not include results for which there is no state data, and order by the number of acquired companies from highest to lowest.



## Count the unique records Joins - SQL

Count the number of unique companies (don't double-count companies) and unique acquired companies by state. Do not include results for which there is no state data, and order by the number of acquired companies from highest to lowest.

**crunchbase_companies -**

| city | funding_rounds | founded_at | founded_month | founded_quarter | founded_year | first_funding_at | last_funding_at | last_milestone_at | id |
|---|---|---|---|---|---|---|---|---|---|
|  | 1 |  |  |  |  | 12/01/13 | 12/01/13 |  | 1 |
| San Francisco | 1 | 01/01/13 | 2013-01 | 2013-Q1 | 2013 | 11/17/13 | 11/17/13 |  | 2 |
| Oakland Park | 1 | 10/10/11 | 2011-10 | 2011-Q4 | 2011 | 5/31/13 | 5/31/13 |  | 3 |
| Buenos Aires | 1 |  |  |  |  | 1/16/07 | 1/16/07 | 07/01/08 | 4 |

**crunchbase_acquisitions -**

| company_permalink | company_name | company_category_code | company_country_code | company_state_code | company_region | company_city |
|---|---|---|---|---|---|---|
| /company/waywire | #waywire | news | USA | NY | New York | New York |
| /company/1-nation-technology | 1 Nation Technology |  |  |  | unknown |  |
| /company/1-stop-financial-service-centers-of-america | 1 Stop Financial Service Centers of America |  | USA | TX | Austin | Round Rock |
| /company/1-800-contacts-2 | 1-800 Contacts |  |  |  | unknown |  |

You have to write select queries from table crunchbase_companies and crunchbase_acquisitions

| acquirer_permalink | acquirer_name | acquirer_category_code | acquirer_country_code | acquirer_state_code | acquirer_region | acquirer_city | acquired_at |
|---|---|---|---|---|---|---|---|
| /company/magnify | Magnify | games_video | USA | NY | New York | New York | 10/17/13 |
| /company/vology | Vology | other |  |  | unknown |  | 01/01/06 |
| /company/confie-seguros | Confie Seguros | enterprise | USA | CA | Los Angeles | Buena Park | 02/03/14 |
| /company/thomas-h-lee-partners | Thomas H. Lee Partners |  | USA | MA | Boston | Boston | 01/07/14 |

| acquired_month | acquired_quarter | acquired_year | price_amount | price_currency_code | id |
|---|---|---|---|---|---|
| 2013-10 | 2013-Q4 | 2013 |  | USD | 1 |
| 2006-01 | 2006-Q1 | 2006 |  | USD | 2 |
| 2014-02 | 2014-Q1 | 2014 |  | USD | 3 |
| 2014-01 | 2014-Q1 | 06/07/05 |  | USD | 4 |

solution:

SELECT companies.state_code,

```
       COUNT(DISTINCT companies.permalink) AS unique_companies,

       COUNT(DISTINCT acquisitions.company_permalink) AS unique_compani
    es_acquired
```

FROM crunchbase_companies companies

LEFT JOIN crunchbase_acquisitions acquisitions

ON companies.permalink = acquisitions.company_permalink

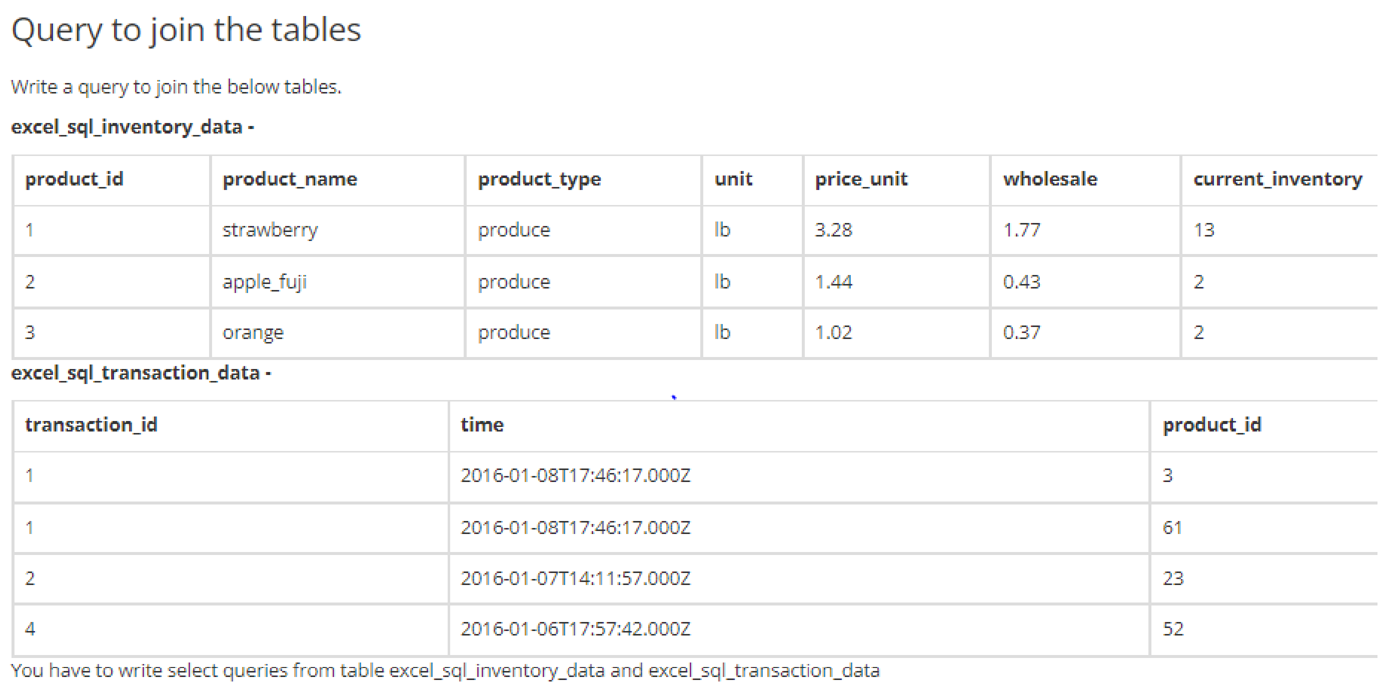
WHERE companies.state_code IS NOT NULL

GROUP BY 1

ORDER BY 3 DESC

## 6. Query to join the tables

Write a query to join the below tables.



Query to join the tables

Write a query to join the below tables.

**excel_sql_inventory_data -**

| product_id | product_name | product_type | unit | price_unit | wholesale | current_inventory |
|---|---|---|---|---|---|---|
| 1 | strawberry | produce | lb | 3.28 | 1.77 | 13 |
| 2 | apple_fuji | produce | lb | 1.44 | 0.43 | 2 |
| 3 | orange | produce | lb | 1.02 | 0.37 | 2 |

**excel_sql_transaction_data -**

| transaction_id | time | product_id |
|---|---|---|
| 1 | 2016-01-08T17:46:17.000Z | 3 |
| 1 | 2016-01-08T17:46:17.000Z | 61 |
| 2 | 2016-01-07T14:11:57.000Z | 23 |
| 4 | 2016-01-06T17:57:42.000Z | 52 |

You have to write select queries from table excel_sql_inventory_data and excel_sql_transaction_data

solution:

select a.*,b.time

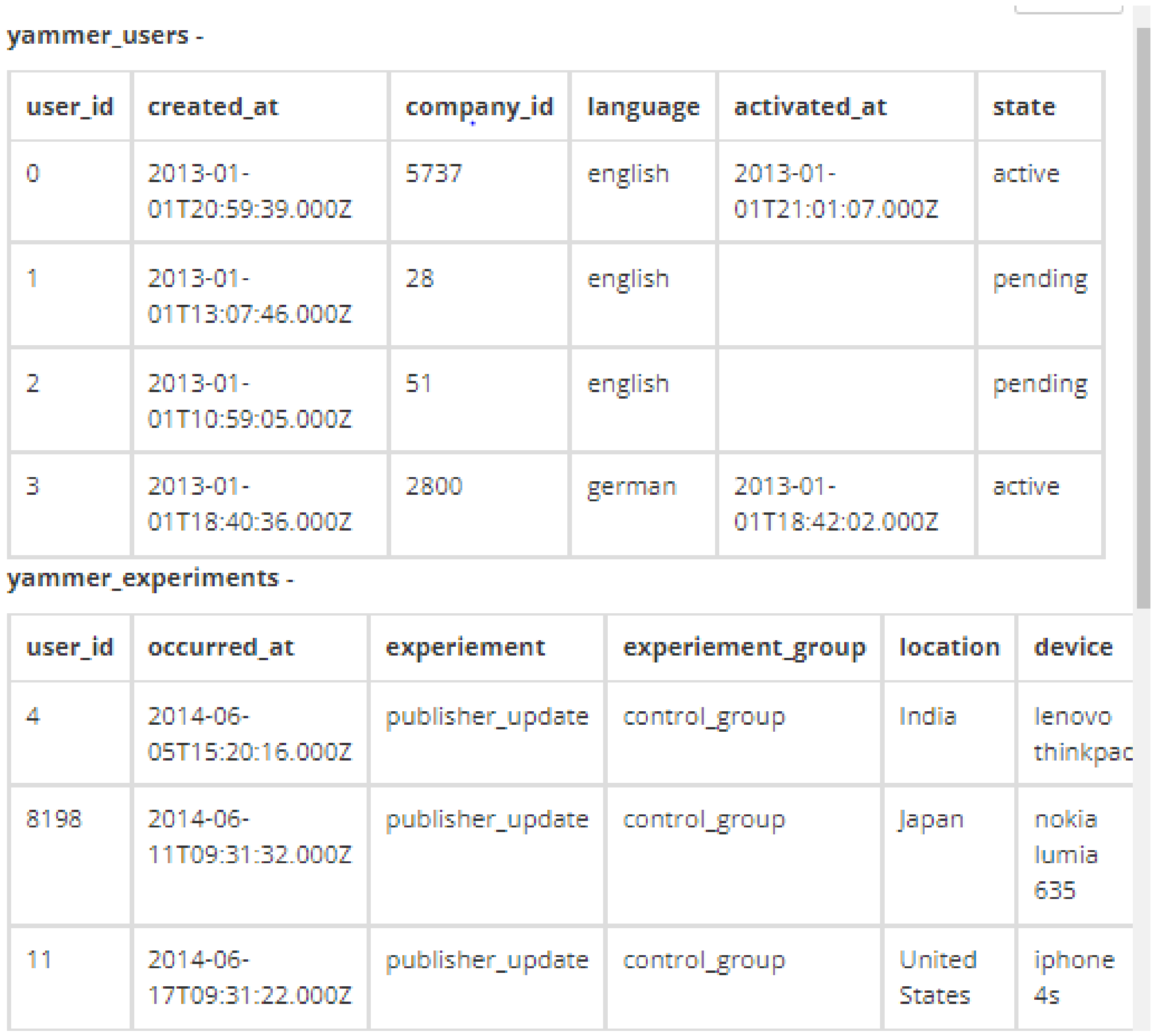
from excel_sql_inventory_data as a

left join excel_sql_transaction_data as b

on a.product_id = b.product_id

## 7. Number of users Joins - SQL

Find the number of users per language type/

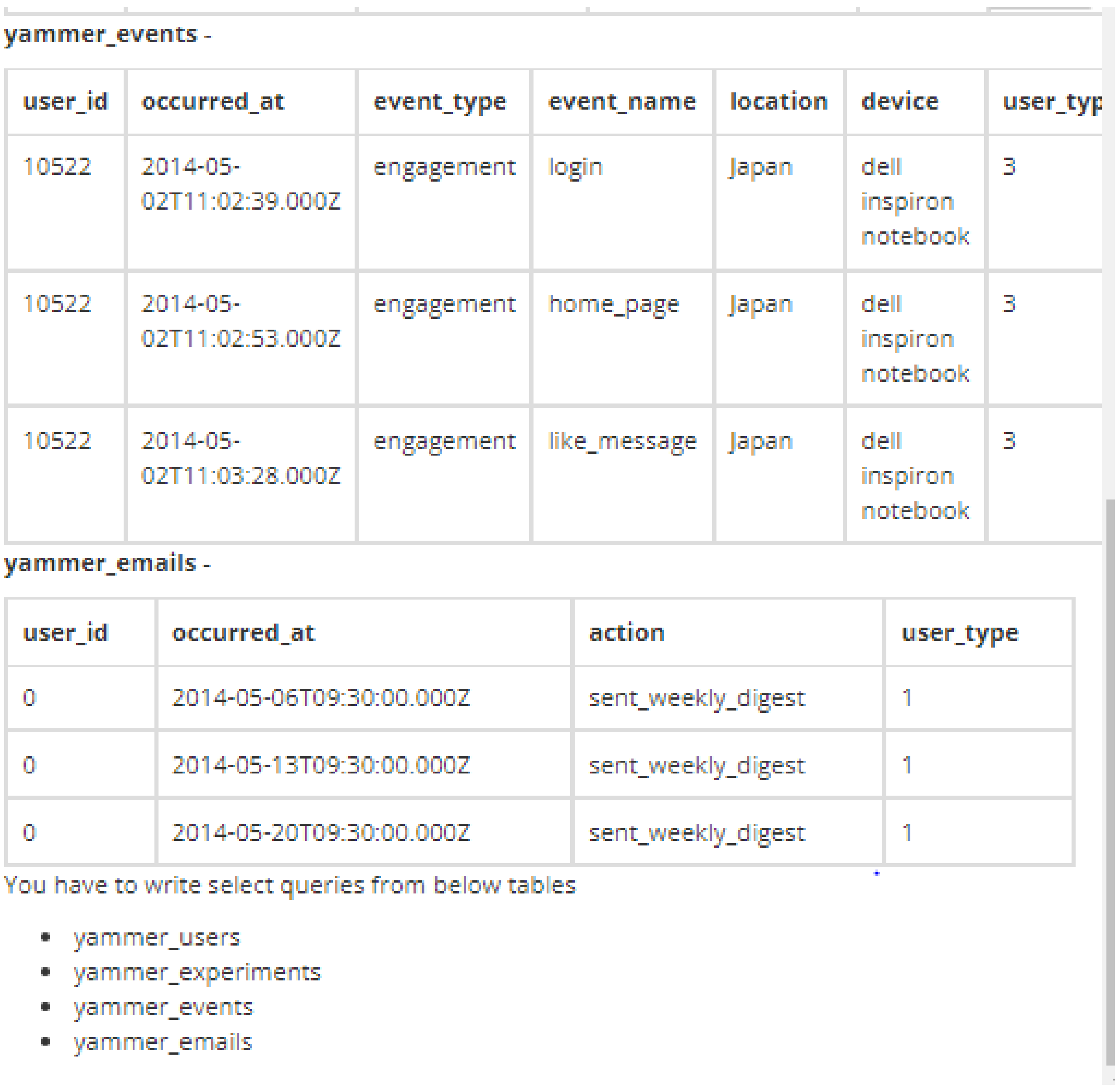


yammer_users -

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | 2013-01-01T20:59:39.000Z | 5737 | english | 2013-01-01T21:01:07.000Z | active |
| 1 | 2013-01-01T13:07:46.000Z | 28 | english |  | pending |
| 2 | 2013-01-01T10:59:05.000Z | 51 | english |  | pending |
| 3 | 2013-01-01T18:40:36.000Z | 2800 | german | 2013-01-01T18:42:02.000Z | active |

yammer_experiments -

| user_id | occurred_at | experiement | experiement_group | location | device |
|---|---|---|---|---|---|
| 4 | 2014-06-05T15:20:16.000Z | publisher_update | control_group | India | lenovo thinkpac |
| 8198 | 2014-06-11T09:31:32.000Z | publisher_update | control_group | Japan | nokia lumia 635 |
| 11 | 2014-06-17T09:31:22.000Z | publisher_update | control_group | United States | iphone 4s |


yammer_events -

| user_id | occurred_at | event_type | event_name | location | device | user_typ |
|---|---|---|---|---|---|---|
| 10522 | 2014-05-02T11:02:39.000Z | engagement | login | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:02:53.000Z | engagement | home_page | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:03:28.000Z | engagement | like_message | Japan | dell inspiron notebook | 3 |

yammer_emails -

| user_id | occurred_at | action | user_type |
|---|---|---|---|
| 0 | 2014-05-06T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-13T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-20T09:30:00.000Z | sent_weekly_digest | 1 |

