

The Relational Thinker

Relational thinker, applied to enterprise cognition. In three parts.

Relational. The substrate, taken faithfully from Codd and Date.

Codd brought set theory to data: atomic domains, tuples, relations, normalization, predicate query, and a schema held apart from its instance. **Carlson BCM takes that same discipline to the capability layer**, construct for construct, the layer where business capability becomes the epicenter of enterprise architecture.

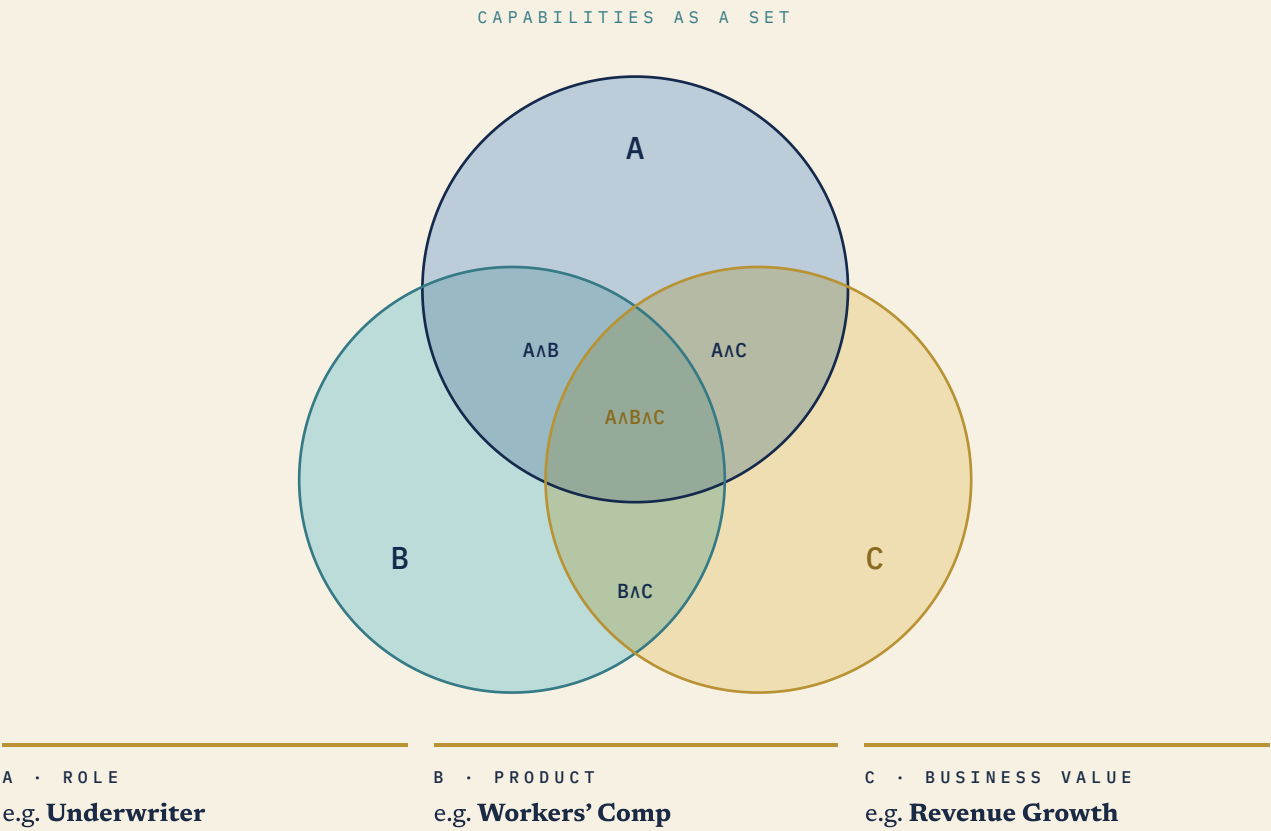
CONSTRUCT	FOR CONSTRUCT
DOMAIN · ATOMIC VALUE SET	Verb & object domain · the governed Verb Reference
TUPLE · ONE ROW, ONE FACT	Ability statement · “Ability to [verb] [object]”
RELATION · SET OF TUPLES, NO DUPES	Capability set · normalized, non-redundant
KEY · MINIMAL IDENTIFIER	Canonical identity · the canonical capability code
ATTRIBUTE · TYPED COLUMN	Enterprise Architecture attributes · strategic, operational, technology
NORMALIZATION · REMOVE REDUNDANCY	Common Functionals · authored once, referenced by code
Σ SELECTION · FILTER BY PREDICATE	Predicate selection · the Venn, evaluated per row
REF. INTEGRITY · FOREIGN KEY RESOLVES	Governed reference · every attribute resolves to a list

THE KEYSTONE

Schema over instance: the logical structure, set at design time, rarely changing. $\cong$

Model over realization: BCM is the enterprise schema; org, process, and platform are the instance a transformation must conform to.

