Maria Toral

Assignment 2

1. What is the total number of documents retrieved by participants during the experiment? (Hint: your query should return only a single record, containing only a count).

**Query**: select count(\*) from docs;

**Answer**: 356

1. What is the average number of documents retrieved per participant? Note: write a single SQL query which gives you the answer. Also, when the result of the query is '5,' you should realize that that is not exactly right.

**Query:** SELECT CAST(COUNT(\*) AS FLOAT) / COUNT(DISTINCT(username)) FROM docs;

**Answer:** 5.5625

1. Which document types were involved in these retrievals? (Hint: your query should return four records, one of which is NULL)

**Query**:

SELECT DISTINCT(doc\_type)

FROM docs;

**Answer**: NULL, activity, lesson, and unit

1. What does the NULL result from the previous query signify? (No SQL needed here; just think about this and provide your answer)

**Answer**: if doc\_type is a nullable field, that means that the users can or can not enter a doc\_type. If this user let the doc\_type blank, the grouping of those documents left blank for their type would show as NULL.

1. Using only the documents table, how many different participants (usernames) did participate in the experiment? (remember; the query should return but a single(!) number)

**Query**:

SELECT COUNT(DISTINCT(username))

FROM docs;

**Answer**: 64

1. Check this result against the participants table and explain the difference. (Hint: the count from the previous question is different from the number of different usernames in the participants table. How can that be?)

**Query**:

SELECT COUNT(DISTINCT(username))

FROM subjects;

**Answer**: 63

Since both the docs table and the subjects table have the username attribute but this attribute is not a foreign key, usernames can be different on each table as long as they don’t violate the primary key constraint on each table. One username must have entered the wrong username.

1. What is the 'extra' username in the documents table which is missing from the participants table? (Hint: SQL query should return only a single record).

**SQL**:

SELECT DISTINCT(username) FROM docs

EXCEPT

SELECT DISTINCT(username) FROM subjects;

**Answer**: Johnson

1. Update the documents table to set the username of the 'extra' username in the documents table to Bang\_Bang\_Johnson and check the result of question 5 again.

**Query**:

UPADTE docs SET username = ‘Bang\_Bang\_Johnson’

WHERE username = ‘Johnson’;

**Answer**= now both counts result in 63 distinct users.

1. Which participant has the longest username and how many characters does that username have? (use a single query) (Hint: the longest username has 18 characters)

**Query**:

SELECT TOP 1 username, LEN(username)

FROM docs

ORDER BY LEN(username) DESC;

**Answer**: Dante\_Thunderstone

1. Which two participants retrieved the most documents and how many documents did each of those participants retrieve? Note: your query should return only two rows. (Hint: both of these users each retrieved 11 documents)

**Query**:

SELECT TOP 2 username, count(\*)

FROM docs

GROUP BY username

ORDER BY count(\*) DESC;

**Answer**:

Carrot- 11

MileyCyrus-11

1. How many tasks were completed for each of the two experimental conditions? (do this with a single query!!).

Hint 1: The experimental condition is not stored in the tasks table; it is stored in the participants table. You will need a join between the participants and the tasks tables to do this.

Hint 2: there is a difference of 23 between the two numbers.

**Query:**

SELECT exp\_condition, COUNT(ISNULL(task,0))

FROM subjects, tasks

WHERE subjects.username = tasks.username

GROUP BY exp\_condition;

**Answer**:

Exp\_condtion 1- 58

Exp\_condition 2-81

1. Complete the table below; show the queries which gave you these results:

|  |  |  |
| --- | --- | --- |
| Experimental condition | List (1) | Map (2) |
| number of participants  SELECT exp\_condition, count(username)  FROM subjects  GROUP BY exp\_condition; | 28 | 35 |
| number of documents retrieved  SELECT exp\_condition, count(\*) docs  FROM docs, subjects  WHERE docs.username = subjects.username  GROUP BY exp\_condition; | 154 | 202 |
| smallest number of documents retrieved by any participant (you may use several queries for this)  SELECT TOP 2 exp\_condition, docs.username, count(\*)  FROM docs, subjects  WHERE docs.username = subjects.username  GROUP BY exp\_condition, docs.username  ORDER BY count(\*) ASC; | 2 | 1 |
| largest number of documents retrieved by any participant (you may use several queries for this)  SELECT TOP 2 exp\_condition, docs.username, count(\*)  FROM docs, subjects  WHERE docs.username = subjects.username  GROUP BY exp\_condition, docs.username  ORDER BY count(\*) DESC; | 11 | 11 |
| average number of documents retrieved per participant (do not use SQL for this; just compute from the numbers you already have) | 5.5 | 5.77 |
| sigma (std. dev.) of the number of documents retrieved per participant: the SQL to compute this is as follows:  select stdev(my\_table.my\_count) from (select count(\*) as my\_count from documents, participants where participants.username = documents.username and participants.exp\_condition = 1 (...or 2) group by participants.username) as my\_table; | 2.545875386 | 2.52150415 |

Hint: check the numbers in the table for consistency with some of the other numbers you have found. For instance, The number of documents retrieved across the two conditions must equal the result of question 1.

