1.2. Data models

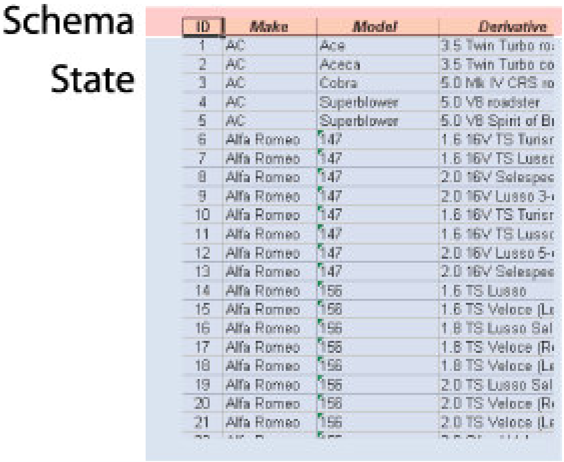
A data model is a model that describes in an abstract way how data are represented in a business organization, an information system or a database management system - Wikipedia.

1.2.01. Introduction

- Relational model
- Abstract operations on relations
  - Set theoretic operations
  - Relational-specific operations
- Basic algebra operations
  - Union, Intersection, Difference
  - Cross product

1.2.02. Relational model

- Table as relation
- Row as tuple
  - real world entity or relationship
  - fact
- Column as attribute
  - Domain



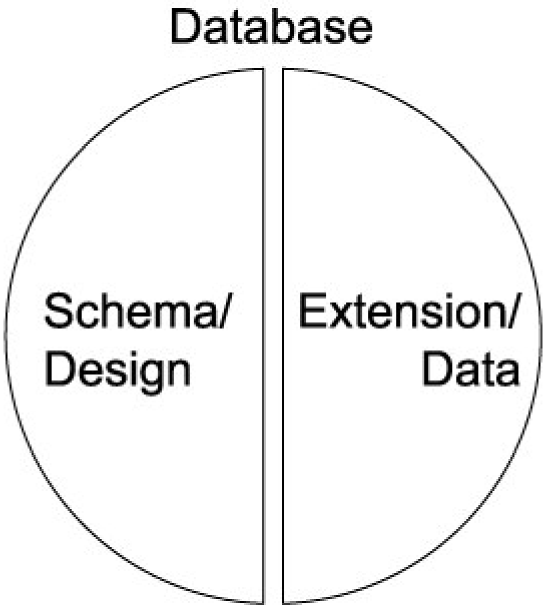
ID	Make	Model	Derivative
1	AC	Ace	3.5 Twin Turbo ro:
2	AC	Aceca	3.5 Twin Turbo co
3	AC	Cobra	5.0 Mk IV CRS ro
4	AC	Superblower	5.0 V8 roadster
5	AC	Superblower	5.0 V8 Spirit of Bi
6	Alfa Romeo	147	1.6 16V TS Turisr
7	Alfa Romeo	147	1.6 16V TS Lussc
8	Alfa Romeo	147	2.0 16V Selespec
9	Alfa Romeo	147	2.0 16V Lusso 3-(
10	Alfa Romeo	147	1.6 16V TS Turisr
11	Alfa Romeo	147	1.6 16V TS Lussc
12	Alfa Romeo	147	2.0 16V Lusso 5-(
13	Alfa Romeo	147	2.0 16V Selespec
14	Alfa Romeo	156	1.6 TS Lusso
15	Alfa Romeo	156	1.6 TS Veloce (Li
16	Alfa Romeo	156	1.8 TS Lusso Sal
17	Alfa Romeo	156	1.8 TS Veloce (R(
18	Alfa Romeo	156	1.8 TS Veloce (Li
19	Alfa Romeo	156	2.0 TS Lusso Sal
20	Alfa Romeo	156	2.0 TS Veloce (R(
21	Alfa Romeo	156	2.0 TS Veloce (Li

The concept of a relation is abstract, therefore we have a number of different ways of visualising it.