You have to write select queries from below tables

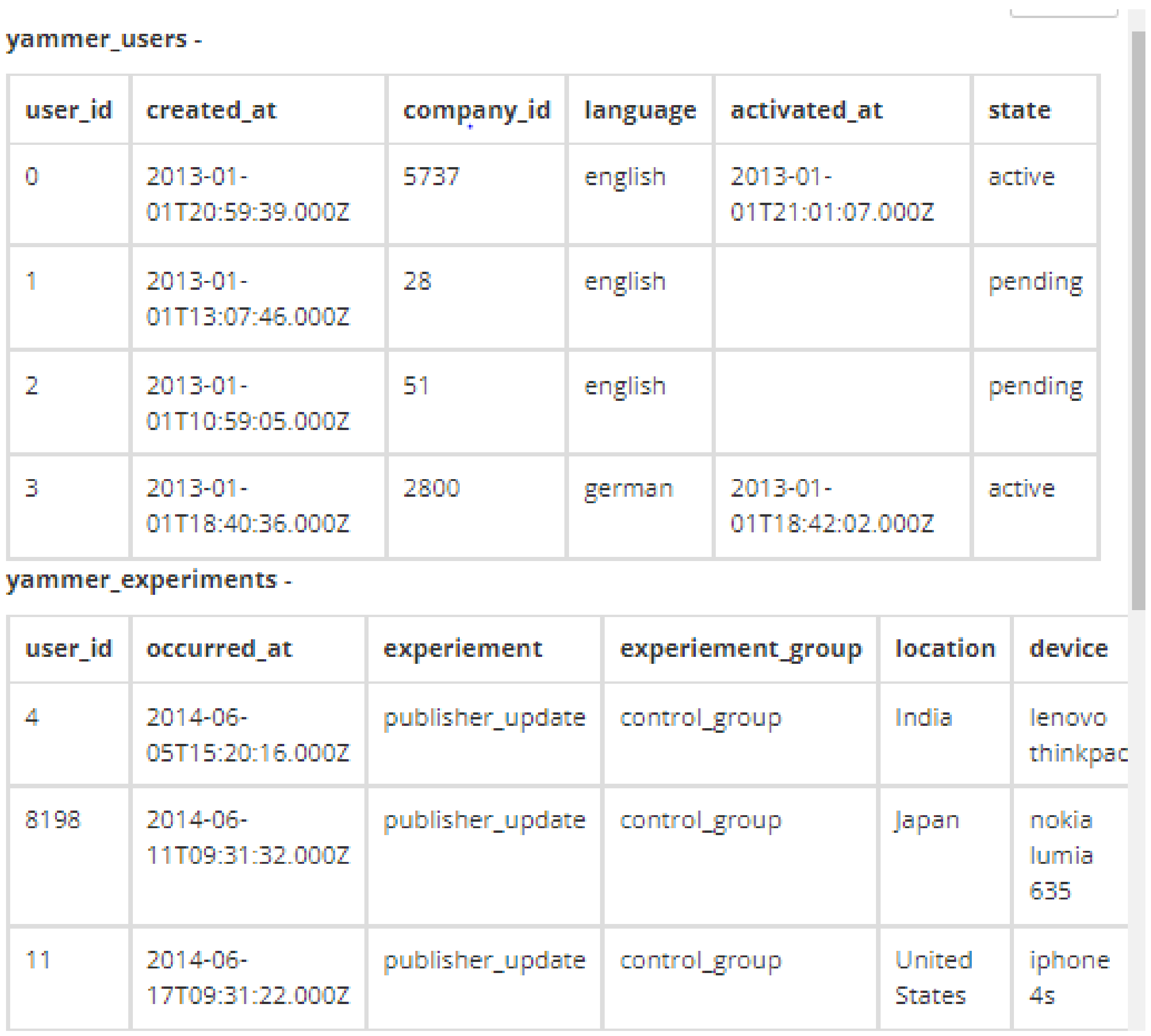
- yammer_users
- yammer_experiments
- yammer_events
- yammer_emails

solution:

```
select language,count(user_id) as lan

from yammer_users

group by 1;
```

**8.Write a query to find how many users are part of experiments SQL**

Write a query to find how many users are part of experiments.

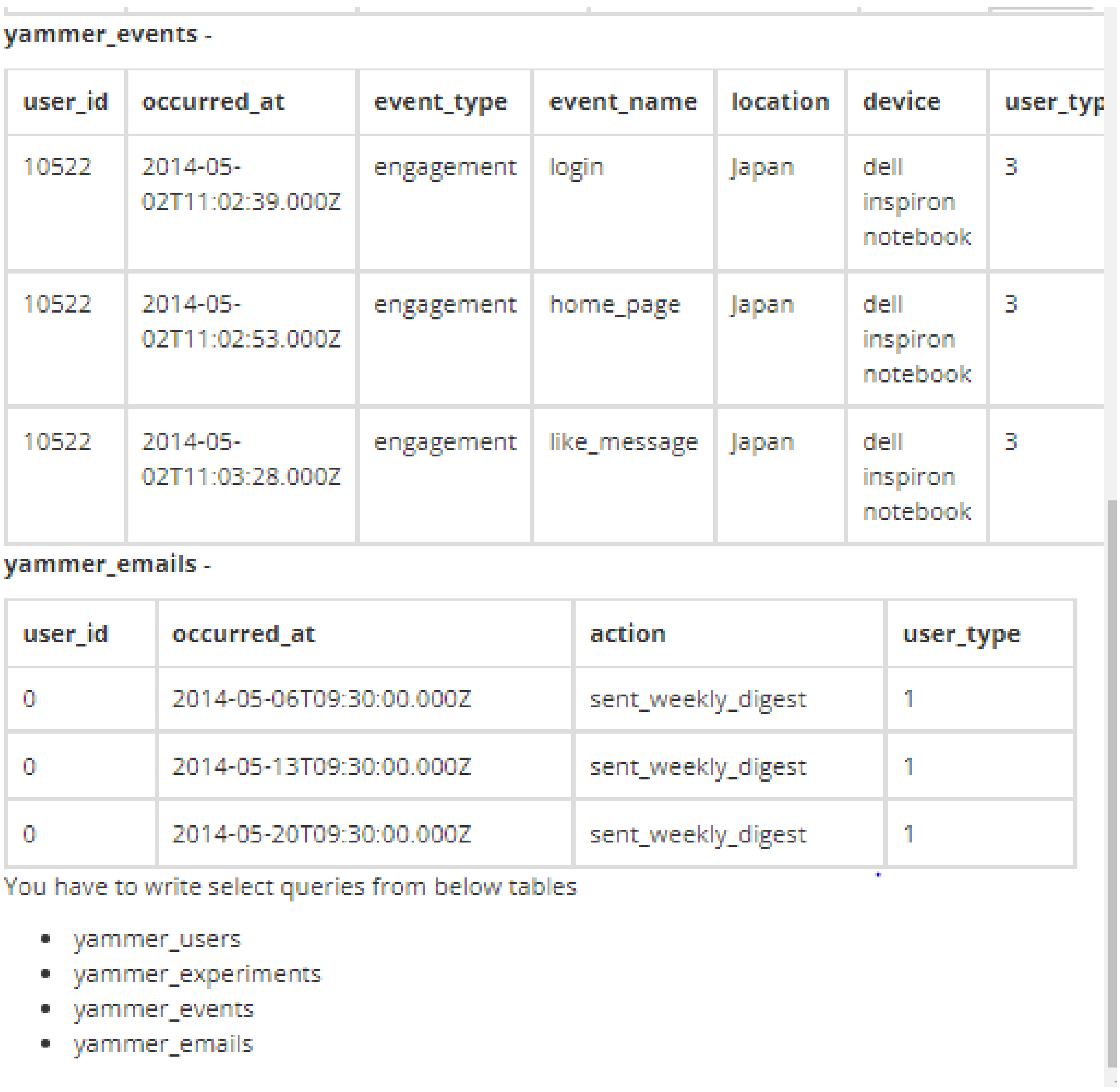


yammer_users -

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | 2013-01-01T20:59:39.000Z | 5737 | english | 2013-01-01T21:01:07.000Z | active |
| 1 | 2013-01-01T13:07:46.000Z | 28 | english | | pending |
| 2 | 2013-01-01T10:59:05.000Z | 51 | english | | pending |
| 3 | 2013-01-01T18:40:36.000Z | 2800 | german | 2013-01-01T18:42:02.000Z | active |

yammer_experiments -

| user_id | occurred_at | experiement | experiement_group | location | device |
|---|---|---|---|---|---|
| 4 | 2014-06-05T15:20:16.000Z | publisher_update | control_group | India | lenovo thinkpac |
| 8198 | 2014-06-11T09:31:32.000Z | publisher_update | control_group | Japan | nokia lumia 635 |
| 11 | 2014-06-17T09:31:22.000Z | publisher_update | control_group | United States | iphone 4s |



yammer_events -

| user_id | occurred_at | event_type | event_name | location | device | user_typ |
|---|---|---|---|---|---|---|
| 10522 | 2014-05-02T11:02:39.000Z | engagement | login | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:02:53.000Z | engagement | home_page | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:03:28.000Z | engagement | like_message | Japan | dell inspiron notebook | 3 |

yammer_emails -

| user_id | occurred_at | action | user_type |
|---|---|---|---|
| 0 | 2014-05-06T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-13T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-20T09:30:00.000Z | sent_weekly_digest | 1 |

You have to write select queries from below tables

- yammer_users
- yammer_experiments
- yammer_events
- yammer_emails

solution:

SELECT COUNT(DISTINCT a.user_id) AS total_users,COUNT(DISTINCT b.user_id) AS users_experiment

