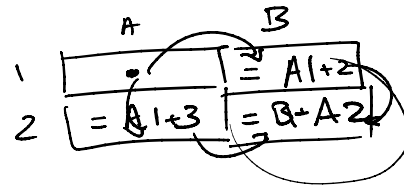
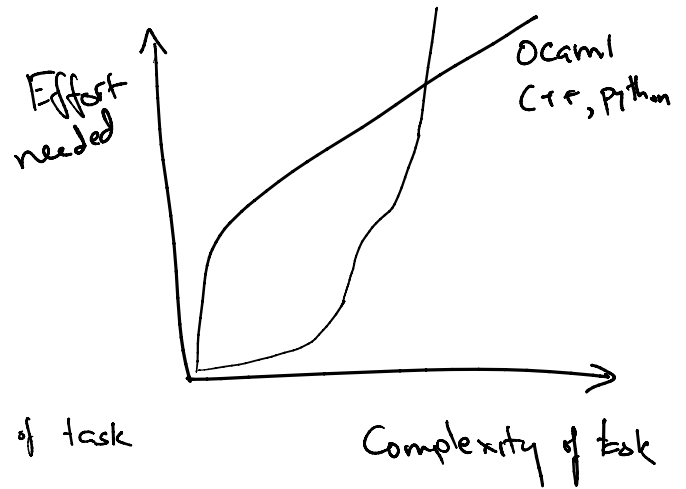
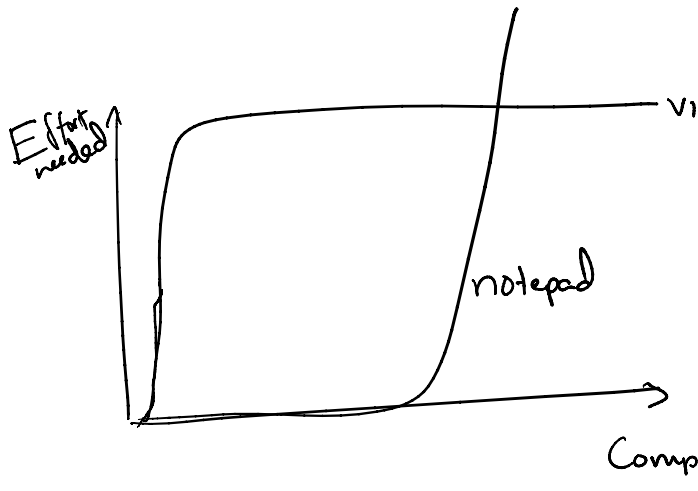
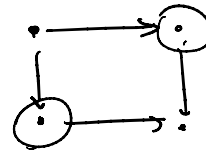


Execute a DFA inside a spreadsheet.

Pivot tables



E

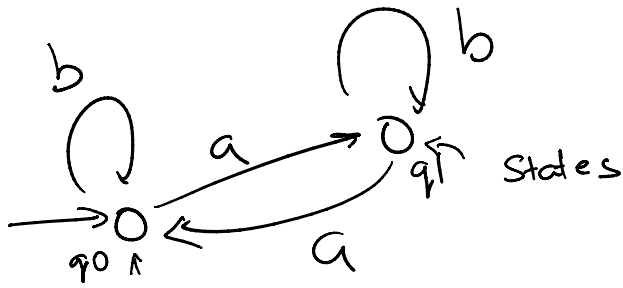


Relational algebra & SQL.

Graph Databases  
Datalog / Prolog.

Deterministic Finite Automata

# Deterministic Finite Automata



Constant time algorithms  $\equiv$  Lookup tables  
 $\equiv$  Automata.

"Follow the matrix"

Vlookup Hlookup

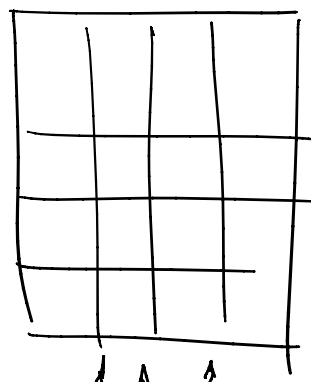
Match

Which artist has recorded the most albums?

SQL the language  $\Rightarrow$  Relational algebra

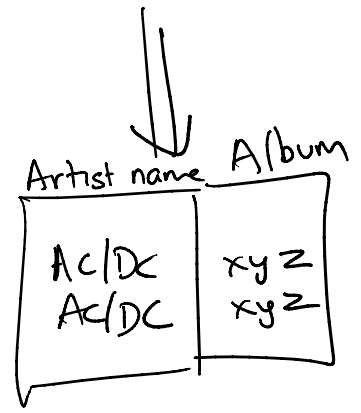
IBM DB2 / Oracle / Postgres

Table = Set of tuples.

