KJC4 4CREDITS

HOMEWORK 10

QUERIES AND NARRATIVES

1. modifying a numerical value as part of the select statement

Modify\_numeric.csv

sqlite> SELECT visited.dated,survey.person,survey.quant,survey.reading,abs(round(survey.reading,0)) as whole\_positive\_reading

...> FROM visited JOIN survey

...> WHERE survey.taken=visited.id

...> AND survey.person IS NOT NULL

...> AND visited.dated IS NOT NULL;

This query selects columns from the visited and survey tables based on specified conditions. The aim is to modify the reading column by rounding the values to whole numbers and finding the absolute value(remove all negative signs). Those values are then put in a new column whole\_positive\_reading which could be useful for basis of comparison or to present the data in a new format.

2) concatenating text data within the select statement

concatenate\_text.csv

sqlite> SELECT survey.taken, survey.person,person.personal||" "||person.family as full\_name,survey.quant,survey.reading

...> FROM survey JOIN person

...> WHERE survey.person = person.id;

This query selects columns from the survey and the person tables that satisfy specified conditions. The aim is to do a select statement that can get the full name of each person by concatenating two text columns.

3) joining three tables

join\_3\_tables.csv

sqlite> SELECT visited.dated,visited.site,survey.person,person.personal,person.family,survey.quant,survey.reading

...> FROM person JOIN survey JOIN visited

...> WHERE person.id = survey.person

...> AND visited.id=survey.taken

...> AND visited.dated IS NOT NULL;

This query selects columns from three tables: person, survey and visited based on some specified conditions . The aim here was to give a full record of each site visit, including full details of who visited and when.

4) aggregating data using group by with multiple groups.

aggregate\_groupby.csv

sqlite> SELECT visited.dated,visited.site,survey.taken,survey.person,count(reading)

...> FROM visited JOIN survey

...> WHERE visited.id=survey.taken

...> GROUP BY survey.taken,survey.person;

This query selects columns from two tables visited and survey based on specified conditions. Each date in the visited table is associated with a particular id . The aim here is to find the number of readings each person took during each specific visit to the site, regardless of the type of readings by making use of GROUP BY.