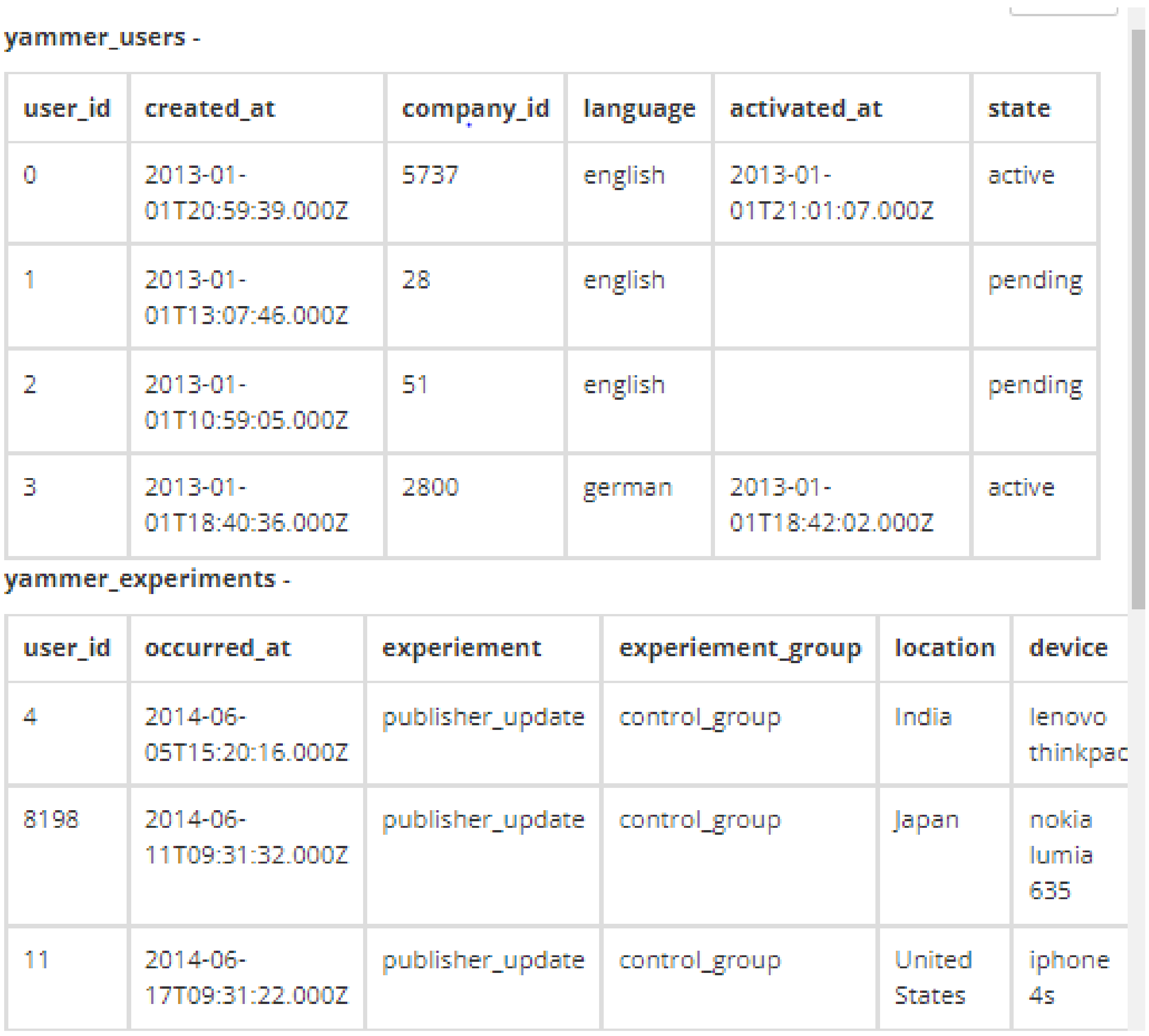
FROM yammer_users a

LEFT JOIN yammer_events b

ON a.user_id = b.user_id

**9. Find the number of users in experiment per language category SQL**

Find the number of users in experiment per language category

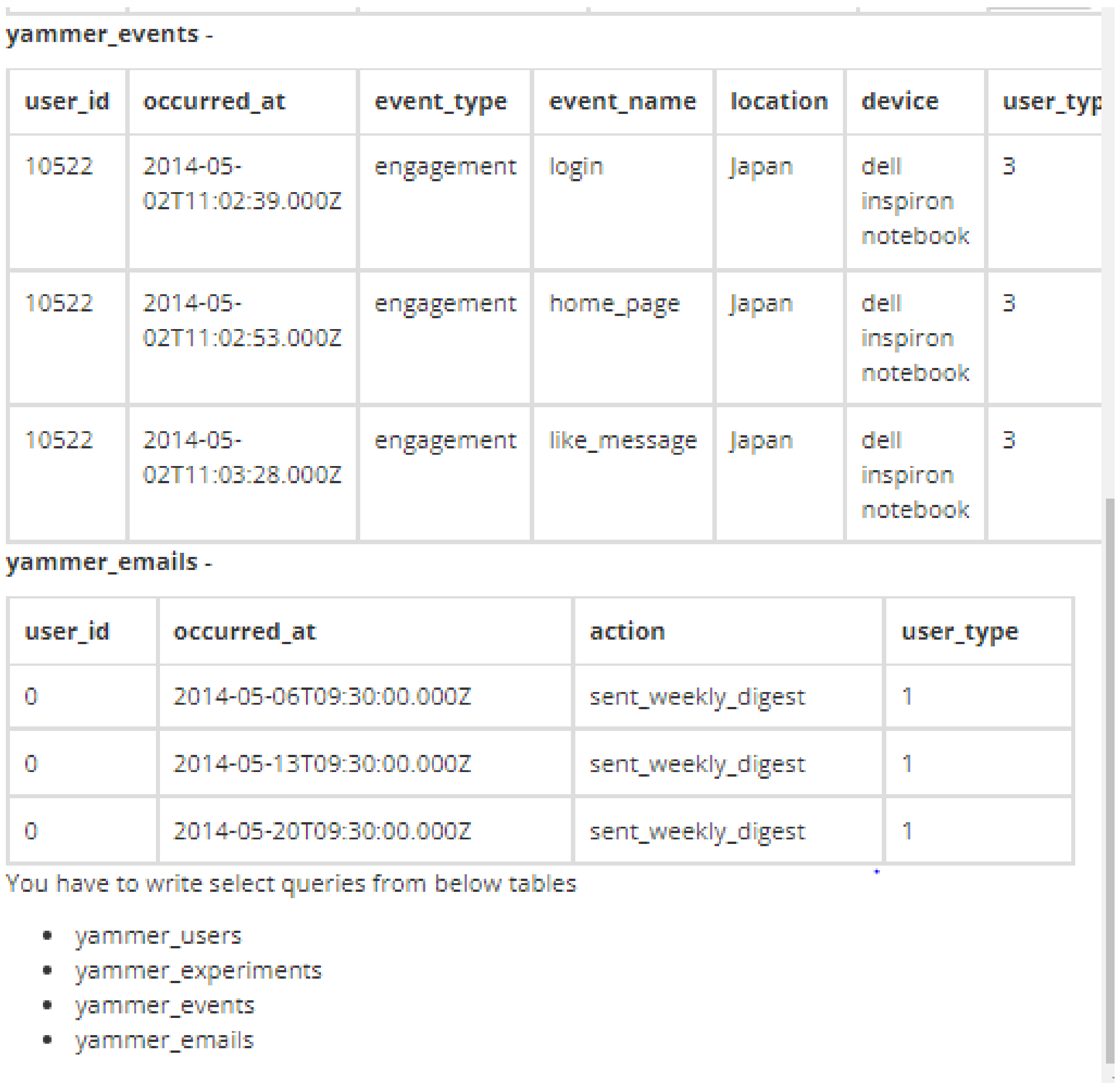


yammer_users -

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | 2013-01-01T20:59:39.000Z | 5737 | english | 2013-01-01T21:01:07.000Z | active |
| 1 | 2013-01-01T13:07:46.000Z | 28 | english | | pending |
| 2 | 2013-01-01T10:59:05.000Z | 51 | english | | pending |
| 3 | 2013-01-01T18:40:36.000Z | 2800 | german | 2013-01-01T18:42:02.000Z | active |

yammer_experiments -

| user_id | occurred_at | experiement | experiement_group | location | device |
|---|---|---|---|---|---|
| 4 | 2014-06-05T15:20:16.000Z | publisher_update | control_group | India | lenovo thinkpac |
| 8198 | 2014-06-11T09:31:32.000Z | publisher_update | control_group | Japan | nokia lumia 635 |
| 11 | 2014-06-17T09:31:22.000Z | publisher_update | control_group | United States | iphone 4s |



yammer_events -

| user_id | occurred_at | event_type | event_name | location | device | user_typ |
|---|---|---|---|---|---|---|
| 10522 | 2014-05-02T11:02:39.000Z | engagement | login | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:02:53.000Z | engagement | home_page | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:03:28.000Z | engagement | like_message | Japan | dell inspiron notebook | 3 |

yammer_emails -

| user_id | occurred_at | action | user_type |
|---|---|---|---|
| 0 | 2014-05-06T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-13T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-20T09:30:00.000Z | sent_weekly_digest | 1 |

You have to write select queries from below tables

- yammer_users
- yammer_experiments
- yammer_events
- yammer_emails

solution:

SELECT a.language,COUNT(DISTINCT b.user_id) AS a1

FROM yammer_users a

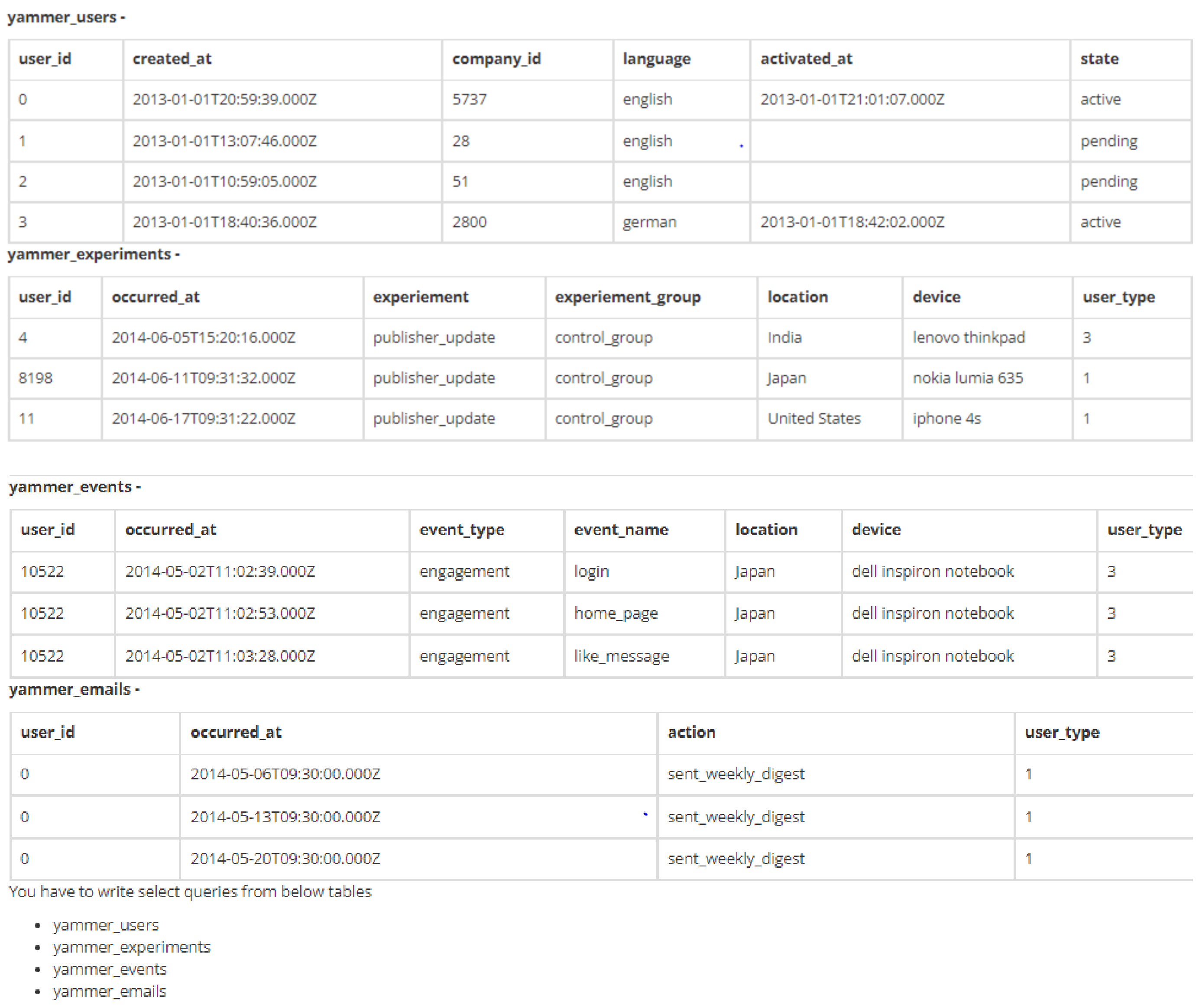
LEFT JOIN yammer_events b

ON a.user_id = b.user_id

GROUP BY 1;

**10. Write a query to find how many users have received at least one email --------->""AT LEAST PROBLEM STARTS FROM HERE""**

Write a query to find how many users have received at least one email



yammer_users -

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | 2013-01-01T20:59:39.000Z | 5737 | english | 2013-01-01T21:01:07.000Z | active |
| 1 | 2013-01-01T13:07:46.000Z | 28 | english |  | pending |
| 2 | 2013-01-01T10:59:05.000Z | 51 | english |  | pending |
| 3 | 2013-01-01T18:40:36.000Z | 2800 | german | 2013-01-01T18:42:02.000Z | active |

yammer_experiments -

| user_id | occurred_at | experiment | experiement_group | location | device | user_type |
|---|---|---|---|---|---|---|
| 4 | 2014-06-05T15:20:16.000Z | publisher_update | control_group | India | lenovo thinkpad | 3 |
| 8198 | 2014-06-11T09:31:32.000Z | publisher_update | control_group | Japan | nokia lumia 635 | 1 |
| 11 | 2014-06-17T09:31:22.000Z | publisher_update | control_group | United States | iphone 4s | 1 |

yammer_events -

| user_id | occurred_at | event_type | event_name | location | device | user_type |
|---|---|---|---|---|---|---|
| 10522 | 2014-05-02T11:02:39.000Z | engagement | login | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:02:53.000Z | engagement | home_page | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:03:28.000Z | engagement | like_message | Japan | dell inspiron notebook | 3 |

yammer_emails -

| user_id | occurred_at | action | user_type |
|---|---|---|---|
| 0 | 2014-05-06T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-13T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-20T09:30:00.000Z | sent_weekly_digest | 1 |

You have to write select queries from below tables

- yammer_users
- yammer_experiments
- yammer_events
- yammer_emails

solutino:

SELECT COUNT(DISTINCT a.user_id) AS a1, COUNT(DISTINCT b.user_id) AS b1

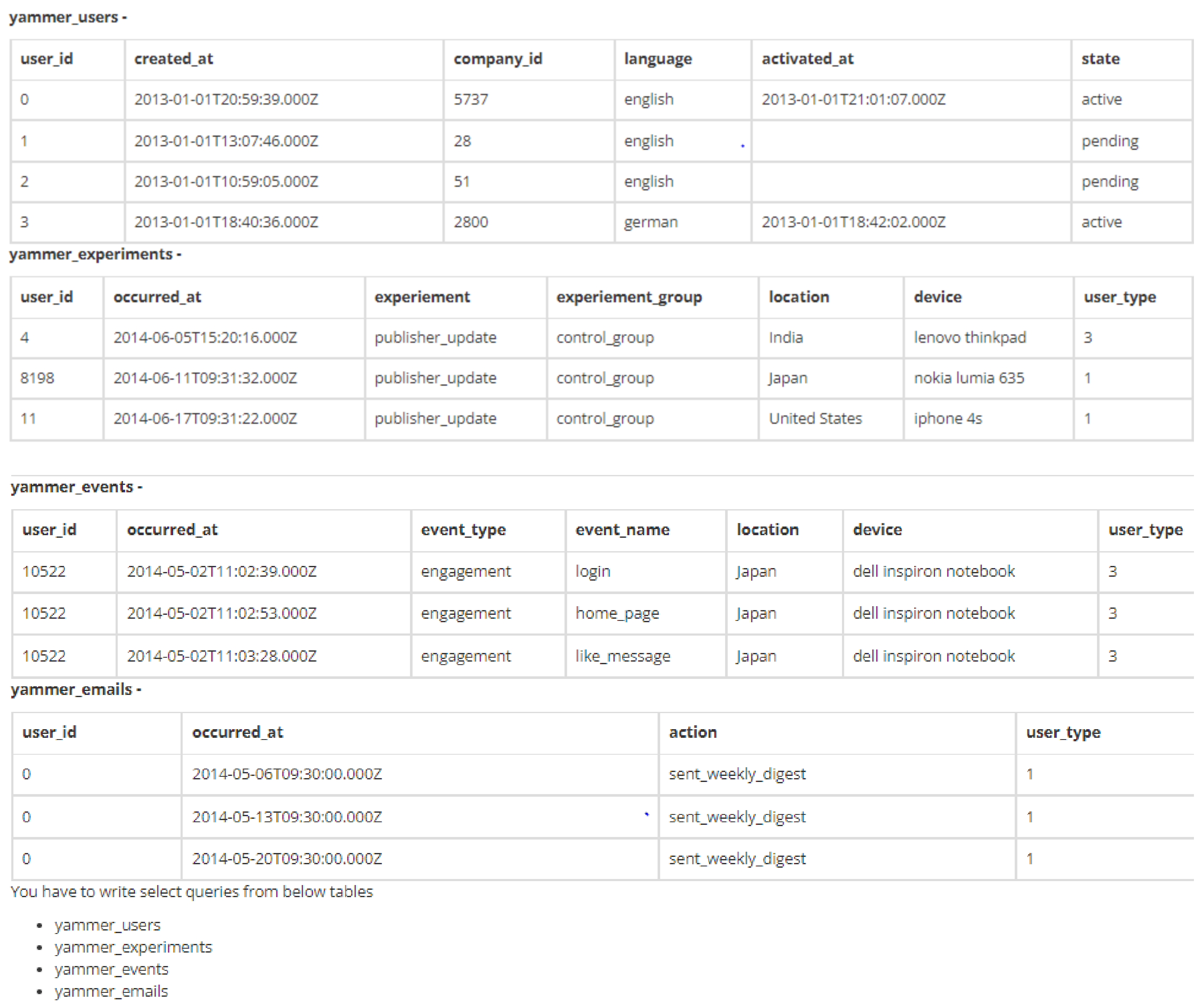
FROM yammer_users a

LEFT JOIN yammer_emails b

ON a.user_id = b.user_id

**11. Write a query to find how many users per company id have received at least one email?**

Write a query to find how many users per company id have received at least one email?



