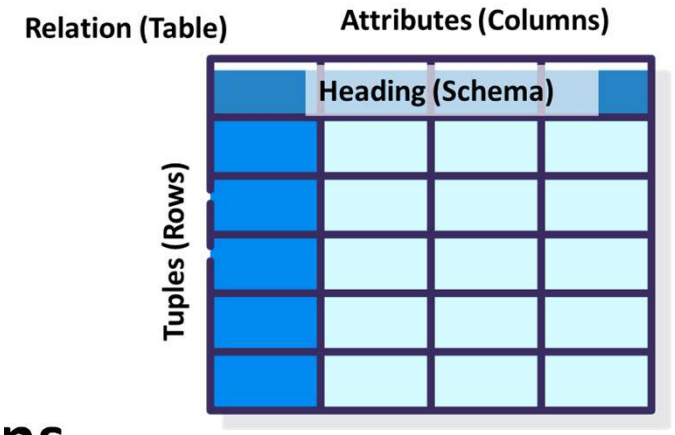
Relational Expressions for Data Transformation and Computation

DAVID PRATTEN, LUKE MATHIESON

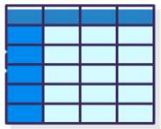
Icons used under license from https://flaticon.com
Images used under license from https://unsplash.com and https://midjourney.com



Data Relations

Predicate Natural Language statement, which is true for every row
e.g. “Cars registered in NSW” `[NSWCars]`

Start: Empty (just a Schema)



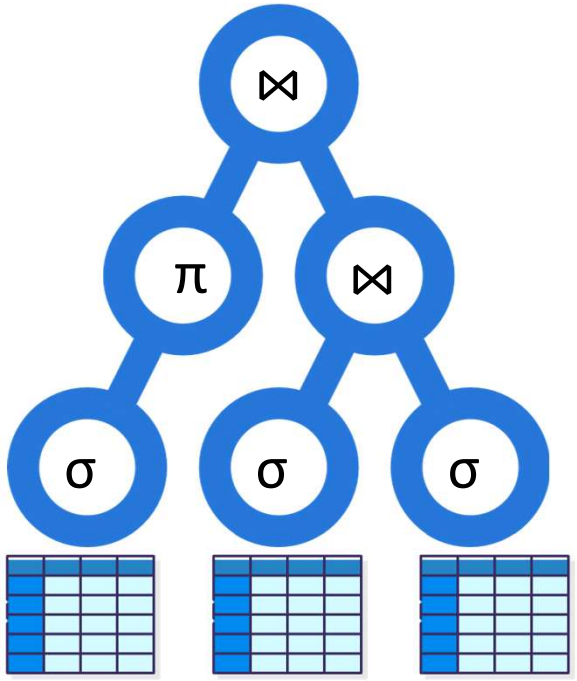
Extent: Finite

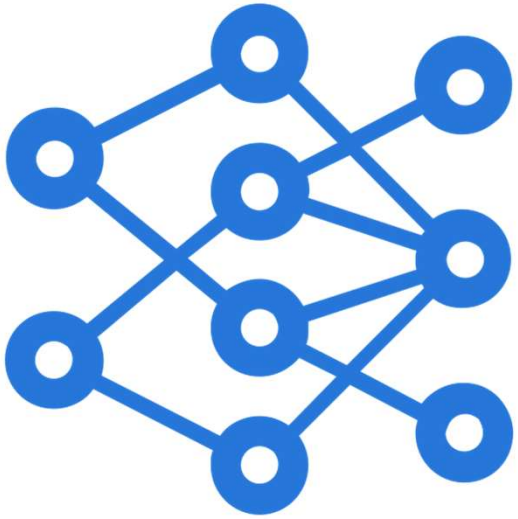
Mutating Yes: `INSERT, UPDATE, DELETE`

Queries Range over the rows that exist.