Edgar Codd earned his doctorate at the University of Michigan, then went on to author the relational model. The author studied it at Yonsei University under a Berkeley-trained professor, and formalized its application to enterprise architecture at MIT.



Every attribute combination is a region. Any region is selectable by predicate.

Capabilities as a set, resolved by predicate logic.

The quintessential relational illustration, in capability terms.

The capability set carries enterprise architecture attributes. **A predicate selects the answer set**, and the answer can be any region or union of regions, not only the center. This is Codd's selection operator, made visible. The model scales across A...N attributes.

```
SELECT Capabilities
WHERE <predicate> // over A, B, C ... N
RETURN the matching Ability set
```

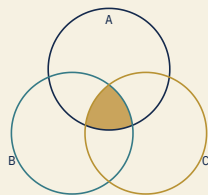
A Role · **Underwriter**
 B Product · **Workers' Compensation**
 C Business Value · **Revenue Growth**

∧ AND ∨ OR ¬ NOT () GROUP

THE CENTER

The center is not the answer. It is one of seven regions. Change the predicate and the answer moves: a single set, a lens, a crescent, a union, a complement.

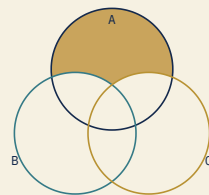
THE PREDICATE SELECTS THE ANSWER SET



$A \wedge B \wedge C$

All three hold

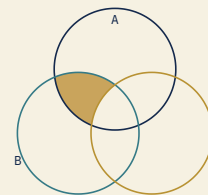
Underwriter · Workers' Comp
· Revenue Growth



$A \wedge \neg B \wedge \neg C$

Role alone

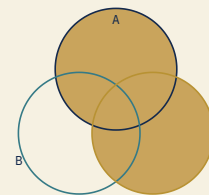
Underwriter, not the product,
not the value



$A \wedge B \wedge \neg C$

Role and Product, not Value

Underwriter on Workers' Comp,
not for growth



$A \vee C$

Role or Value (union)

any Underwriter, or any
growth driver

One set. Any region, on demand.

Applied to Enterprise Cognition.

Where the discipline is kept and the database is left behind.

The relational model is deliberately limited. It has no recursion and is not Turing complete; Codd traded power for provable simplicity. **Enterprise transformation is made of exactly what he excluded.** Two things carry BCM across that line: a set of rules that make it impossible to fake, and a reach past the algebra into cognition.

FIRST, THE RULES THAT PREVENT THE FAKE

Codd's defense

When vendors bolted a query layer onto older engines and called it relational, Codd answered with **twelve rules**: a pass or fail test of authenticity a veneer could not satisfy. The rules, not the vocabulary, were what made imitation impossible. Defending them cost him his standing at IBM.

The Carlson BCM test · eight authored principles

- | | |
|----|---|
| 01 | Render strategic intent as capabilities |
| 02 | Apply relational technique to ensure non-redundancy |
| 03 | Model in a structured hierarchy |
| 04 | Model needs with common capabilities |
| 05 | Attribute for enterprise architecture impact and traceability |
| 06 | Treat capability models as durable, reusable assets |
| 07 | Enable enterprise alignment and collaboration on capability |
| 08 | Ensure relevance across the investment lifecycle |

Relational algebra cannot express recursion. It cannot follow a chain of dependencies to its end. That boundary is exactly the work of transformation.

Capability depends on capability; a change ripples through a dependency closure. Architectures are traversed as graphs. A new target state invalidates the prior one, the non-monotone case Codd needed his difference operator for. Judgment, scoring, and inference sit above query entirely. BCM keeps the relational discipline as its substrate and operates in the space the database draws as its own outer edge. That space is cognition.

Relational

the substrate, kept faithfully: sets, normalization, predicate, schema over instance

Thinker

authored by design, the one act the algebra cannot perform for itself

Enterprise

the capability layer, the epicenter of the architecture

Cognition

recursion, traversal, inference: the territory above the ceiling

Relational by discipline. Beyond relational by design.