1.2.03. Relation

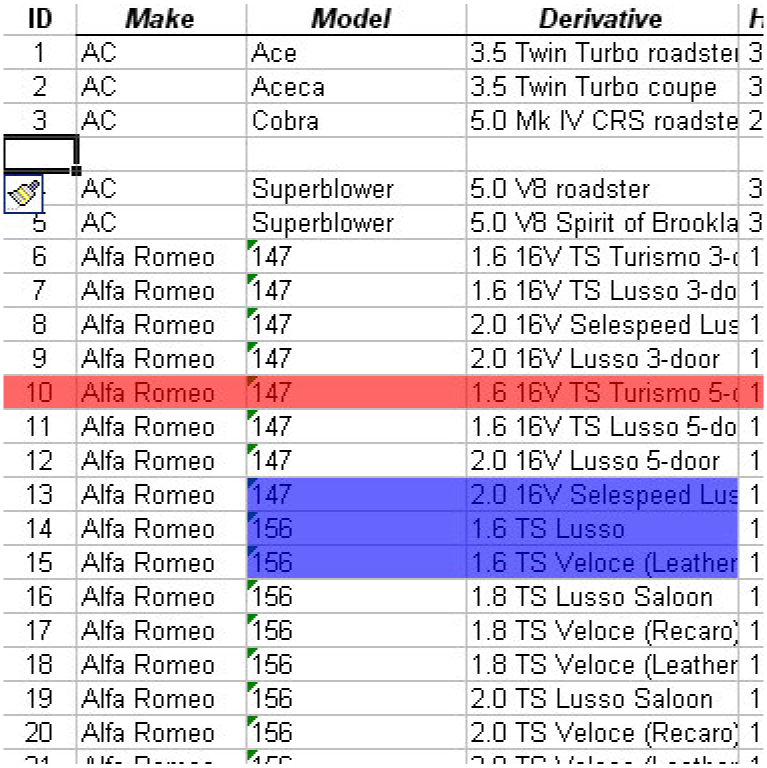
- Relation schema $R(A_1, A_2, A_3.. A_n)$

  - Design side
  - Assertion/declaration
- Relation state
  - Data side
  - set of n-tuples
    - each one an ordered list of values
      - 1NF: each value is atomic, no composite/multivalue



1.2.04. Abstract operations

- Database lifecycle
  - design, populate, evolve
- Insert
  - tuple $(a_1,a_2,a_3\ldots a_n)$
- Delete
  - tuple $(a_1,a_2,a_3\ldots a_n)$
- Update (or modify)
  - tuple $(a_1,a_2,a_3\ldots a_n)$
  - attribute to change, new value



ID	Make	Model	Derivative	H
1	AC	Ace	3.5 Twin Turbo roadster	3
2	AC	Aceca	3.5 Twin Turbo coupe	3
3	AC	Cobra	5.0 Mk IV CRS roadste	2
	AC	Superblower	5.0 V8 roadster	3
5	AC	Superblower	5.0 V8 Spirit of Brookla	3
6	Alfa Romeo	147	1.6 16V TS Turismo 3-(	1
7	Alfa Romeo	147	1.6 16V TS Lusso 3-do	1
8	Alfa Romeo	147	2.0 16V Selespeed Lus	1
9	Alfa Romeo	147	2.0 16V Lusso 3-door	1
10	Alfa Romeo	147	1.6 16V TS Turismo 5-(	1
11	Alfa Romeo	147	1.6 16V TS Lusso 5-do	1
12	Alfa Romeo	147	2.0 16V Lusso 5-door	1
13	Alfa Romeo	147	2.0 16V Selespeed Lus	1
14	Alfa Romeo	156	1.6 TS Lusso	1
15	Alfa Romeo	156	1.6 TS Veloce (Leather	1
16	Alfa Romeo	156	1.8 TS Lusso Saloon	1
17	Alfa Romeo	156	1.8 TS Veloce (Recaro)	1
18	Alfa Romeo	156	1.8 TS Veloce (Leather	1
19	Alfa Romeo	156	2.0 TS Lusso Saloon	1
20	Alfa Romeo	156	2.0 TS Veloce (Recaro)	1
21	Alfa Romeo	156	2.0 TS Veloce (Leather	1

All the operations described in the next few sections are abstract. We're going to see how valuable they can be in processing real world data later.

