

03/08/2021

Sharing and Reuse of Data

Data sharing is the practice of making data used for research publicly available

- Promotes transparency
- Increases confidence in the scientific results
- Helps in replication/catches errors

A lot of scientific data is still kept secret

- For privacy reasons
- Because researchers don't want to get "scooped" from someone else using their data

Some research journals now actually require that data used in their publications be made public (to some degree)

Open data is becoming more a more desirable practice

Data.gov is outgrowth of this Open Data movement

Using shared data can help increase sample sizes, especially for data that may be difficult to collect

Data reuse – using data for a purpose other than the one it was collected for

Many companies sell your data to someone else is the most common example of data reuse

Data sharing often done through data repositories:

- Have the data described (the meaning of variables, etc.) to make the data easily reused by someone else
- Metadata
- Data dictionaries – provide detailed definitions of the variables, how they were collected, methodologies, data types, etc.

Joins in SQL (specifically in SQLite)

Join types: are Inner join (natural join), Left (outer) join, Cross join

If you don't specify, Inner join is the default

Left Join keeps all data in the "left" table (the first one mentioned in the SQL command), then adds in data from the "right" table that has a match in the left table (discards unmatched things from the right table (second table in command)).

Right Join does the opposite : discards data in the left table that does not match the right table.

Full Outer Join: matches data in both tables, but discards

Cross join : use this very, very sparingly, especially if you have a lot of data
Pairs every row in table 1 with every row in table 2.

{1, 2, 3, 4} x {a, b, c, d}

{(1,a), (1,b), (1, c), (1,d), (2,a), (2,b), (2,c), (2,d), (3,a), (3,b), (3,c), (3,d), (4,a), (4,b), (4,c), (4,d)}

```
SELECT * FROM tablename1 INNER JOIN tablename2 ON tablename1.fieldname =  
tablename2.fieldname;
```

```
SELECT * FROM tablename1 INNER JOIN tablename2 USING(fieldname);
```

```
SELECT * FROM tablename1 NATURAL JOIN tablename2;
```

Left join works basically the same way as INNER JOIN

```
SELECT * FROM tablename1 LEFT JOIN tablename2 ON tablename1.fieldname = tablename2.fieldname;
```

Or: LEFT OUTER JOIN