Obligations Confidentiality, Integrity

Relational Expressions



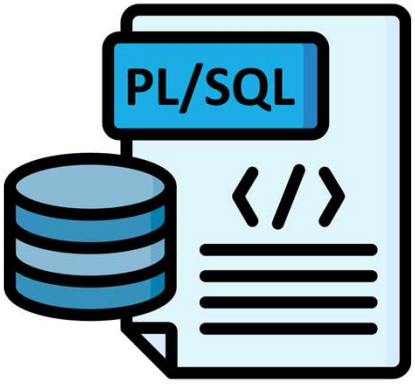


```
SELECT columns
FROM tables
WHERE expr
```

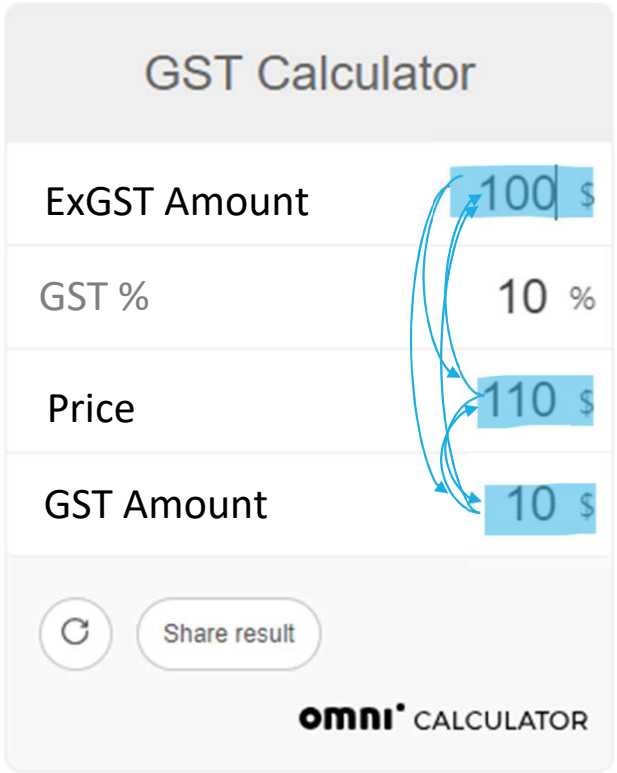
- *Codd 1970 relational model data large shared*

```
SELECT * FROM A INNER JOIN B
    ON A.I = B.I
WHERE A.J = 2
GROUP BY …
HAVING …;
```