1. This not a SQL question, but it is a natural follow up from what you computed in the previous question (plus it explains why we take stats classes!). From the results in the previous question we may conclude that participants in the 'Map' condition, on average retrieve more documents than participants in the 'List' condition. Before we draw that conclusion, however, we should ask ourselves if the apparent difference is likely the result of random effects; i.e., we must check for the statistical significance of the difference. The test that applies here is the t-test for testing equality of means in two samples.

One way to run this test is to plug in the values from the table above in an on-line t-test utility such as <https://www.graphpad.com/quickcalcs/ttest1/?Format=SD>

Run this test and provide the resulting two-tailed p-value and your conclusion on whether or not the difference in means is statistically significant at α=0.05.

**Answer:** The result of this two tailed p-value equals to 0.6756 . With this p-value we fail to reject the hypothesis that participants in the map condition retrieve more documents than those in the list category. There is no significant evidence to prove that this is true and there is no big difference in the amount of documents the two categories retrieve.

1. If we consider the two left-most bytes of an IP address to indicate the organization hosting the address, how many different host\_ips in our experiment are associated with the 128.138.\* network? (Hint: the count should be 33)

**Query**:

SELECT COUNT(DISTINCT(host\_ip))

FROM exp\_data

WHERE host\_ip LIKE ‘1\_8%’;

**Answer**: 33

1. Not a SQL question: Which organization is associated with the 128.138 addresses? How about the 129.123 ones? Hint: find a few host\_ips in the exp\_data table which are associated with these organizations and then do a reverse DNS lookup (on Windows: nslookup command). If a reverse lookup does not work, use the Internet to answer this question.

**Answer**: The organization associated with the 128.138 ip addresses is the University of Colorado at Boulder. And the one associated with the 129.123 ones is Utah State University.

1. List, in order from earliest to latest, the different dates (your SQL should return dates only, not times of day!) during which experimental data were collected and the number of exp\_data records collected on each of those dates. Again, use a single query! (Hint: your list of dates should have 13 dates)

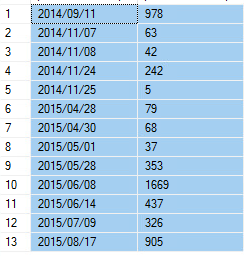
**Query**: (Assuming earliest means furthest away from current date)

SELECT CONVERT(VARCHAR(10), timestamp, 111), count(\*)

FROM exp\_data

GROUP BY CONVERT(VARCHAR(10), timestamp, 111);

**Answer**:



1. What is the daily minimum, maximum and average number of exp\_data records collected? Again, do not read or compute these numbers manually from the results in the previous question. Use SQL to do it. Hint: perhaps the easiest way to do this is in two steps: first make a new table from the results of your previous query; one which contains the totals for each day, and then query that new table for the minimum, maximum and average. You can make a new table from a query using the select ... into new-table-name from ... syntax. (Also: if the result for your average is 400, you should realize that that is not exactly correct!)

**Queries**: **(for creating a new table and inserting previous answer into this table)**

CREATE TABLE temp\_table

(exp\_day VARCHAR(15) NOT NULL

exp\_records INT NOT NULL,

PRIMARY KEY (exp\_day));

INSERT INTO temp\_table

SELECT CONVERT(VARCHAR(10), timestamp, 111), count(\*)

FROM exp\_data

GROUP BY CONVERT(VARCHAR(10), timestamp, 111);

**(for getting the minimum and maximum)**

SELECT MIN(exp\_records), MAX(exp\_records), AVG(exp\_records \*1.0)

FROM temp\_table;

**Answer**: min- 5, max- 1669, avg- 400.307692

1. How many days have passed between the first record being collected and the last (in the exp\_data table)? Note: you must write a SQL query which computes this number. Do not compute it yourself from the min and max timestamps you computed in the previous question.

**Query:**

SELECT DATEDIFF (DAY, MIN(timestamp), MAX(timestamp))

FROM exp\_data;

**Answer:** 340 days