1.2.05. Basic algebra

- Two categories
    - Set theoretic operations
        - Union, Intersection etc.
    - Relational specific
        - Select, project and join

  At this stage we're talking about set theoretical operators on the Relational model, not SQL instructions which confusingly have identical names and only similar behaviour.

1.2.06. Select operation

- SELECT a subset of tuples from a relation
    - Uses selection condition
        - Evaluate each tuple to true of false
        - False tuples discarded
    - Sigma (s)
    - output = s(cond)(input_relation)
    - Relation schema: R(output) = R(input_relation)
    - Commutative

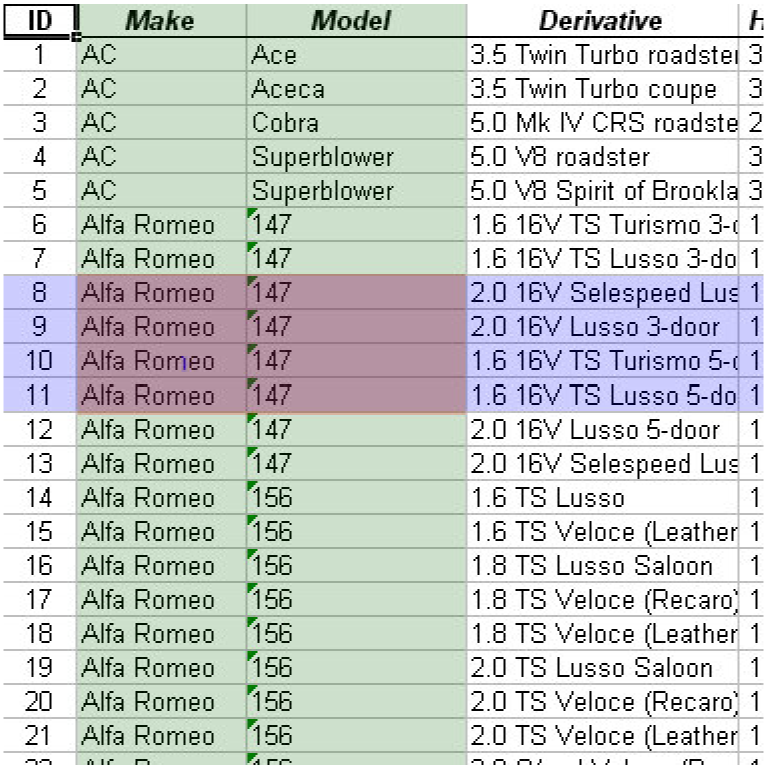
1.2.07. Project operation

- PROJECT a subset of attributes for all tuples from a relation
    - Pi (p)
    - $p_{\text{<attribute list>}}(R)$
- If sublist is only non-key attributes
    - might get duplicates
- Removes duplicates
- Attribute list:sublist example

The result set of the operation is itself a relational. That output relation will contain the same number of rows as the input, however it may contain a different number of columns; fewer if a subset of attributes is projected; more if derived or aggregated attributes are included.