Business Rules
Regulations
Laws



2



$(b,c) = F_1(a)$

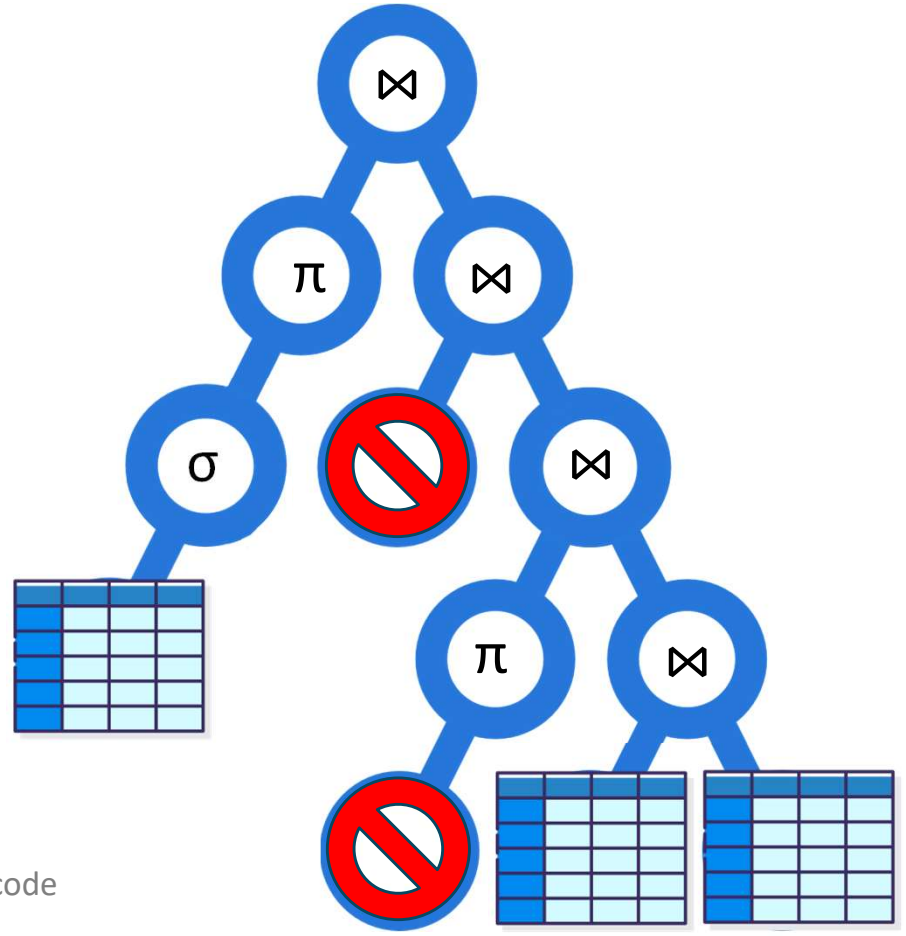
$(a,c) = F_2(b)$

$(a,b) = F_3(c)$

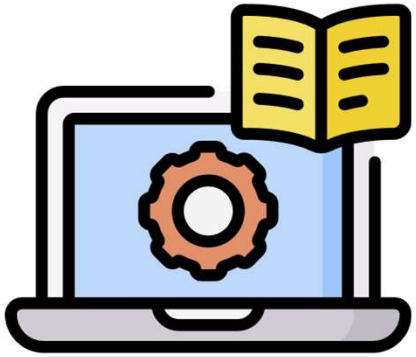
`Australian_GST(a,b,c)`

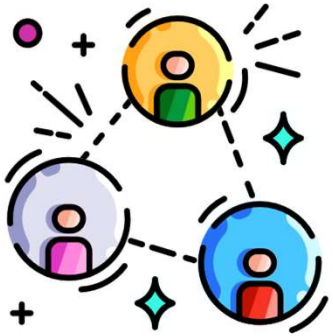
https://www.omnicalculator.com/finance/gst

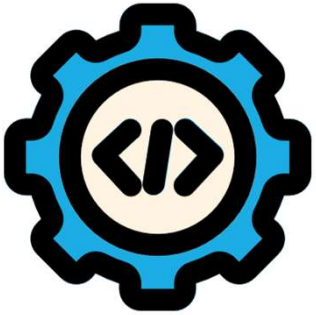
3

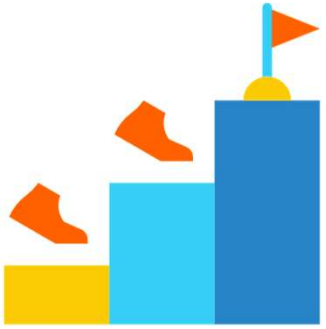


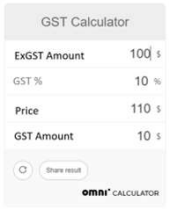