**yammer_users -**

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | 2013-01-01T20:59:39.000Z | 5737 | english | 2013-01-01T21:01:07.000Z | active |
| 1 | 2013-01-01T13:07:46.000Z | 28 | english | | pending |
| 2 | 2013-01-01T10:59:05.000Z | 51 | english | | pending |
| 3 | 2013-01-01T18:40:36.000Z | 2800 | german | 2013-01-01T18:42:02.000Z | active |

**yammer_experiments -**

| user_id | occurred_at | experiement | experiement_group | location | device | user_type |
|---|---|---|---|---|---|---|
| 4 | 2014-06-05T15:20:16.000Z | publisher_update | control_group | India | lenovo thinkpad | 3 |
| 8198 | 2014-06-11T09:31:32.000Z | publisher_update | control_group | Japan | nokia lumia 635 | 1 |
| 11 | 2014-06-17T09:31:22.000Z | publisher_update | control_group | United States | iphone 4s | 1 |

**yammer_events -**

| user_id | occurred_at | event_type | event_name | location | device | user_type |
|---|---|---|---|---|---|---|
| 10522 | 2014-05-02T11:02:39.000Z | engagement | login | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:02:53.000Z | engagement | home_page | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:03:28.000Z | engagement | like_message | Japan | dell inspiron notebook | 3 |

**yammer_emails -**

| user_id | occurred_at | action | user_type |
|---|---|---|---|
| 0 | 2014-05-06T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-13T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-20T09:30:00.000Z | sent_weekly_digest | 1 |

You have to write select queries from below tables

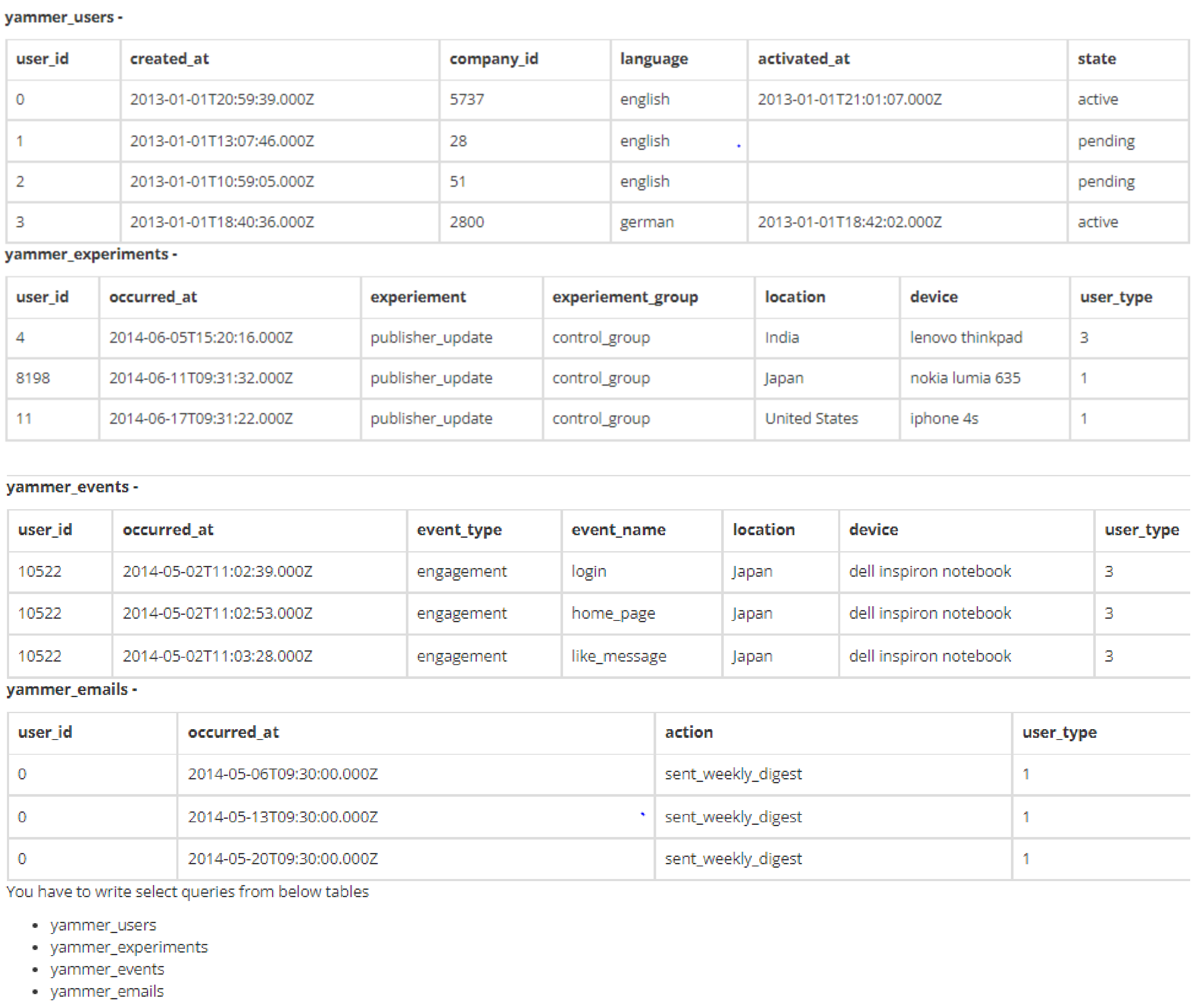
- yammer_users
- yammer_experiments
- yammer_events
- yammer_emails

Solution:

```
SELECT company_id AS a1, COUNT(DISTINCT b.user_id) AS b1

FROM yammer_users a

LEFT JOIN yammer_emails b

ON a.user_id = b.user_id

group by a1;
```

## 12. Write a query to find how many users have received at least one event SQL

Write a query to find how many users have received at least one event



yammer_users -

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | 2013-01-01T20:59:39.000Z | 5737 | english | 2013-01-01T21:01:07.000Z | active |
| 1 | 2013-01-01T13:07:46.000Z | 28 | english |  | pending |
| 2 | 2013-01-01T10:59:05.000Z | 51 | english |  | pending |
| 3 | 2013-01-01T18:40:36.000Z | 2800 | german | 2013-01-01T18:42:02.000Z | active |

yammer_experiments -

| user_id | occurred_at | experiment | experiement_group | location | device | user_type |
|---|---|---|---|---|---|---|
| 4 | 2014-06-05T15:20:16.000Z | publisher_update | control_group | India | lenovo thinkpad | 3 |
| 8198 | 2014-06-11T09:31:32.000Z | publisher_update | control_group | Japan | nokia lumia 635 | 1 |
| 11 | 2014-06-17T09:31:22.000Z | publisher_update | control_group | United States | iphone 4s | 1 |

yammer_events -

| user_id | occurred_at | event_type | event_name | location | device | user_type |
|---|---|---|---|---|---|---|
| 10522 | 2014-05-02T11:02:39.000Z | engagement | login | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:02:53.000Z | engagement | home_page | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:03:28.000Z | engagement | like_message | Japan | dell inspiron notebook | 3 |

yammer_emails -

| user_id | occurred_at | action | user_type |
|---|---|---|---|
| 0 | 2014-05-06T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-13T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-20T09:30:00.000Z | sent_weekly_digest | 1 |

You have to write select queries from below tables

- yammer_users
- yammer_experiments
- yammer_events
- yammer_emails

Solution:

```
SELECT count(distinct a.user_id) AS a1, COUNT(DISTINCT b.user_id) AS b1

FROM yammer_users a

LEFT JOIN yammer_events as b

ON a.user_id = b.user_id;
```

### 13. Write a query to find how many distinct users per state have at least one event-SQL

Write a query to find how many distinct users per state have at least one event?

**yammer_users -**

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | 2013-01-01T20:59:39.000Z | 5737 | english | 2013-01-01T21:01:07.000Z | active |
| 1 | 2013-01-01T13:07:46.000Z | 28 | english | | pending |
| 2 | 2013-01-01T10:59:05.000Z | 51 | english | | pending |
| 3 | 2013-01-01T18:40:36.000Z | 2800 | german | 2013-01-01T18:42:02.000Z | active |

**yammer_experiments -**

| user_id | occurred_at | experiement | experiement_group | location | device | user_type |
|---|---|---|---|---|---|---|
| 4 | 2014-06-05T15:20:16.000Z | publisher_update | control_group | India | lenovo thinkpad | 3 |
| 8198 | 2014-06-11T09:31:32.000Z | publisher_update | control_group | Japan | nokia lumia 635 | 1 |
| 11 | 2014-06-17T09:31:22.000Z | publisher_update | control_group | United States | iphone 4s | 1 |

**yammer_events -**

| user_id | occurred_at | event_type | event_name | location | device | user_type |
|---|---|---|---|---|---|---|
| 10522 | 2014-05-02T11:02:39.000Z | engagement | login | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:02:53.000Z | engagement | home_page | Japan | dell inspiron notebook | 3 |
| 10522 | 2014-05-02T11:03:28.000Z | engagement | like_message | Japan | dell inspiron notebook | 3 |

**yammer_emails -**

| user_id | occurred_at | action | user_type |
|---|---|---|---|
| 0 | 2014-05-06T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-13T09:30:00.000Z | sent_weekly_digest | 1 |
| 0 | 2014-05-20T09:30:00.000Z | sent_weekly_digest | 1 |

You have to write select queries from below tables

- yammer_users
- yammer_experiments
- yammer_events
- yammer_emails

Solution:

SELECT a.state, COUNT(DISTINCT b.user_id) AS b1

FROM yammer_users a

LEFT JOIN yammer_events as b

ON a.user_id = b.user_id

group by 1;

In [ ]:

```

```

In [ ]:

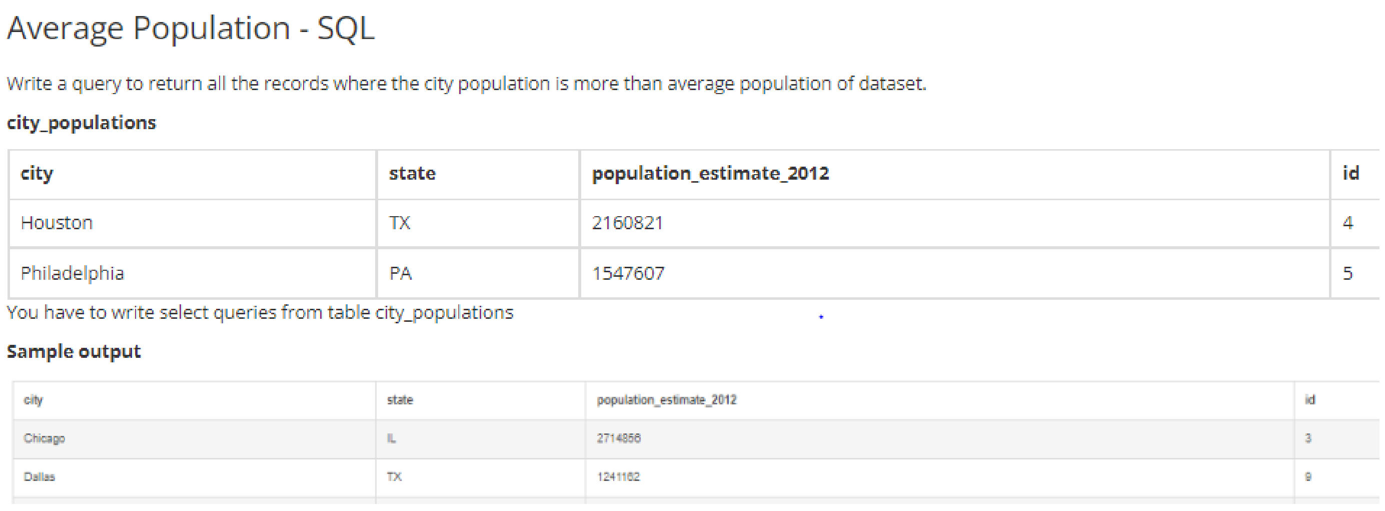
```

```

# SQL Logic Building - 5

## SUB-QUERIES

### 1. Average Population - SQL (Avg)

Write a query to return all the records where the city population is more than average population of dataset.



Average Population - SQL

Write a query to return all the records where the city population is more than average population of dataset.