1.2.07. Sequences of operations

- Select followed by projection
- Area clipping: rows then columns
- $p_{\text{<attr list>}}$
  $(s_{\text{(select\_cond)}}(R))$
- Rename operation (r)
    - Renames attributes list2 from list1
    - $r_{\text{(new\_attr\_names)}}(R)$



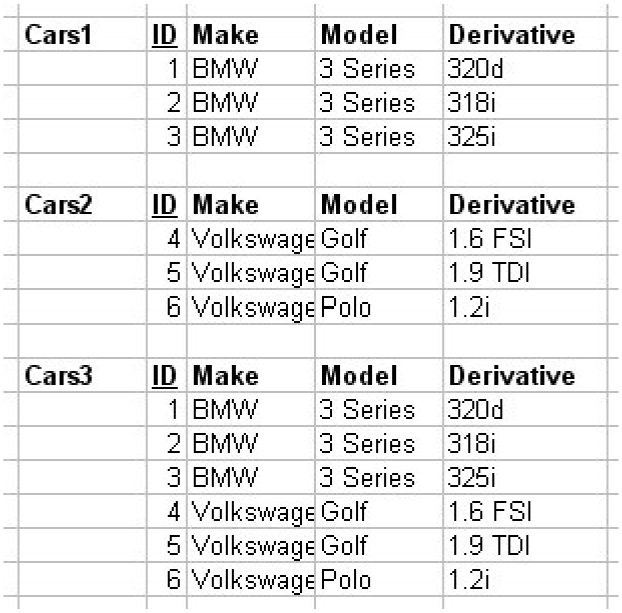
ID	Make	Model	Derivative	H
1	AC	Ace	3.5 Twin Turbo roadster	3
2	AC	Aceca	3.5 Twin Turbo coupe	3
3	AC	Cobra	5.0 Mk IV CRS roadste	2
4	AC	Superblower	5.0 V8 roadster	3
5	AC	Superblower	5.0 V8 Spirit of Brookla	3
6	Alfa Romeo	147	1.6 16V TS Turismo 3-(	1
7	Alfa Romeo	147	1.6 16V TS Lusso 3-do	1
8	Alfa Romeo	147	2.0 16V Selespeed Lus	1
9	Alfa Romeo	147	2.0 16V Lusso 3-door	1
10	Alfa Romeo	147	1.6 16V TS Turismo 5-(	1
11	Alfa Romeo	147	1.6 16V TS Lusso 5-do	1
12	Alfa Romeo	147	2.0 16V Lusso 5-door	1
13	Alfa Romeo	147	2.0 16V Selespeed Lus	1
14	Alfa Romeo	156	1.6 TS Lusso	1
15	Alfa Romeo	156	1.6 TS Veloce (Leather	1
16	Alfa Romeo	156	1.8 TS Lusso Saloon	1
17	Alfa Romeo	156	1.8 TS Veloce (Recaro)	1
18	Alfa Romeo	156	1.8 TS Veloce (Leather	1
19	Alfa Romeo	156	2.0 TS Lusso Saloon	1
20	Alfa Romeo	156	2.0 TS Veloce (Recaro)	1
21	Alfa Romeo	156	2.0 TS Veloce (Leather	1

1.2.08. Rename operation

- Attribute renaming only
    - Cannot alter domain, or add/remove attr
- Rename operation (r)
    - Renames attributes list2 from list1
    - r(new_attr_names)(R)
- Implicit renaming
    - Order dictated by relational schema

1.2.08. Set Theoretic

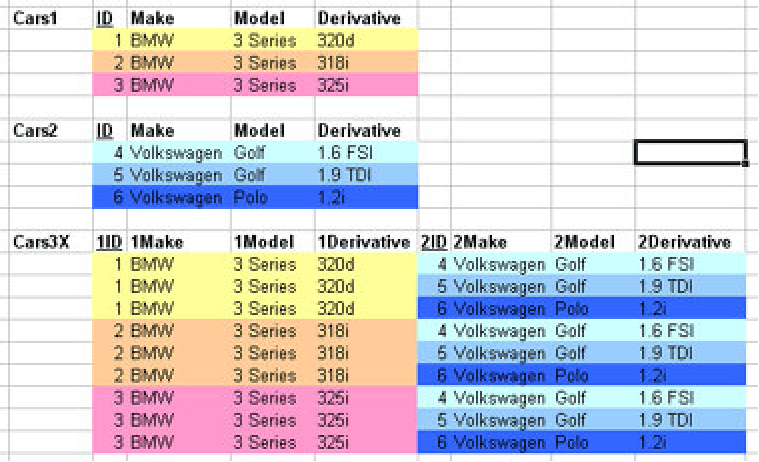
- Binary operation: two relations
    - Sets of tuples
    - Union compatibility (same attributes)
- Union (R u S)
- Intersection (R n S)
    - Commutative (R u (S u T) = (R u S) u T)
- Set difference
    - Non-commutative (R-S != S-R)