- Emani 2016 extracting equivalent sql imperative code
- hirn 2020 plsql without the pl
- zhang 2023 automated functional big data queries





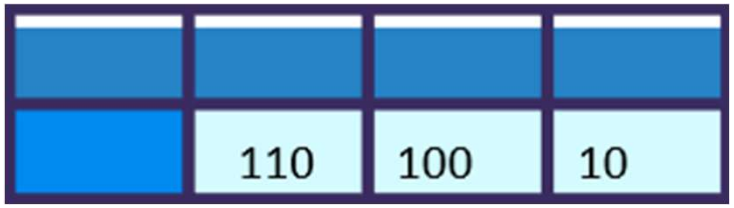


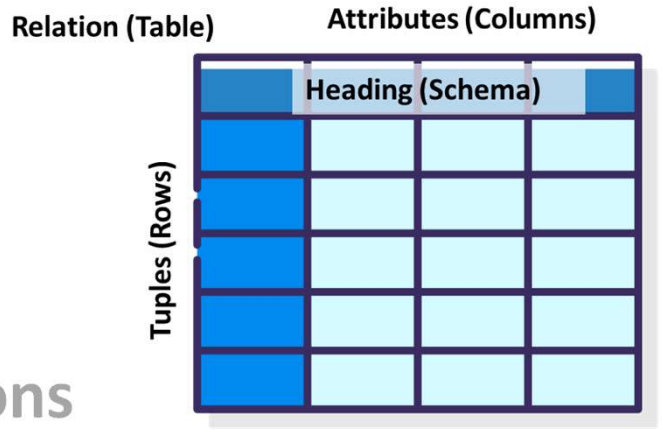




```
Australian_GST(Price, ExGSTAmount, GSTAmount)
```

```
SELECT * FROM Australian_GST WHERE Price=110;
                                   ExGSTAmount = 100;
                                   GSTAmount = 10;
```

