Missing Data

Sorting by Known Date

* Write a query that sorts the records in Visited by date, omitting entries for which the date is not known (i.e., is null).

Solution: SELECT \* FROM Visited WHERE dated IS NOT NULL ORDER BY dated ASC;

Graphical user interface, text, application

Description automatically generated

NULL in a Set

* What do you expect the following query to produce?

SELECT \* FROM Visited WHERE dated IN ('1927-02-08', NULL);

Solution:

Expected the above query to return rows where dated is either ‘1927-02-08’ or NULL. Instead it only returned rows where dated is ‘1927-02-08’, the same as we would get from this simpler query:

SELECT \* FROM Visited WHERE dated IN ('1927-02-08');

The reason is that the IN operator works with a set of values, but NULL is by definition not a value and is therefore simply ignored.

If we wanted to actually include NULL, we would have to rewrite the query to use the IS NULL condition:

SELECT \* FROM Visited WHERE dated = '1927-02-08' OR dated IS NULL;

Graphical user interface, text, application, email

Description automatically generated

Pros and Cons of Sentinels

* Some database designers prefer to use a [sentinel value](https://swcarpentry.github.io/sql-novice-survey/reference.html#sentinel-value) to mark missing data rather than null. For example, they will use the date “0000-00-00” to mark a missing date, or -1.0 to mark a missing salinity or radiation reading (since actual readings cannot be negative). What does this simplify? What burdens or risks does it introduce?

Solution: The sentinel value is chosen when the input value of the record is empty or not acceptable. As the sentinel value can be an integer, it can be confusing with the database records and can change the output of the query.

Counting Temperature Readings

* How many temperature readings did Frank Pabodie record, and what was their average value?

Solution: SELECT count(reading), avg(reading) FROM Survey WHERE quant = 'temp' AND person = 'pb';

Graphical user interface, text, application

Description automatically generated

Averaging with NULL

* The average of a set of values is the sum of the values divided by the number of values. Does this mean that the avg function returns 2.0 or 3.0 when given the values 1.0, null, and 5.0?

Solution: The answer is 3.0. NULL is not a value; it is the absence of a value. As such it is not included in the calculation.

Graphical user interface, text, application, email

Description automatically generated

What Does This Query Do?

* We want to calculate the difference between each individual radiation reading and the average of all the radiation readings. We write the query:

*SELECT reading - avg(reading) FROM Survey WHERE quant = 'rad';*

What does this actually produce, and can you think of why?

Using the group\_concat function

* The function group\_concat(field, separator) concatenates all the values in a field using the specified separator character (or ‘,’ if the separator isn’t specified). Use this to produce a one-line list of scientists’ names, such as:

Combining Data

Listing Radiation Readings

Write a query that lists all radiation readings from the DR-1 site.

Solution: SELECT

Survey.reading

FROM

Site

JOIN

Visited

JOIN

Survey

ON Site.name = Visited.site

AND Visited.id = Survey.taken

WHERE

Site.name = 'DR-1'

AND Survey.quant = 'rad';

Graphical user interface, text, application

Description automatically generated

Where’s Frank?

Write a query that lists all sites visited by people named “Frank”.

Solution:

SELECT

DISTINCT Site.name

FROM

Site

JOIN Visited

JOIN Survey

JOIN Person ON Site.name = Visited.site

AND Visited.id = Survey.taken

AND Survey.person = Person.id

WHERE

Person.personal = 'Frank';

Graphical user interface, text, application, email

Description automatically generated

Reading Queries

Describe in your own words what the following query produces:

SELECT Site.name FROM Site JOIN Visited

ON Site.lat < -49.0 AND Site.name = Visited.site AND Visited.dated >= '1932-01-01';

Solution:

The Query selects the site name based on the conditions that the Site’s lat is less that -49.0 and the visited site’s name is the same and is after the date 1932-01-01

Graphical user interface, text, application

Description automatically generated

Who Has Been Where?

Write a query that shows each site with exact location (lat, long) ordered by visited date, followed by personal name and family name of the person who visited the site and the type of measurement taken and its reading. Please avoid all null values. Tip: you should get 15 records with 8 fields.

Solution:

SELECT Site.name, Site.lat, Site.long, Person.personal, Person.family, Survey.quant, Survey.reading, Visited.dated

FROM

Site

JOIN

Visited

JOIN

Survey

JOIN

Person

ON Site.name = Visited.site

AND Visited.id = Survey.taken

AND Survey.person = Person.id

WHERE

Survey.person IS NOT NULL

AND Visited.dated IS NOT NULL

ORDER BY

Visited.dated;

Text

Description automatically generated

# Data Hygiene

Identifying Atomic Values

* Which of the following are atomic values? Which are not? Why?

New Zealand

87 Turing Avenue

January 25, 1971

the XY coordinate (0.5, 3.3)

Solution:

New Zealand is the only clear-cut atomic value.

The address and the XY coordinate contain more than one piece of information which should be stored separately:

House number, street name

X coordinate, Y coordinate

The date entry is less clear cut, because it contains month, day, and year elements. However, there is a DATE datatype in SQL, and dates should be stored using this format. If we need to work with the month, day, or year separately, we can use the SQL functions available for our database software

* Identifying a Primary Key

What is the primary key in this table? I.e., what value or combination of values uniquely identifies a record?

latitude longitude date temperature

57.3 -22.5 2015-01-09 -14.2

Solution:

Latitude, longitude, and date are all required to uniquely identify the temperature record.

# Creating and Modifying Data

Replacing NULL

* Write an SQL statement to replace all uses of null in Survey.person with the string 'unknown'.

Solution:

UPDATE Survey SET person = 'unknown' WHERE person IS NULL;

* Backing Up with SQL

SQLite has several administrative commands that aren’t part of the SQL standard. One of them is .dump, which prints the SQL commands needed to re-create the database. Another is .read, which reads a file created by .dump and restores the database. A colleague of yours thinks that storing dump files (which are text) in version control is a good way to track and manage changes to the database. What are the pros and cons of this approach? (Hint: records aren’t stored in any particular order.)

Solution:

Advantages

A version control system will be able to show differences between versions of the dump file; something it can’t do for binary files like databases

A VCS only saves changes between versions, rather than a complete copy of each version (save disk space)

The version control log will explain the reason for the changes in each version of the database

Disadvantages

Artificial differences between commits because records don’t have a fixed order