Cars1	**ID**	**Make**	**Model**	**Derivative**
	1	BMW	3 Series	320d
	2	BMW	3 Series	318i
	3	BMW	3 Series	325i
Cars2	**ID**	**Make**	**Model**	**Derivative**
	4	Volkswage	Golf	1.6 FSI
	5	Volkswage	Golf	1.9 TDI
	6	Volkswage	Polo	1.2i
Cars3	**ID**	**Make**	**Model**	**Derivative**
	1	BMW	3 Series	320d
	2	BMW	3 Series	318i
	3	BMW	3 Series	325i
	4	Volkswage	Golf	1.6 FSI
	5	Volkswage	Golf	1.9 TDI
	6	Volkswage	Polo	1.2i

1.2.09. Cross product

- Cartesian product of two relations
- R x S
- Also known as
    - Cross product
    - Cross join
        - Cross product diagram
    - Introduction to complexity
        - Computationally explosive



Cars1	ID	Make	Model	Derivative
	1	BMW	3 Series	320d
	2	BMW	3 Series	318i
	3	BMW	3 Series	325i

Cars2	ID	Make	Model	Derivative
	4	Volkswagen	Golf	1.6 FSI
	5	Volkswagen	Golf	1.9 TDI
	6	Volkswagen	Polo	1.2i

Cars3X	1ID	1Make	1Model	1Derivative	2ID	2Make	2Model	2Derivative
	1	BMW	3 Series	320d	4	Volkswagen	Golf	1.6 FSI
	1	BMW	3 Series	320d	5	Volkswagen	Golf	1.9 TDI
	1	BMW	3 Series	320d	6	Volkswagen	Polo	1.2i
	2	BMW	3 Series	318i	4	Volkswagen	Golf	1.6 FSI
	2	BMW	3 Series	318i	5	Volkswagen	Golf	1.9 TDI
	2	BMW	3 Series	318i	6	Volkswagen	Polo	1.2i
	3	BMW	3 Series	325i	4	Volkswagen	Golf	1.6 FSI
	3	BMW	3 Series	325i	5	Volkswagen	Golf	1.9 TDI
	3	BMW	3 Series	325i	6	Volkswagen	Polo	1.2i

1.2.10. Relational algebra/model notation

- Relational schema $R(A_1, A_2, \ldots, A_n)$
- Relation state r or $r(R)$
  - Set of unordered tuples
  - $r = \{t_1, t_2, \ldots, t_n\}$
- Each n-tuple is an ordered list of values
  - $t = <v_1, v_2, \ldots, v_n>$
- i^{th} value in $t = v_i$ called $t[A_i]$
- $r(R)$ subset of $(dom(A_1) \text{ x } dom(A_2) \ldots \text{ x } dom(A_n))$

1.2.11. Constraints

- Domain constraint
  - For all v in t of $r(R)$
    v_i is an element of $dom(A_i)$
- Entity constraint
  - $K = SK_{min}$
  - $t[K]$!= null
- Key constraint
  - Superkey SK as identifying subset of attributes
  - $t_1[SK]$!= $t_2[SK]$

1.2.12. Referential integrity

- Given two relations R_1 and R_2
  - R_1 contains a foreign key (FK) that references
  - A primary key (PK) in R_2
    - R_1 referencing relation, R_2 referenced relation
  - Shared domains: dom(FK) = dom(PK)
  - Foreign exists: t_1 in $r(R_1)$, t_2 in $r(R_2)$

- $t_1[FK] = t_2[PK] \parallel NULL$