**city_populations**

| city | state | population_estimate_2012 | id |
|---|---|---|---|
| Houston | TX | 2160821 | 4 |
| Philadelphia | PA | 1547607 | 5 |

You have to write select queries from table city_populations

**Sample output**

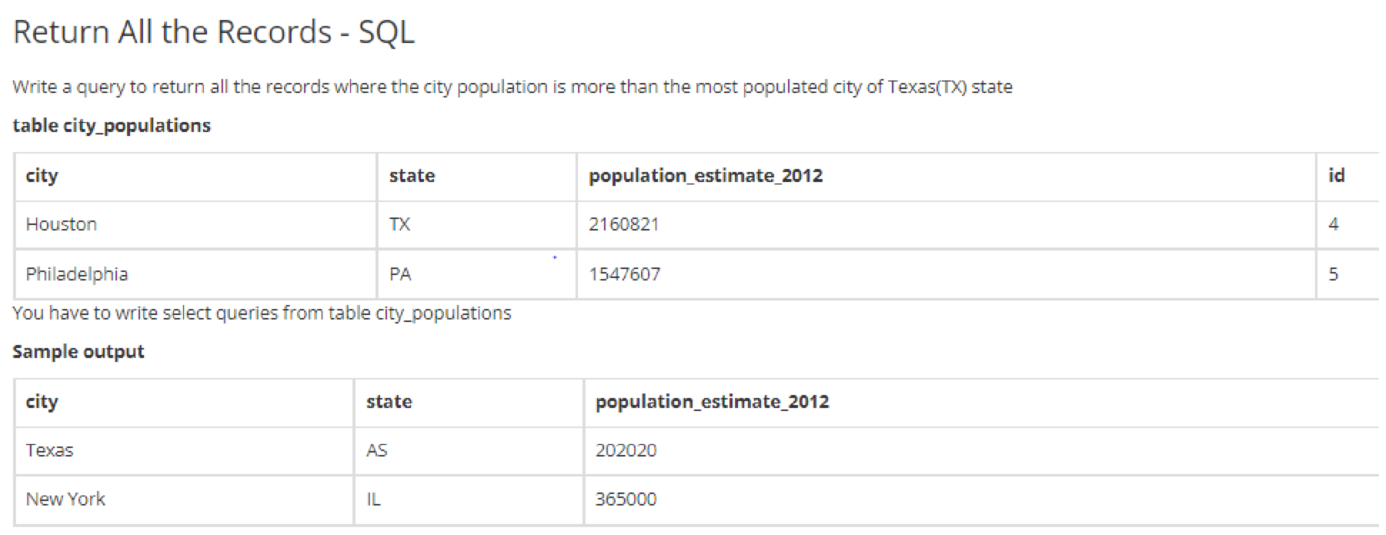
| city | state | population_estimate_2012 | id |
|---|---|---|---|
| Chicago | IL | 2714856 | 3 |
| Dallas | TX | 1241162 | 9 |

solution:

```
select *

from city_populations

where population_estimate_2012 >

    (select avg(population_estimate_2012) from city_populations)
```

### 2. Return All the Records - SQL (max)

Write a query to return all the records where the city population is more than the most populated city of Texas(TX) state



Return All the Records - SQL

Write a query to return all the records where the city population is more than the most populated city of Texas(TX) state

**table city_populations**

| city | state | population_estimate_2012 | id |
|---|---|---|---|
| Houston | TX | 2160821 | 4 |
| Philadelphia | PA | 1547607 | 5 |

You have to write select queries from table city_populations

**Sample output**

| city | state | population_estimate_2012 |
|---|---|---|
| Texas | AS | 202020 |
| New York | IL | 365000 |

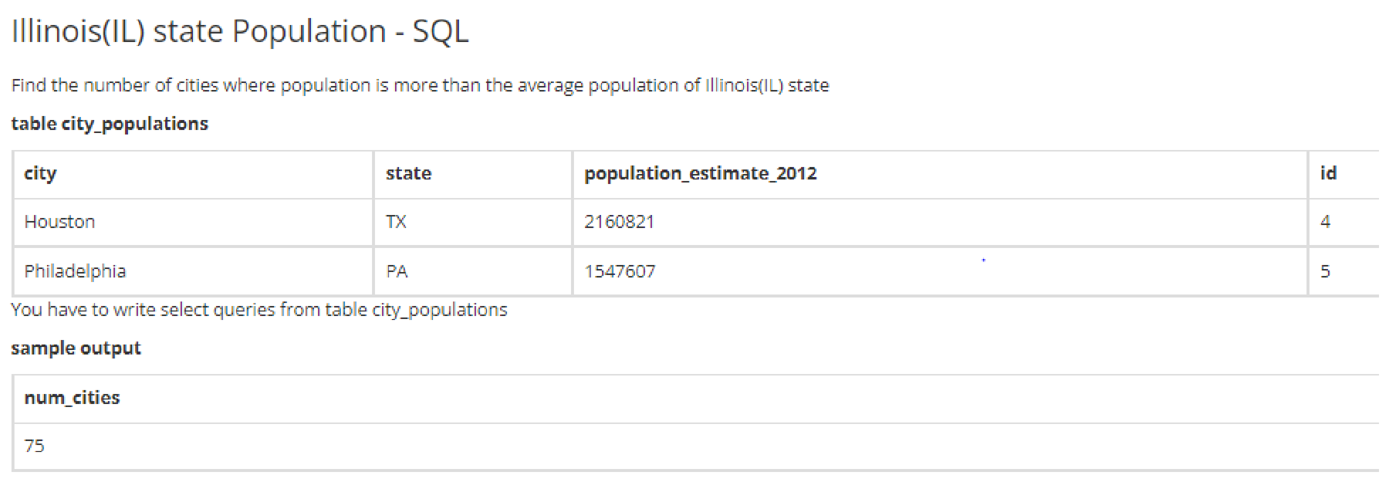
Solution:

select * from city_populations where population_estimate_2012 >

```
     (select max(population_estimate_2012) from city_populations wer
e state = "TX")
```

## 3. Illinois(IL) state Population - SQL

Find the number of cities where population is more than the average population of Illinois(IL) state



Illinois(IL) state Population - SQL

Find the number of cities where population is more than the average population of Illinois(IL) state

**table city_populations**

| city | state | population_estimate_2012 | id |
|---|---|---|---|
| Houston | TX | 2160821 | 4 |
| Philadelphia | PA | 1547607 | 5 |

You have to write select queries from table city_populations

**sample output**

| num_cities |
|---|
| 75 |

Solution:

select count(city)

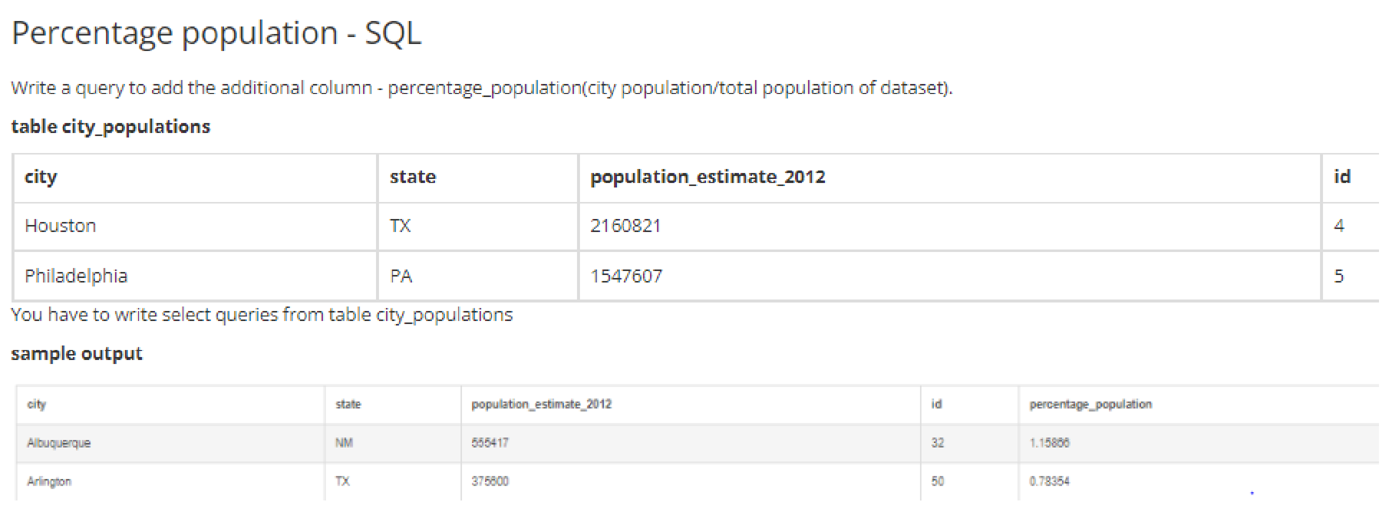
from city_populations

where population_estimate_2012 >

```
(select avg(population_estimate_2012) as num_cities from city_popul
ations where state = "IL")
```

## 4. Percentage population - SQL

Write a query to **Add the Additional Column** - percentage_population(city population / total population of dataset).



Percentage population - SQL

Write a query to add the additional column - percentage_population(city population/total population of dataset).

**table city_populations**

| city | state | population_estimate_2012 | id |
|---|---|---|---|
| Houston | TX | 2160821 | 4 |
| Philadelphia | PA | 1547607 | 5 |

You have to write select queries from table city_populations

**sample output**

| city | state | population_estimate_2012 | id | percentage_population |
|---|---|---|---|---|
| Albuquerque | NM | 555417 | 32 | 1.15866 |
| Arlington | TX | 375600 | 50 | 0.78354 |

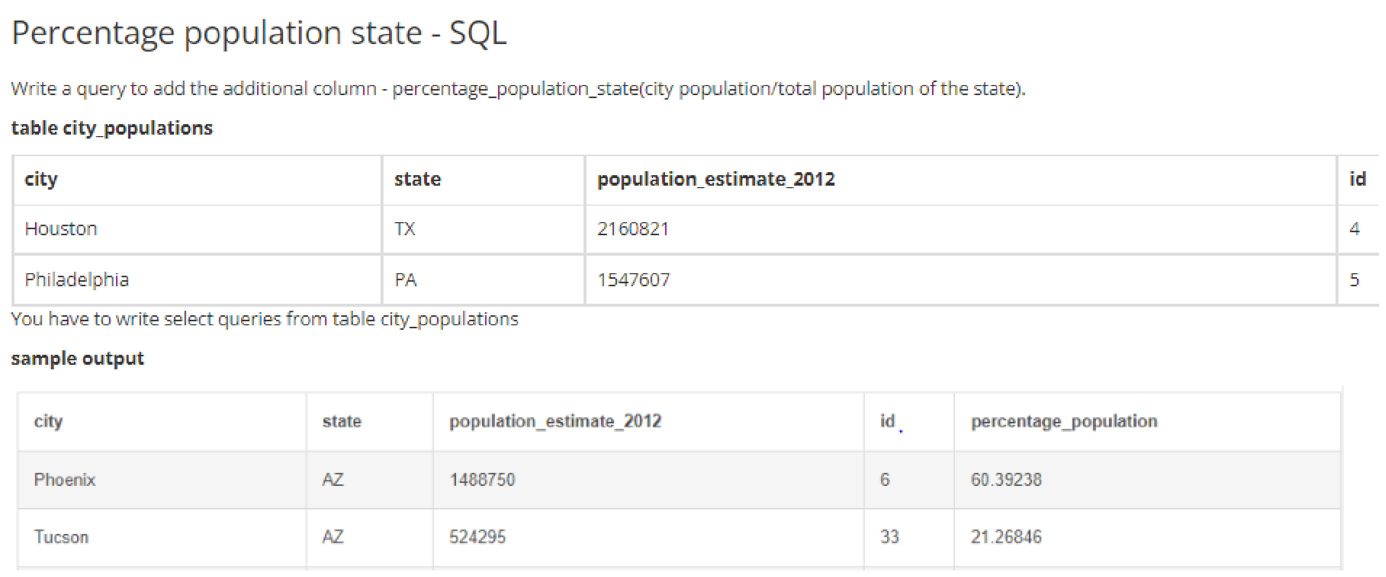
Solution:

SELECT *, 100.0 * population_estimate_2012 / (SELECT SUM(population_estimate_2012) FROM city_populations) AS percentage_population

FROM city_populations

## 5. Percentage population state - SQL -------------------------->""NOT UNDERSTOOD ""

Write a query to add the additional column - percentage_population_state(city population/total population of the state).



Percentage population state - SQL

Write a query to add the additional column - percentage_population_state(city population/total population of the state).

**table city_populations**

| city | state | population_estimate_2012 | id |
|---|---|---|---|
| Houston | TX | 2160821 | 4 |
| Philadelphia | PA | 1547607 | 5 |

You have to write select queries from table city_populations

**sample output**

| city | state | population_estimate_2012 | id | percentage_population |
|---|---|---|---|---|
| Phoenix | AZ | 1488750 | 6 | 60.39238 |
| Tucson | AZ | 524295 | 33 | 21.26846 |

Solution:

SELECT a.*,100.0 * population_estimate_2012/state_population AS percentage_population

FROM city_populations a

LEFT JOIN (SELECT state, SUM(population_estimate_2012) AS state_population FROM city_populations

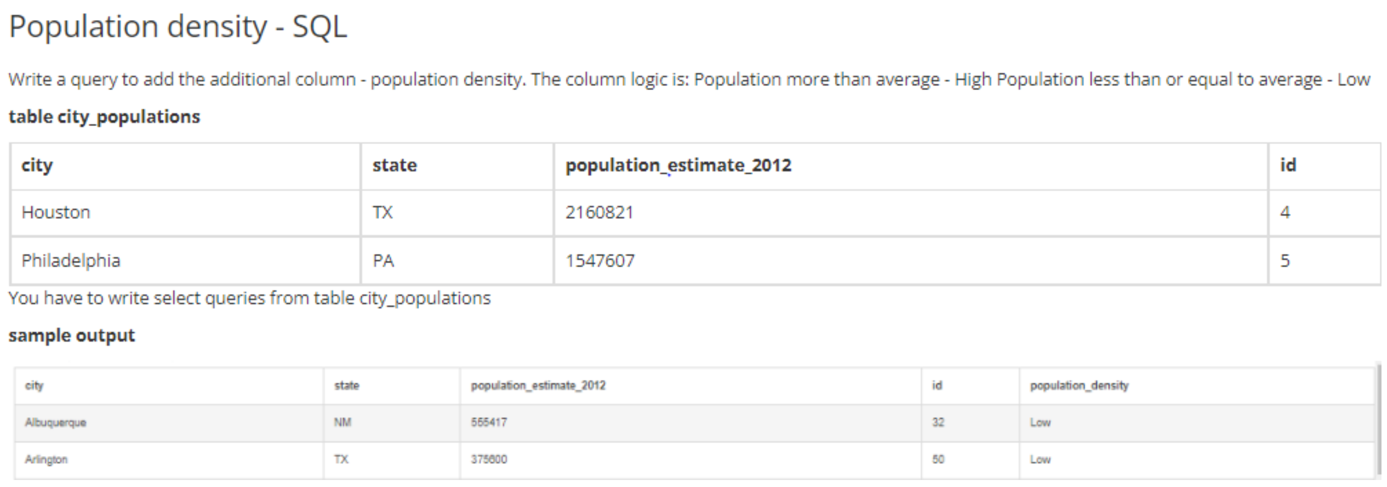
GROUP BY state) b

ON a.state = b.state

ORDER BY a.state

**6. Population density - SQL =======>>>> READ QUESTION Properly**

Write a query to add the additional column - population density. The column logic is: Population more than average - High Population less than or equal to average - Low