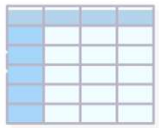


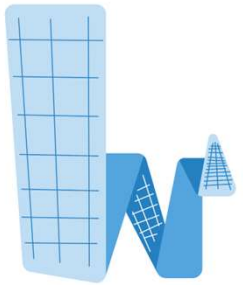


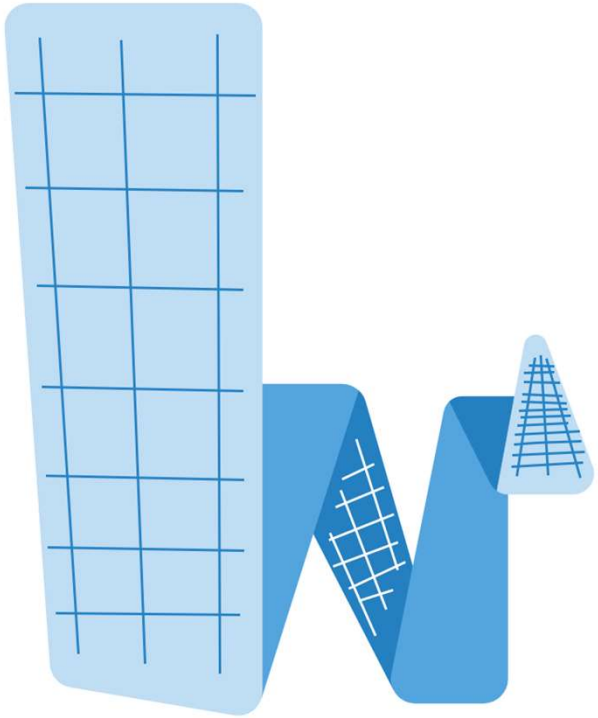
Data Relations

Sigma Complete Relations

Predicate	Natural Language statement e.g. “Cars registered in NSW”	Boolean expression (σ `SELECT / WHERE`) `ExGSTAmount = Price - GSTAmount AND GSTAmount = Price/11`
Start	Empty (just a Schema)	Complete (Full) and shaped by Predicate
Extent	Finite	Finite or infinite
Mutating	Yes: `INSERT, UPDATE, DELETE`	No (Constants)
Queries	Range over the rows that exist.	Range over all possible rows.
Obligations	Confidentiality, Integrity	No Information about real-world entities





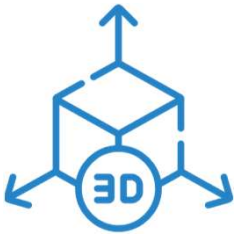


SQL-like form

```
COMPLETE(Price float, ExGSTAmount float, GSTAmount float)
```

Relational Algebra

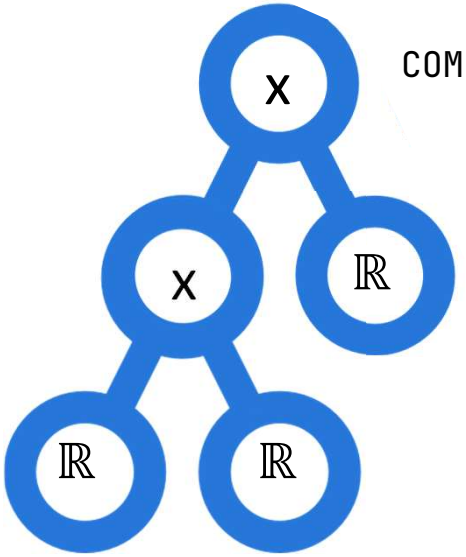
```
(Price float) X (ExGSTAmount float) X (GSTAmount float)
```



