Aggregation:

(Class, Instance, Attribute, Value);

* Class (Employee, Married) : Table
* Instance (anEmployee, aMarried) : PK (row)
* Attribute (Employer) : Column
* Value (anEmployer) : Value

Activation:

(Metaclass, Context, Role, Occurrence);

* Metaclass (Employment, Marriage)
* Context (anEmployment, aMarriage)
* Role (Employee, Employer, Wife, Husband)
* Occurrence (anEmployee, anEmployer, aWife, aHusband)

Mappings (Alignment):

(Dimension, Measure, Unit, Value);

* Dimension (Age, Distance)
* Measure (DoB, elapsedTime)
* Unit (Years, Km/h)
* Value (anAgeYears, aDistanceKmts)

Augmentations:

* Aggregation: sort / train. Clustering. N-ary relationships (marriages, employments). FCA: P(S, O) Contexts.
* Activation: predict. Classification. Age / DoB, married / marriage (wife, husband, dateMarried). Sets (CSPO, Kinds, Statements).
* Alignment: complete / map. Regression. State (single / married). Dimensions: compose / translate. Sort: current Value, next Measure. Tag: Classifications, Clusterings.
* Merge Graphs: Ontology Matching. Dimensional Alignment.

Image Recognition:

* Context (Class, Metaclass, Dimension): Class
* Subject (Instance, Context, Measure): Blob
* Predicate (Attribute, Role, Unit): x1, y1, x2, y2, props / tags
* Object (Value, Occurrence, Value): x1, y1, x2, y2, props / tags
* Context: Subject Classification Class (Person).
* Subject (train / predict): example.org?aPerson&data=Base64Data.
* Attributes / Values: Metadata.

Train / Predict / Matching:

Aggregation, Activation, Alignment: Statements, Kinds, CSPO (URNs).

CSPOs: URNs, Reified Kinds, Reified Statements.

Kinds. Inferred SPO Types (Sets):

SubjectKind: S(P, O)\*. Aggregated Subject Predicates / Objects.

PredicateKind: P(S, O)\*. Aggregated Predicate Subjects / Objects.

ObjectKind: O(P, S)\*. Aggregated Object Predicates / Subjects.

Augmentation Statements:

Aggregation / Reification:

Statements / Kinds Reified as URNs.

Aggregation:

* Context (Class): SubjectKind. Employer
* Subject (Instance): URN. anEmployer
* Predicate (Attribute): URN. Employee
* Object (Value): URN. anEmployee
* Context (Class): SubjectKind. Employee
* Subject (Instance): URN. anEmployee
* Predicate (Attribute): URN. Employer
* Object (Value): URN. anEmployer
* Context (Class): SubjectKind. Position
* Subject (Instance): URN. aPosition
* Predicate (Attribute): URN. Employee
* Object (Value): URN. anEmployee
* Context (Class): SubjectKind. Employee
* Subject (Instance): URN. anEmployee
* Predicate (Attribute): URN. Position
* Object (Value): URN. aPosition

Activation:

* Context (Metaclass): SubjectKind. Employment
* Subject (Context): AggregatedReifiedAggregation. anEmployer, anEmployee, aPosition
* Predicate (Role): AggregationSubjectKind. Employer / Employee / Position
* Object (Occurrence): AggregatedReifiedAggregation. anEmployer, anEmployee, aPosition

Alignment:

* Context (Dimension): SubjectKind. LaboralState. MaritalStatus.
* Subject (Measure): AggregatedReifiedActivation. aLaboralState : S(P, O) Kinds Layered Predicates Matching (Employee::dateHired from Employment PK). aMaritalStatus (dateMarried from Marriage PK).
* Predicate (Unit): ActivationSubjectKind. Employment, Marriage
* Object (Value): AggregatedReifiedActivation. hired, unemployed, junior, senior, single, married, wife, husband.

Aggregated Reified Statements:

(SubjectKind, Subject)

Predicates. Subject Kind P(S, O). Layered Pattern Matching:

Employment[PK: laboralState, senior; maritalStatus, single] : Employments, Marriages. Activation SKs.

Employer[PK: anEmployer, anEmployee] : Employers. Aggregation SKs.

Employee[PK: anEmployee, aPosition] : Employees. Aggregation SKs.

Position[PK: aPosition, anEmploye] : Positions. Aggregation SKs.

Functional Composition:

C: Transform Context. Mapping State.

P: Transforms, Mappings.

S, O: Monads, Functors.

S.flatMap(P) : O;

O.flatMap(P) : S;

(anEmployer(Employee) : anEmployee)(Position) : aPosition