Population density - SQL

Write a query to add the additional column - population density. The column logic is: Population more than average - High Population less than or equal to average - Low

**table city_populations**

| city | state | population_estimate_2012 | id |
|---|---|---|---|
| Houston | TX | 2160821 | 4 |
| Philadelphia | PA | 1547607 | 5 |

You have to write select queries from table city_populations

**sample output**

| city | state | population_estimate_2012 | id | population_density |
|---|---|---|---|---|
| Albuquerque | NM | 555417 | 32 | Low |
| Arlington | TX | 375600 | 50 | Low |

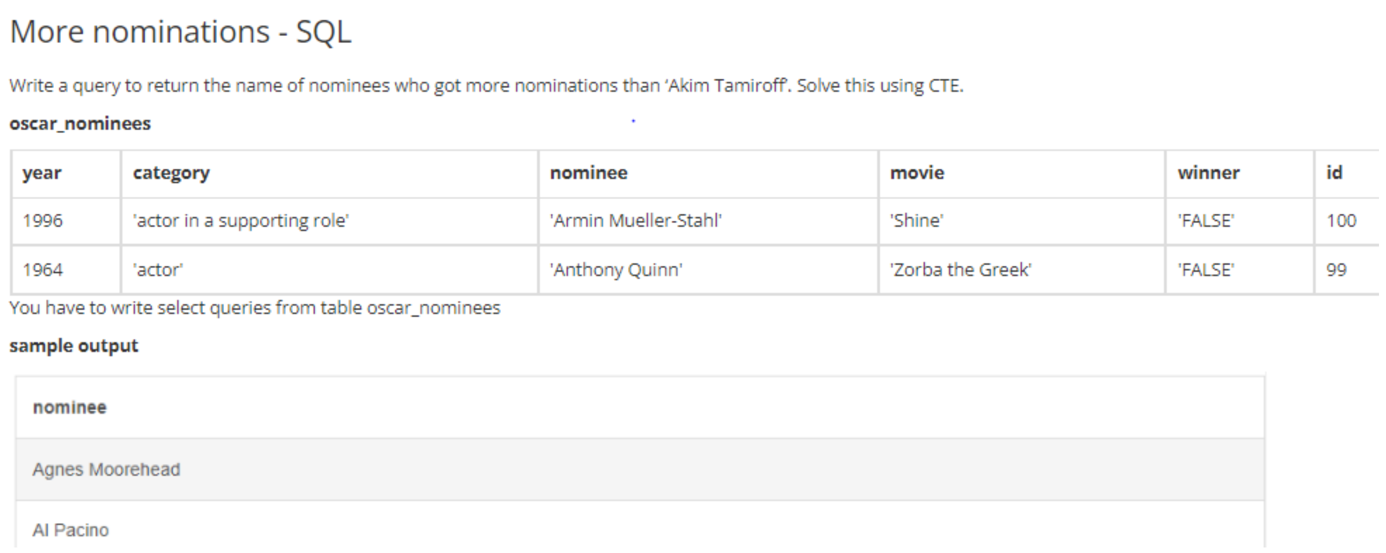
Solution:

SELECT *,CASE WHEN population_estimate_2012 > (SELECT AVG(population_estimate_2012) FROM city_populations) THEN 'High' ELSE 'Low' END AS population_density

FROM city_populations

**7. More nominations - SQL ------------------------->"".......NOT UNDERSTOOD ...........""**

Write a query to return the name of nominees who got more nominations than ‘Akim Tamiroff’. Solve this using CTE.



More nominations - SQL

Write a query to return the name of nominees who got more nominations than 'Akim Tamiroff'. Solve this using CTE.

**oscar_nominees**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 1996 | 'actor in a supporting role' | 'Armin Mueller-Stahl' | 'Shine' | 'FALSE' | 100 |
| 1964 | 'actor' | 'Anthony Quinn' | 'Zorba the Greek' | 'FALSE' | 99 |

You have to write select queries from table oscar_nominees

**sample output**

| nominee |
|---|
| Agnes Moorehead |
| Al Pacino |

Solution:

WITH nominees AS (

SELECT nominee,COUNT(*) AS nomination_count

FROM oscar_nominees

GROUP BY nominee

)

SELECT nominee

FROM nominees

WHERE nomination_count > (SELECT COUNT(*) FROM oscar_nominees WHERE nominee IN ('Akim Tamiroff'))

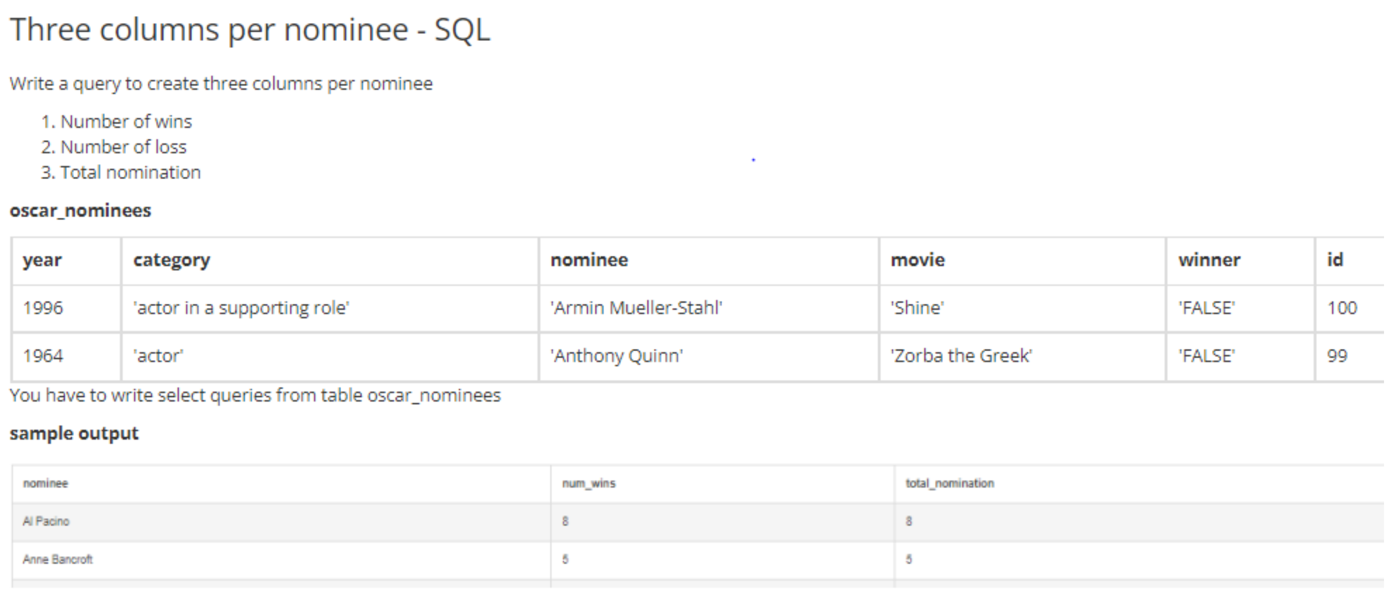
**8. Three columns per nominee - SQL**

Write a query to **------->>>Create three columns per nominee**

Number of wins

Number of loss

Total nomination



## Three columns per nominee - SQL

Write a query to create three columns per nominee

1. Number of wins
2. Number of loss
3. Total nomination

**oscar_nominees**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 1996 | 'actor in a supporting role' | 'Armin Mueller-Stahl' | 'Shine' | 'FALSE' | 100 |
| 1964 | 'actor' | 'Anthony Quinn' | 'Zorba the Greek' | 'FALSE' | 99 |

You have to write select queries from table oscar_nominees

**sample output**

| nominee | num_wins | total_nomination |
|---|---|---|
| Al Pacino | 8 | 8 |
| Anne Bancroft | 5 | 5 |

Solution:

SELECT nominee,

SUM(CASE WHEN winner = true THEN 1 ELSE 0 END) AS num_wins,

SUM(CASE WHEN winner = false THEN 1 ELSE 0 END) AS num_wins,

COUNT(*) AS total_nomination

FROM oscar_nominees

GROUP BY nominee

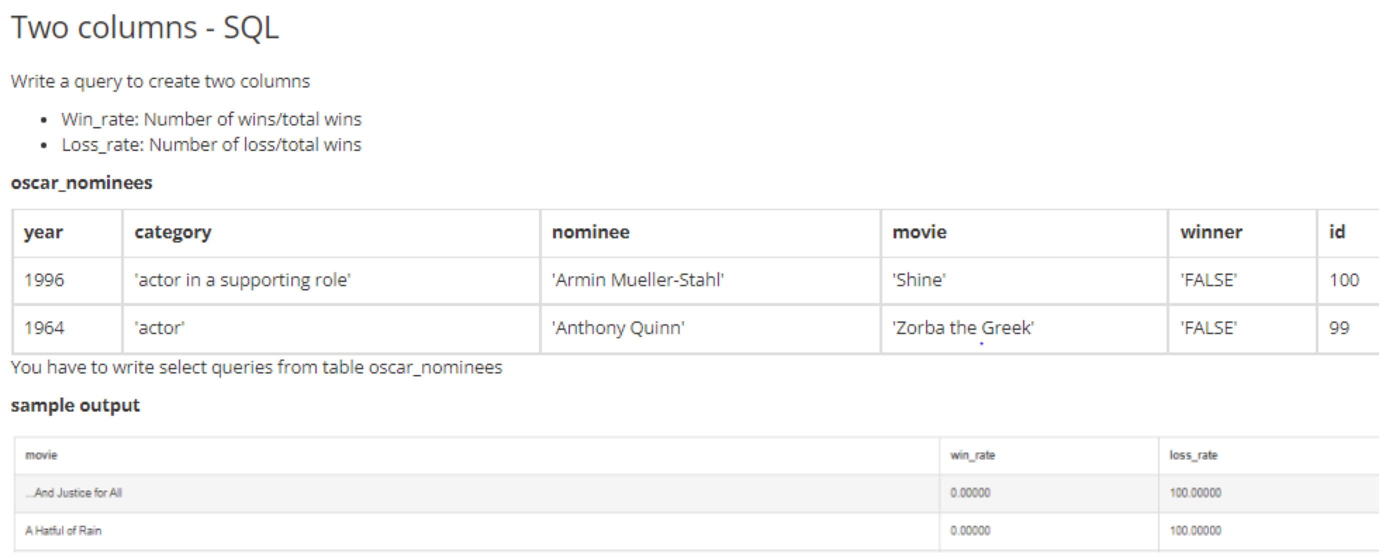
ORDER BY total_nomination DESC

### 9.Two columns - SQL

Write a query to **---------->create two columns**

Win_rate: Number of wins / total wins

Loss_rate: Number of loss / total wins



Two columns - SQL

Write a query to create two columns

- Win_rate: Number of wins/total wins
- Loss_rate: Number of loss/total wins

**oscar_nominees**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 1996 | 'actor in a supporting role' | 'Armin Mueller-Stahl' | 'Shine' | 'FALSE' | 100 |
| 1964 | 'actor' | 'Anthony Quinn' | 'Zorba the Greek' | 'FALSE' | 99 |

You have to write select queries from table oscar_nominees

**sample output**

| movie | win_rate | loss_rate |
|---|---|---|
| ...And Justice for All | 0.00000 | 100.00000 |
| A Hatful of Rain | 0.00000 | 100.00000 |

Solution:

SELECT movie,

100.0 * SUM(CASE WHEN winner = true THEN 1 ELSE 0 END)/COUNT(*) AS win_rate,

100.0 * SUM(CASE WHEN winner = false THEN 1 ELSE 0 END)/COUNT(*) AS loss_rate

FROM oscar_nominees

GROUP BY Movie

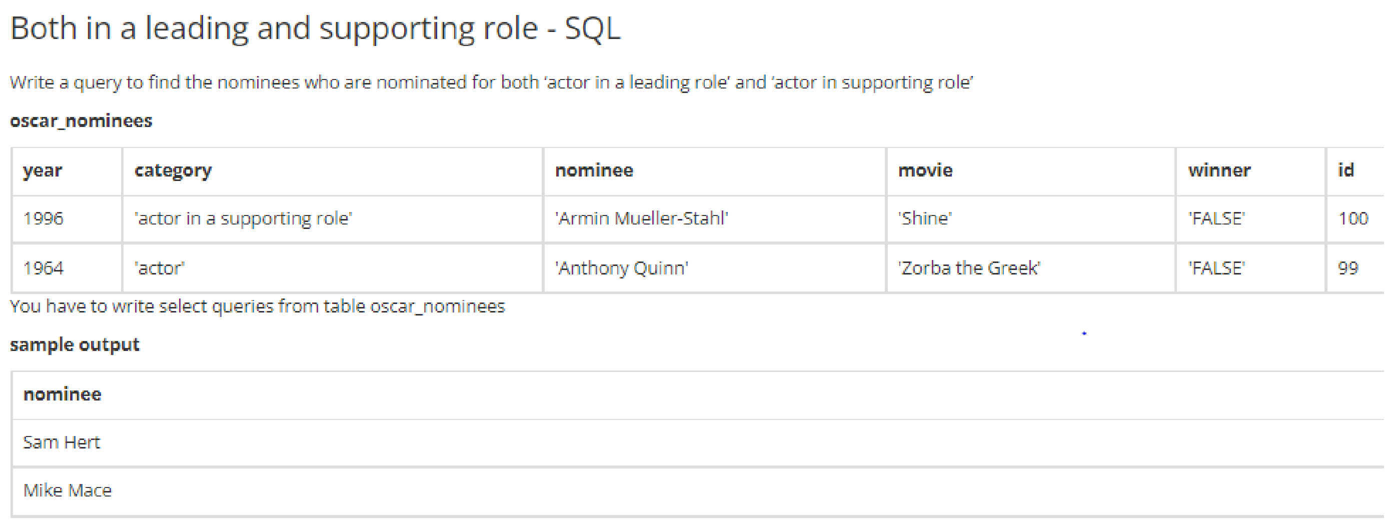
In [ ]:

```

```

# SQL Logic Building - 6

### 1. Both in a leading and supporting role - SQL

Write a query to find the nominees who are nominated for both ‘actor in a leading role’ and ‘actor in supporting role’