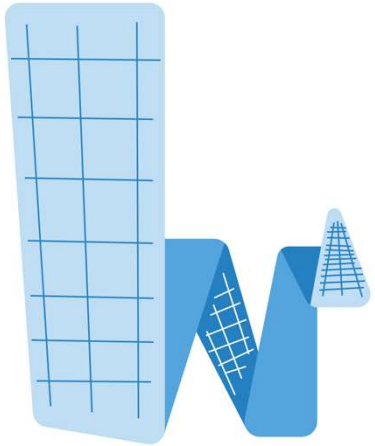
```
Australian_GST(Price float, ExGSTAmount float, GSTAmount float)
```

Australian_GST

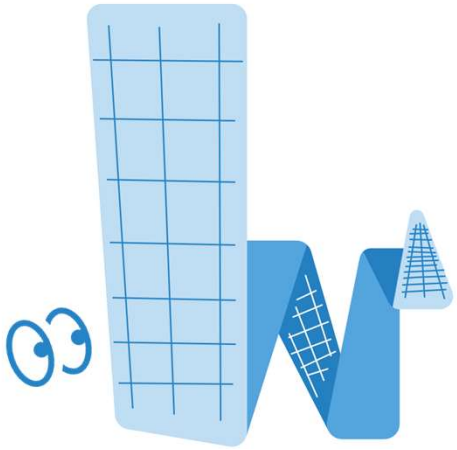
```
CREATE VIEW Australian_GST AS
SELECT *
FROM …
WHERE GSTAmount = Price/11 AND ExGSTAmount = Price-GSTAmount;
```

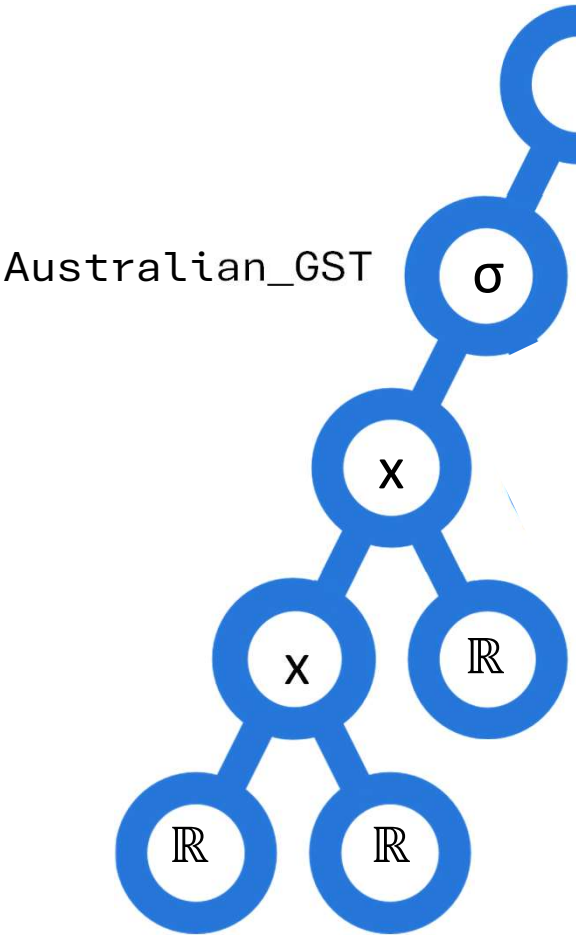



```
COMPLETE(Price float, ExGSTAmount float, GSTAmount float)
```



```
SELECT * FROM Australian_GST WHERE Price=110;
```





Price = 110 AND
GSTAmount = Price/11 AND
ExGSTAmount = Price-GSTAmount

GSTAmount = **110**/11 AND
ExGSTAmount = **110**-GSTAmount

GSTAmount = **10** AND
ExGSTAmount = 110-**10**

GSTAmount = 10 AND
ExGSTAmount = **100**

Am I Up to date with my COVID-19 Vaccinations?

Depending on:

- Age
- Number of doses you have had
- Your immuno-compromised status
- Months since your last dose

Do I need COVID-19 vaccinations for my workplace?