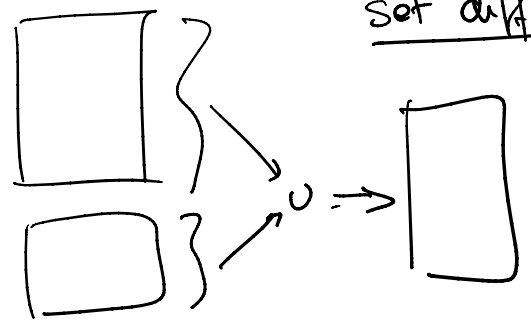


"order by" + no duplicates

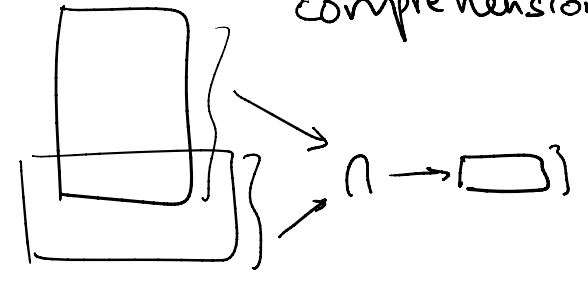
Rows / Tuple  
Records



Sets: unions



intersections



set difference

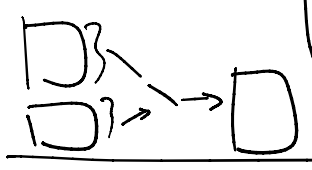
selections

"where" comprehension.

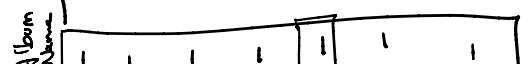
Album Name	Year
xyz	1998
abc	1932

xyz	1998
-----	------

All albums recorded after 1945.

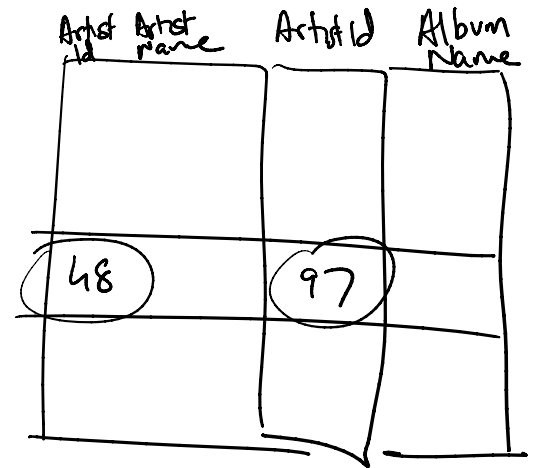
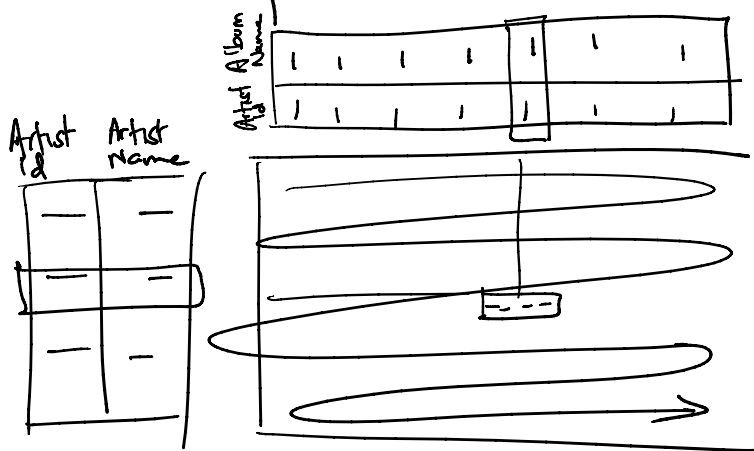


Cartesian products



Artist id Artist name Artistid Album Name

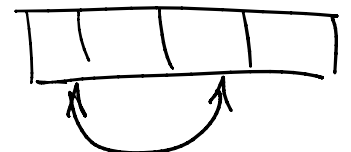
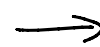
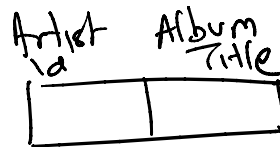
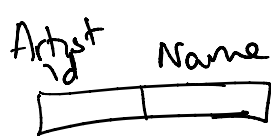
CARTESIAN



Cartesian Product ( $T_1, T_2$ )

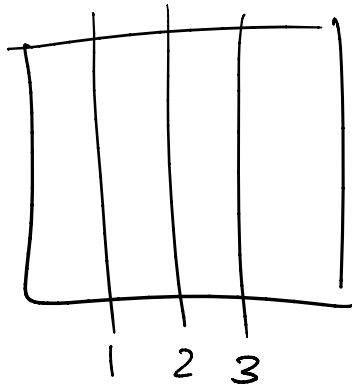
$$= \{ (t_1, t_2) \mid \text{for each } t_1 \in T_1, t_2 \in T_2 \}$$

Join: Take the Cartesian product & then find tuples which match.



Project (select in SQL)

# Project (select in SQL)



"Project the columns 1 & 3  
of the table."

Same as: "Delete column 2"

select col1, col3  
from \_\_\_\_\_

---

RA : Select	Project	Join / Cartesian product	Union	difference
SQL : where	Select	Join (various)	Union	<u>minus?</u>
			intersection	
			<u>intersection</u>	where. exists.