## Both in a leading and supporting role - SQL

Write a query to find the nominees who are nominated for both 'actor in a leading role' and 'actor in supporting role'

**oscar_nominees**

| year | category | nominee | movie | winner | id |
|---|---|---|---|---|---|
| 1996 | 'actor in a supporting role' | 'Armin Mueller-Stahl' | 'Shine' | 'FALSE' | 100 |
| 1964 | 'actor' | 'Anthony Quinn' | 'Zorba the Greek' | 'FALSE' | 99 |

You have to write select queries from table oscar_nominees

**sample output**

| nominee |
|---|
| Sam Hert |
| Mike Mace |

Solution:

SELECT DISTINCT

FROM oscar_nominees

WHERE nominee IN (SELECT DISTINCT nominee FROM oscar_nominees WHERE category IN ('actor in a supporting role'))

```
                                                  AND cat
  egory IN ('actor in a leading role')
```

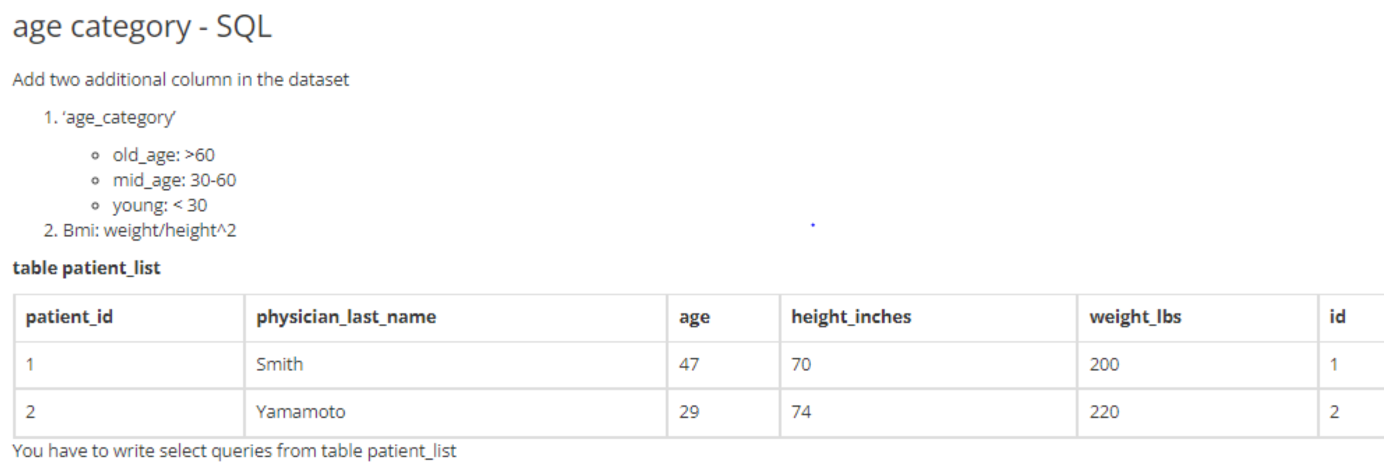
## 2. age category - SQL

Add two additional column in the dataset

1.'age_category'

old_age: >60 mid_age: 30-60 young: < 30

2.Bmi: weight/height^2



## age category - SQL

Add two additional column in the dataset

1. 'age_category'
   - old_age: >60
   - mid_age: 30-60
   - young: < 30
2. Bmi: weight/height^2

**table patient_list**

| patient_id | physician_last_name | age | height_inches | weight_lbs | id |
|---|---|---|---|---|---|
| 1 | Smith | 47 | 70 | 200 | 1 |
| 2 | Yamamoto | 29 | 74 | 220 | 2 |

You have to write select queries from table patient_list

Solution:

SELECT *,

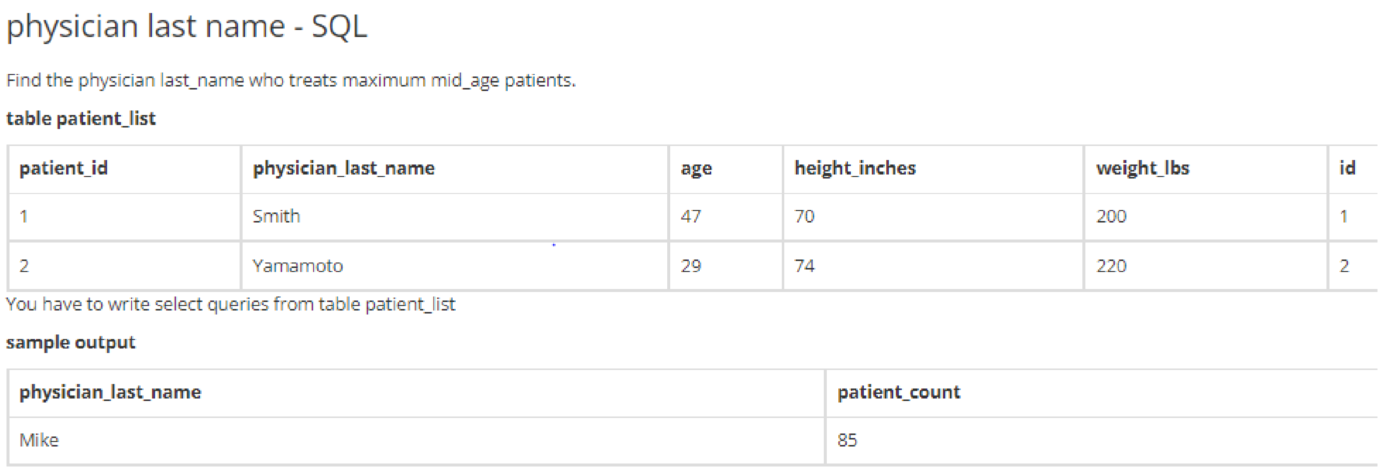
CASE WHEN age > 60 THEN 'old_age'

```
    WHEN age BETWEEN 30 AND 60 THEN 'mid_age

    ELSE 'young' END AS age_category,1.0 * weight_lbs/(height_inches *
     height_inches) AS BMI
```

FROM patient_list

### 3. physician last name - SQL

Find the physician last_name who treats maximum mid_age patients.



physician last name - SQL

Find the physician last_name who treats maximum mid_age patients.

**table patient_list**

| patient_id | physician_last_name | age | height_inches | weight_lbs | id |
|---|---|---|---|---|---|
| 1 | Smith | 47 | 70 | 200 | 1 |
| 2 | Yamamoto | 29 | 74 | 220 | 2 |

You have to write select queries from table patient_list

**sample output**

| physician_last_name | patient_count |
|---|---|
| Mike | 85 |

Solution:

SELECT physician_last_name, COUNT(*) AS patient_count

FROM (SELECT *,

```
            CASE WHEN age > 60 THEN 'old_age'

            WHEN age BETWEEN 30 AND 60 THEN 'mid_age'

            ELSE 'young' END AS age_category

            FROM patient_list) a
```

WHERE age_category = 'mid_age'

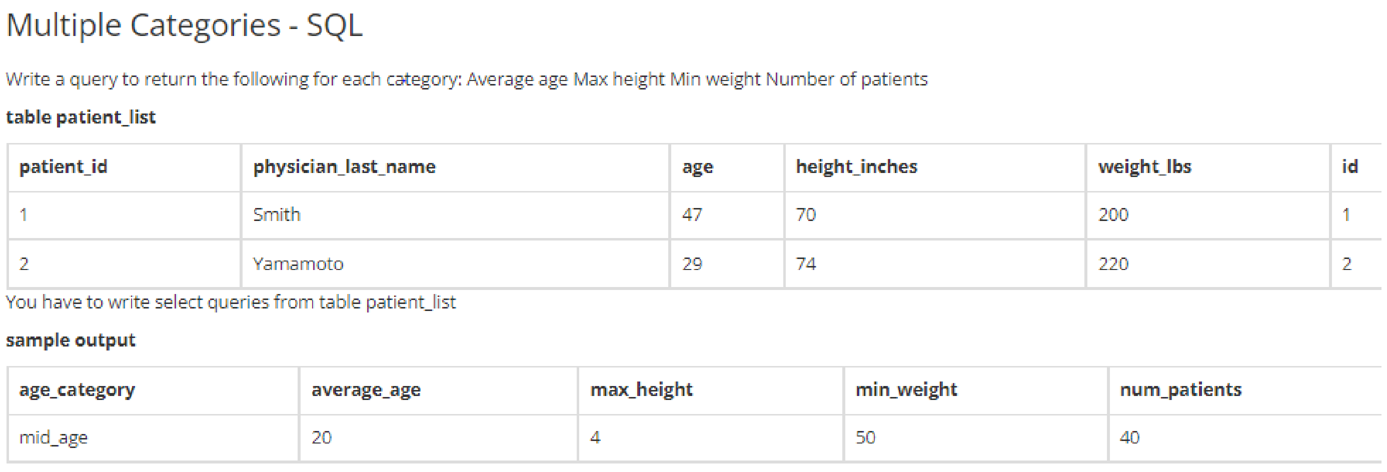
GROUP BY physician_last_name

ORDER BY patient_count DESC

LIMIT 1

## 4. Multiple Categories - SQL

Write a query to return the following for **each category: Average age Max height Min weight Number of patients**



Multiple Categories - SQL

Write a query to return the following for each category: Average age Max height Min weight Number of patients

**table patient_list**

| patient_id | physician_last_name | age | height_inches | weight_lbs | id |
|---|---|---|---|---|---|
| 1 | Smith | 47 | 70 | 200 | 1 |
| 2 | Yamamoto | 29 | 74 | 220 | 2 |

You have to write select queries from table patient_list

**sample output**

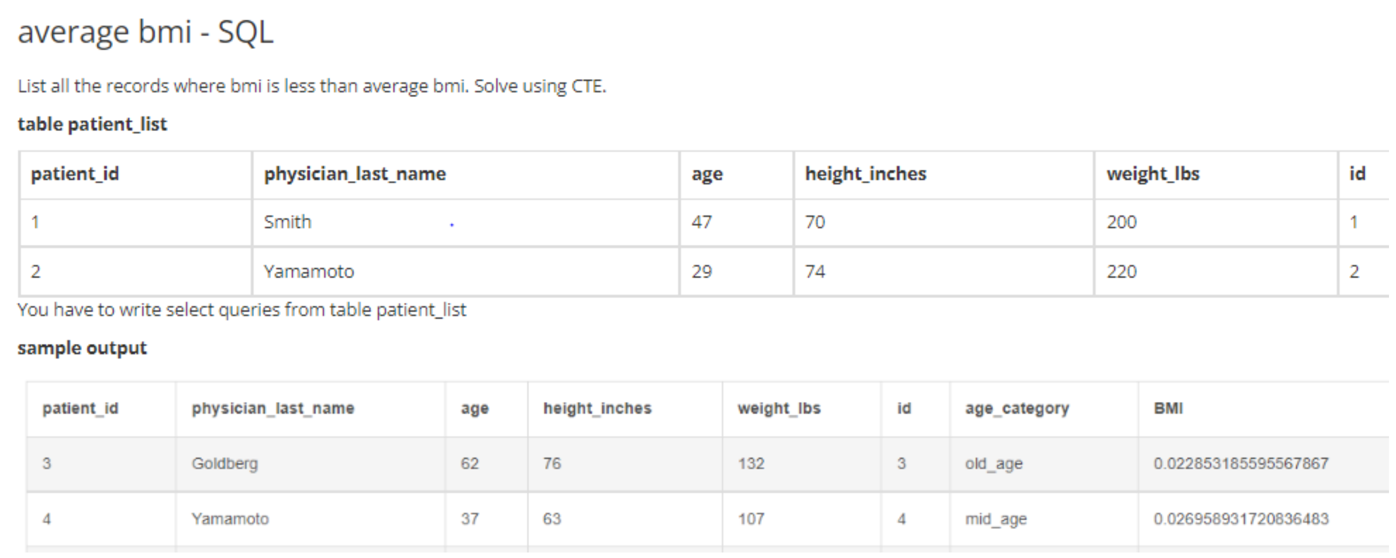
| age_category | average_age | max_height | min_weight | num_patients |
|---|---|---|---|---|
| mid_age | 20 | 4 | 50 | 40 |

Solution:

```
SELECT age_category, AVG(age) AS average_age, MAX(height_inches) AS max_height,
MIN(weight_lbs) AS min_weight, COUNT(id) AS num_patients

FROM (SELECT *, CASE WHEN age > 60 THEN 'old_age'

                WHEN age BETWEEN 30 AND 60 THEN 'mid_age'

                ELSE 'young' END AS age_category

                FROM patient_list) a

GROUP BY age_category
```

## 5. average bmi - SQL

List all the records where bmi is less than average bmi. **Solve using CTE.**


average bmi - SQL

List all the records where bmi is less than average bmi. Solve using CTE.