Depending on:

- State
- Work sector
- Place of work
- Employer

Delivering a personalised citizen experience using Rules as Code as a shared utility

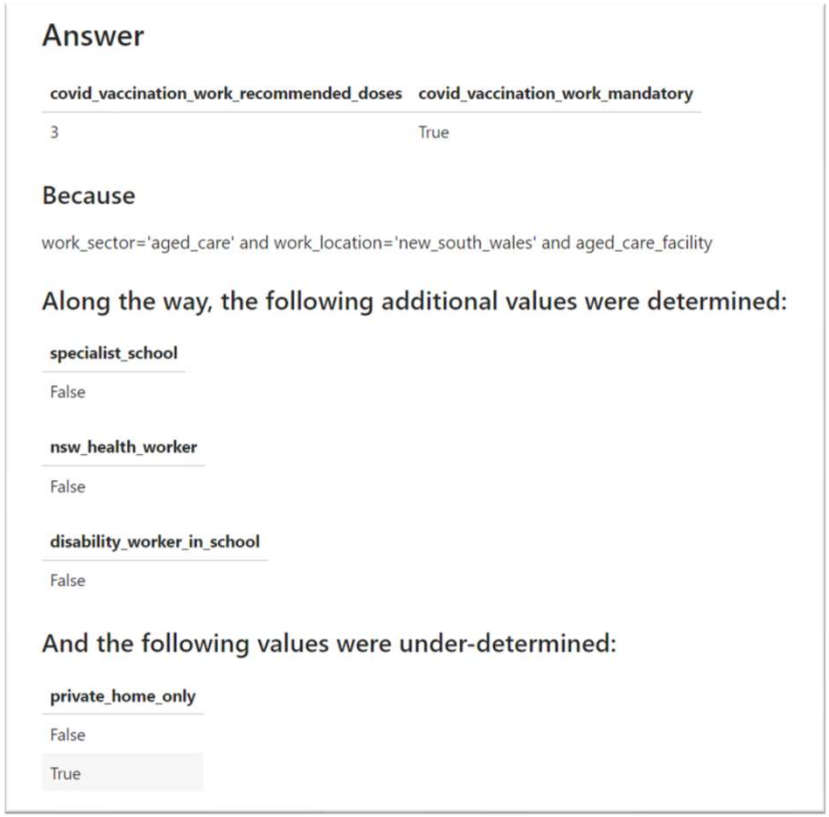
Results

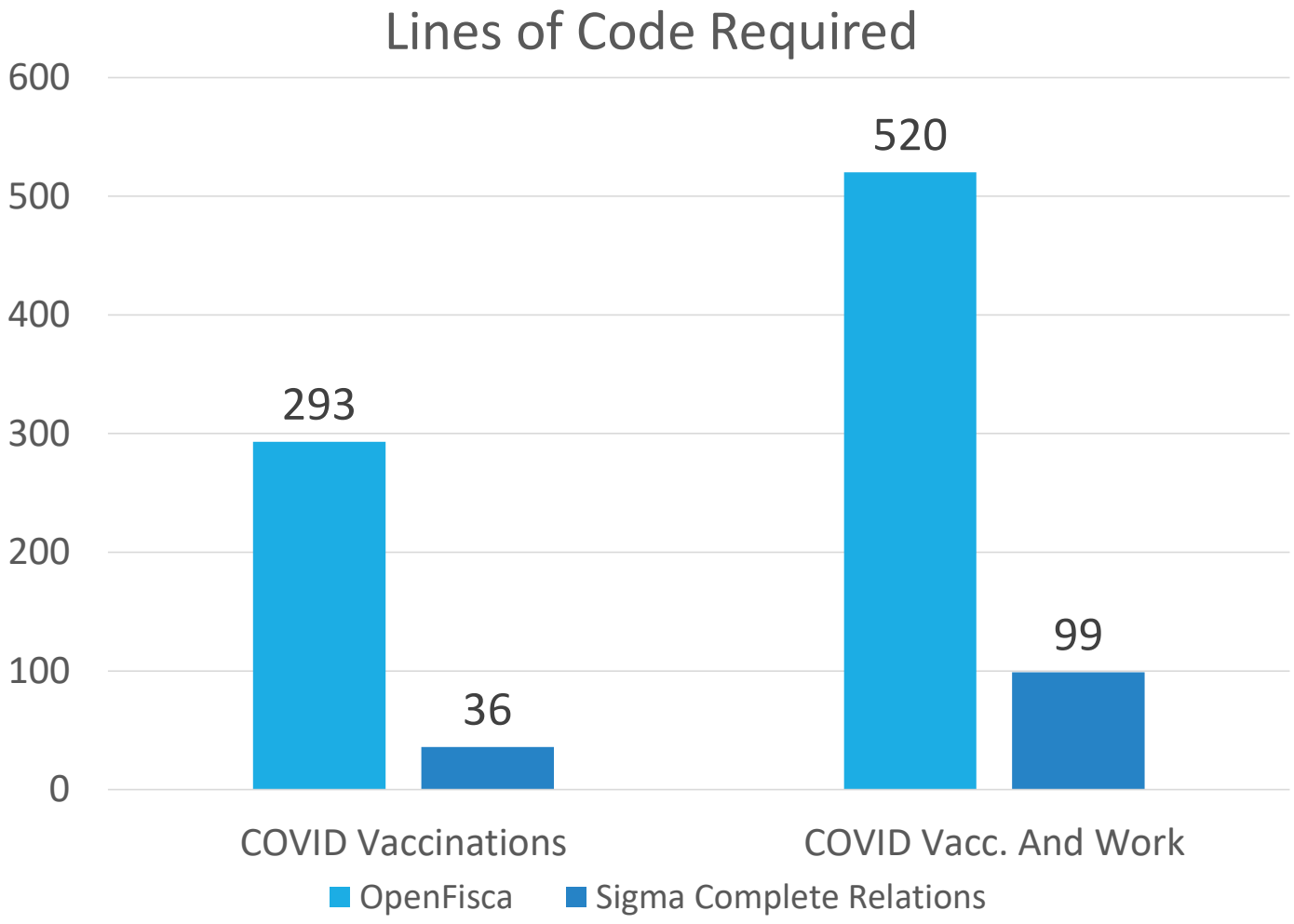
- Passed all 103 test cases of the Pilot

Key Benefits

- Automatic question sequencing based on the fastest path to an answer
- Supports querying in forward and reverse directions. E.g.

"In Western Australia, which roles require 3 vaccinations?"




Lines of Code Required
600
500
400
300
200
100
0
293
36
520
99
COVID Vaccinations
COVID Vacc. And Work
OpenFisca
Sigma Complete Relations

Morel (prototype)

MiniZinc, OPL

Satisfaction Modulo Theories

LP, MIP, Search

Boolean SATisfaction

Answer Set Programming

Constraint Database

Prolog III, and desc.

Verse

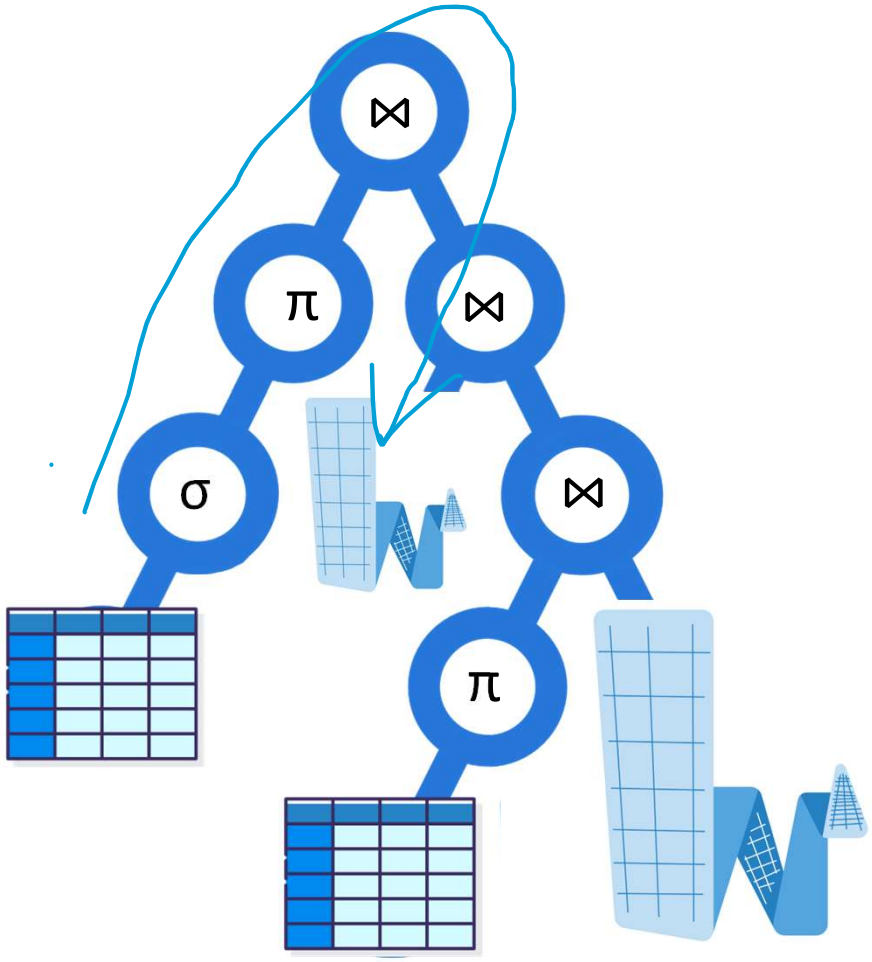
SQL, PRQL, Malloy

Datalog

Prolog

Functional Programming

- *hall 1975 an algebra relations machine computation*
- *maier 1981 incorporating computed relations relational databases*
- *jaffar 1987 constraint logic programming*
- *colmerauer 1989 an introduction to prolog iii*
- *hentenryck 1999 the opl optimization programming language*
- *nethercote 2007 minizinc standard cp modelling language*
- *revesz 2010 introduction databases biological spatio temporal*
- *kifer 2018 declarative logic programming systems applications*
- *arias 2018 constraint answer set programming without*
- *hyde 2022 morel standard ml interpreter relational*
- *augustsson 2023 verse calculus core calculus deterministic*

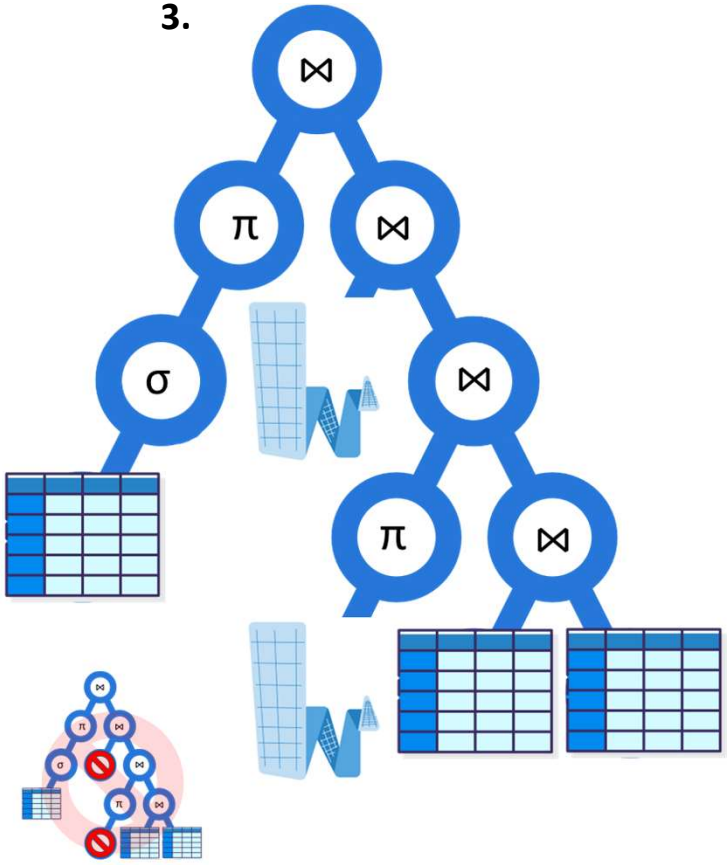

⋈
π
⋈
σ
⋈
π

1.

2.

```
SELECT * FROM Australian_GST
        WHERE Price=110;
              ExGSTAmount = 100;
              GSTAmount = 10;
```

3.




```
CREATE VIEW Australian_GST AS
SELECT *
FROM COMPLETE(Price, ExGSTAmount, GSTAmount)
WHERE GSTAmount = Price/11
    AND ExGSTAmount = Price-GSTAmount;
```

Thank you

Any Questions?