**table patient_list**

| patient_id | physician_last_name | age | height_inches | weight_lbs | id |
|---|---|---|---|---|---|
| 1 | Smith | 47 | 70 | 200 | 1 |
| 2 | Yamamoto | 29 | 74 | 220 | 2 |

You have to write select queries from table patient_list

**sample output**

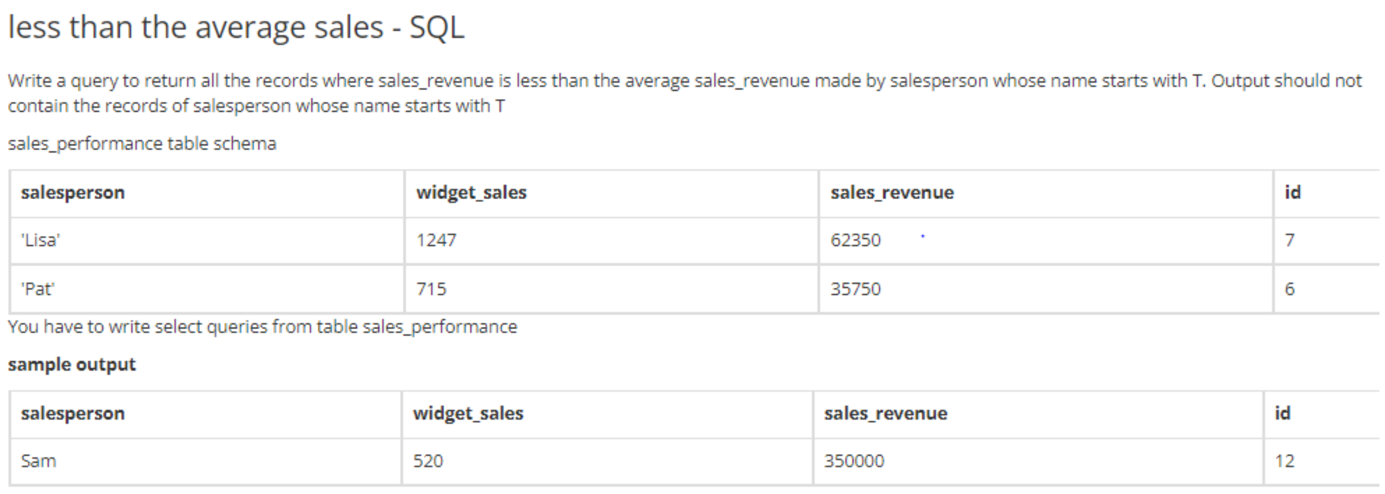
| patient_id | physician_last_name | age | height_inches | weight_lbs | id | age_category | BMI |
|---|---|---|---|---|---|---|---|
| 3 | Goldberg | 62 | 76 | 132 | 3 | old_age | 0.022853185595567867 |
| 4 | Yamamoto | 37 | 63 | 107 | 4 | mid_age | 0.026958931720836483 |

Solution:

```
WITH cte_patient AS (

SELECT *,

CASE WHEN age > 60 THEN 'old_age'

     WHEN age BETWEEN 30 AND 60 THEN 'mid_age'

     ELSE 'young' END AS age_category,1.0 * weight_lbs/(height_inches *
    height_inches) AS BMI

     FROM patient_list)

SELECT *

FROM cte_patient

WHERE BMI < (SELECT AVG(BMI) FROM cte_patient)
```

### 6. less than the average sales - SQL

Write a query to return all the records where sales_revenue is less than the average sales_revenue made by salesperson whose name starts with T. Output should not contain the records of salesperson whose name starts with T


less than the average sales - SQL

Write a query to return all the records where sales_revenue is less than the average sales_revenue made by salesperson whose name starts with T. Output should not contain the records of salesperson whose name starts with T

sales_performance table schema

| salesperson | widget_sales | sales_revenue | id |
|---|---|---|---|
| 'Lisa' | 1247 | 62350 | 7 |
| 'Pat' | 715 | 35750 | 6 |

You have to write select queries from table sales_performance

**sample output**

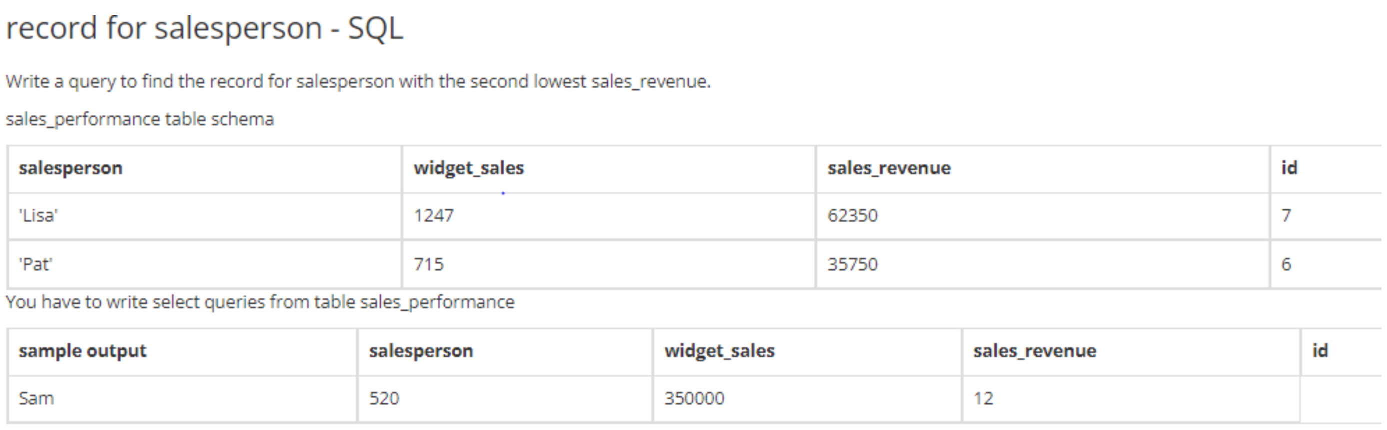
| salesperson | widget_sales | sales_revenue | id |
|---|---|---|---|
| Sam | 520 | 350000 | 12 |

Solution:

```
select *

from sales_performance

where sales_revenue < (select avg(sales_revenue)

                  from sales_performance where salesperson like
    "T%") and salesperson not like "T%"
```

**7. record for salesperson - SQL**

Write a query to find the record for salesperson with the second lowest sales_revenue.


record for salesperson - SQL

Write a query to find the record for salesperson with the second lowest sales_revenue.

sales_performance table schema

| salesperson | widget_sales | sales_revenue | id |
|---|---|---|---|
| 'Lisa' | 1247 | 62350 | 7 |
| 'Pat' | 715 | 35750 | 6 |

You have to write select queries from table sales_performance

| sample output | salesperson | widget_sales | sales_revenue | id |
|---|---|---|---|---|
| Sam | 520 | 350000 | 12 | |

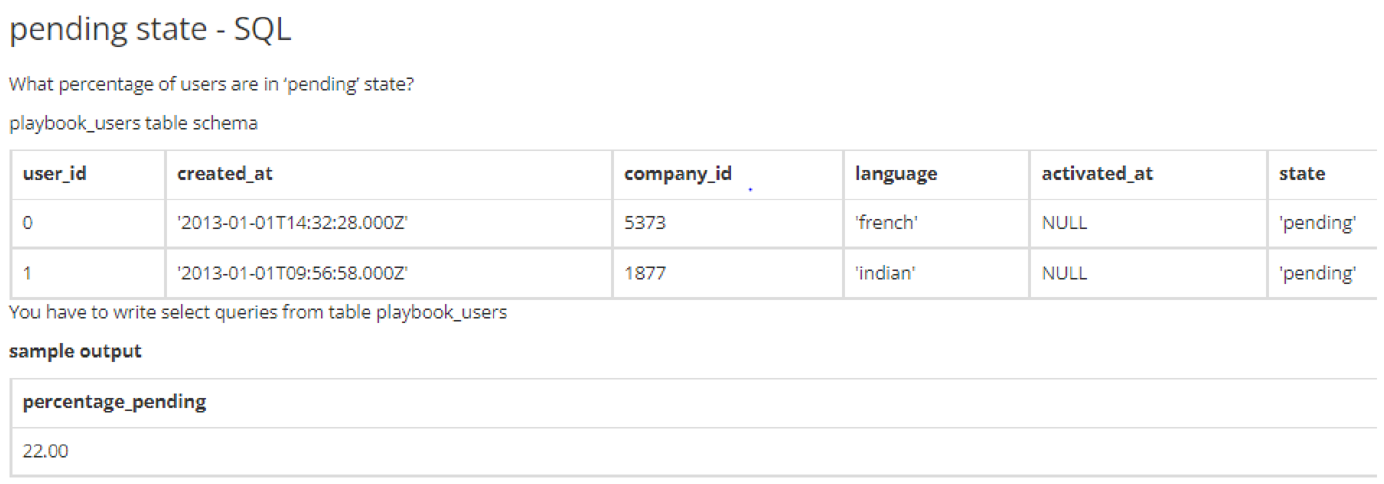
Solution:

```
SELECT *

FROM sales_performance

WHERE sales_revenue = (SELECT MIN(sales_revenue) FROM sales_performance
WHERE sales_revenue >
```

```
                                (SELECT MIN(sales_revenue) FROM sales_p
    erformance))
```

## 8. pending state - SQL

What percentage of users are in ‘pending’ state?



pending state - SQL

What percentage of users are in ‘pending’ state?

playbook_users table schema

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | '2013-01-01T14:32:28.000Z' | 5373 | 'french' | NULL | 'pending' |
| 1 | '2013-01-01T09:56:58.000Z' | 1877 | 'indian' | NULL | 'pending' |

You have to write select queries from table playbook_users

**sample output**

| percentage_pending |
|---|
| 22.00 |

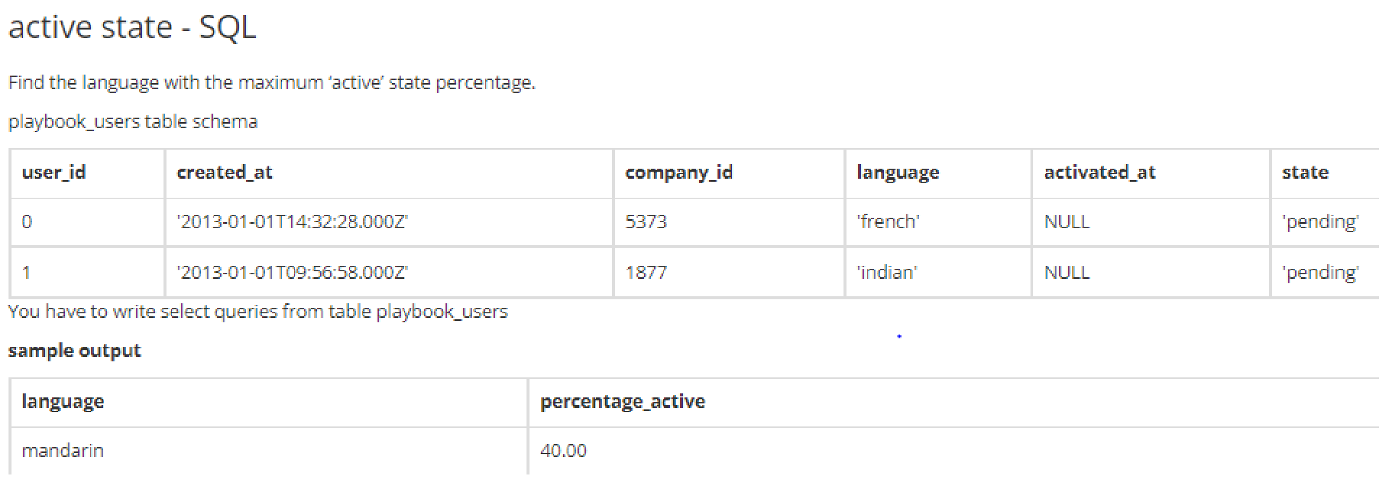
Solution:

SELECT 100.0 * SUM(CASE WHEN state = 'pending' THEN 1 ELSE 0 END )/COUNT(user_id) AS percentage_pending

FROM playbook_users

## 9. active state - SQL

Find the language with the maximum ‘active’ state percentage.



active state - SQL

Find the language with the maximum ‘active’ state percentage.

playbook_users table schema

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | '2013-01-01T14:32:28.000Z' | 5373 | 'french' | NULL | 'pending' |
| 1 | '2013-01-01T09:56:58.000Z' | 1877 | 'indian' | NULL | 'pending' |

You have to write select queries from table playbook_users

**sample output**

| language | percentage_active |
|---|---|
| mandarin | 40.00 |

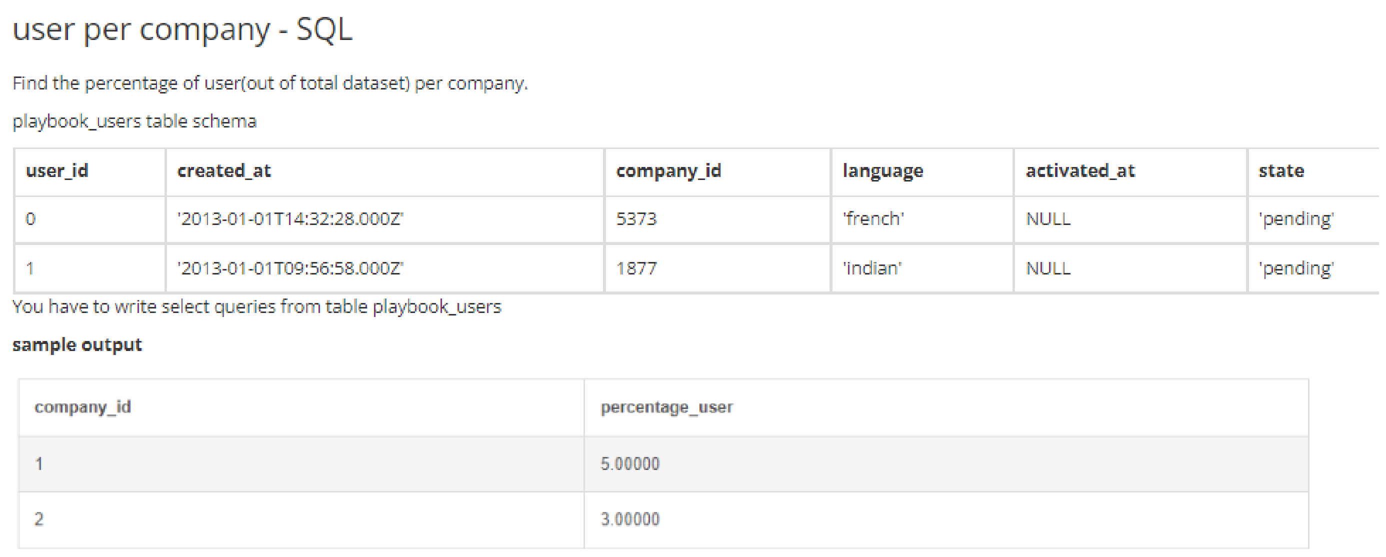
Solution:

SELECT language,100.0 * SUM(CASE WHEN state = 'active' THEN 1 ELSE 0 END )/COUNT(user_id) AS percentage_active

```
FROM playbook_users

GROUP BY language

ORDER BY percentage_active DESC

LIMIT 1
```

## 10. user per company - SQL

Find the percentage of user(out of total dataset) per company.



user per company - SQL

Find the percentage of user(out of total dataset) per company.

playbook_users table schema

| user_id | created_at | company_id | language | activated_at | state |
|---|---|---|---|---|---|
| 0 | '2013-01-01T14:32:28.000Z' | 5373 | 'french' | NULL | 'pending' |
| 1 | '2013-01-01T09:56:58.000Z' | 1877 | 'indian' | NULL | 'pending' |

You have to write select queries from table playbook_users

**sample output**

| company_id | percentage_user |
|---|---|
| 1 | 5.00000 |
| 2 | 3.00000 |

Solution:

```
SELECT company_id,100.0 * COUNT(*)/(SELECT COUNT(user_id) FROM
playbook_users) AS percentage_user

FROM playbook_users

GROUP BY 1

ORDER BY 2 DESC;
```

In [ ]: