IDEF1X Model Anatomy

Data Model [Item Event]

This explains an IDEF1X Data Model in terms of its anatomy, it is suited to technically competent readers. For a more formal introduction refer to the

IDEF1X Introduction

Relational Model

- Based on First Order Predicate Calculus,
  - therefore it is Logical, the diagram is semantic
  - therefore Predicates are fundamental
- Rows, the Logical data (not records) are **Identified** by Logical Key
- (In pre-relational Record Filing Systems physical records are related by Record ID)

Key

- Logical row (data) Identifier
- Must be *made up from the data*
- Rows must be unique
- Compound Keys are ordinary Relational fare
- Logical Keys have meaning, provide context

Primary Key

- Primary Identifier, above the line
- Atrributes (non-PK) are below
- Migrated to child as FK

Alternate Key AK

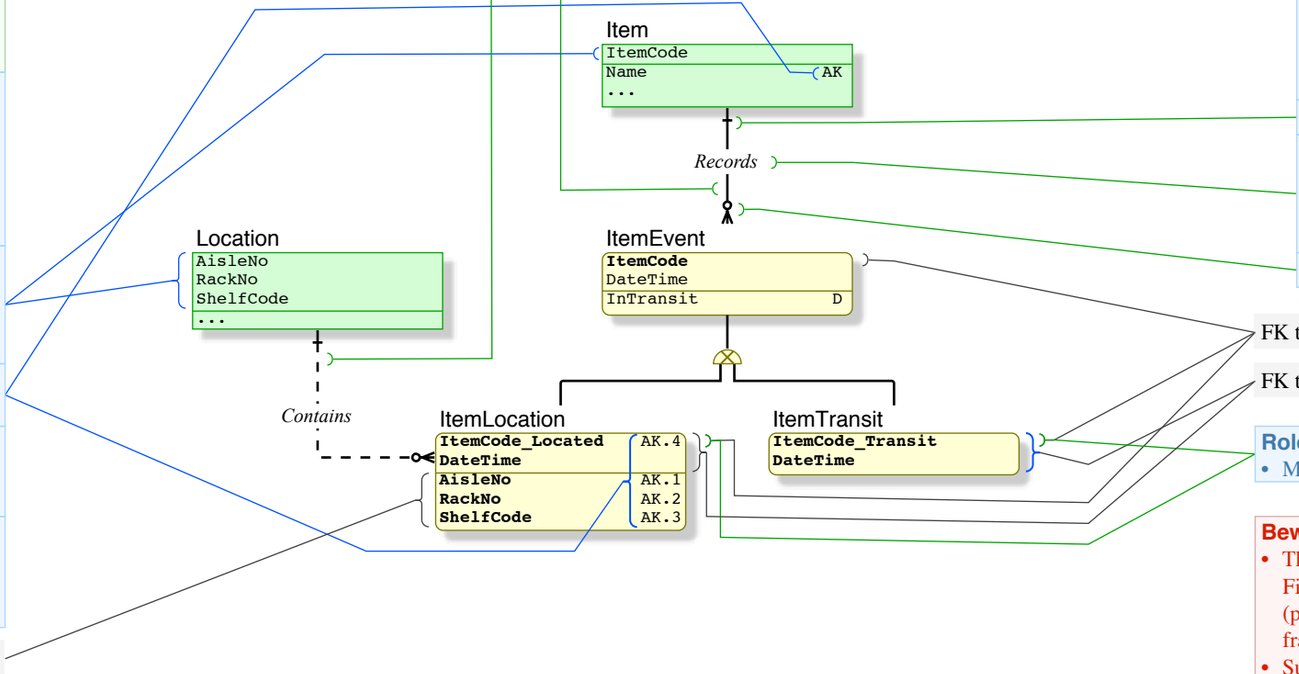
- Additional row Identifier

Independent Square corners

- Row is Independent
- It exists without dependence on any other row

Dependent Round corners

- Dependency means
- Row is Dependent on an Identifying parent row
- It exists only in the context of the parent row



Relation[ship]

Identifying Solid line

- Parent row identifies the child row
- Parent PK (migrated FK) forms the child PK

Non-identifying Dashed line

- Parent row does not identify the child row

Foreign Key

- Parent PK is an FK in the child
- Bold means a migrated data element
- Rows, the Logical data (not records) are **related** by Logical Key
- Meaning is carried in the migrated Keys

- Cardinality at parent (always 1)

Verb Phrase

- Predicate for Relation
- Action between subject and object
- Name for FK Constraint

- Cardinality at child

RoleName

- Meaningful (contextual) name for FK

Predicate

The constraints placed on the Data. All Predicates can be *read* from the model. Stated explicitly for users & newbies:

- Row **Existence** Dependent/Independent Identifyin/Non-Identifying
- Row **Identification** Primary/Alternate Key
- **Relations** between rows Verb Phrase (determine and *read* reverse VerbPhrase as well)
- **Descriptors** (Attributes that describe the Primary Key) are below PK line

Predicate (Relevant, not all)

```
Item is independent
    Item is primarily   identified by ( ItemCode )
    Item is alternately identified by ( Name )
    Item recorded 0-to-n ItemEvents

ItemEvent is dependent on, and identified by, Item
    ItemEvent is a recording of 1 Item
    ItemEvent is identified by ( ItemCode, DateTime )
    ItemEvent is an exclusive basetype
    ItemEvent is one of { Location | InTransit }

ItemLocation is a subtype of ItemEvent
    ItemLocation is 1 ItemEvent
    ItemLocation is contained in 1 Location
    ItemLocation is identified by ( ItemCode, DateTime )

ItemTransit is a subtype of ItemEvent
    ItemTransit is 1 ItemEvent
    ItemTransit is identified by ( ItemCode, DateTime )

Location is independent
    Location is identified by ( AisleNo, RackNo,
    ShelfCode )
    Location contained o-to-n ItemLocations
```

SQL

```
CREATE TABLE ItemLocation (
    ItemCode_Located    ItemCode    NOT NULL,
    DateTime            _DateTime   NOT NULL,
    AisleNo             AisleNo     NOT NULL,
    RackNo              RackNo      NOT NULL,
    ShelfCode           ShelfCode   NOT NULL
    CONSTRAINT PK       -- Subtype
        PRIMARY KEY CLUSTERED ( ItemCode_Located, DateTime )
    CONSTRAINT AK       -- Locates Item
        UNIQUE ( AisleNo, RackNo, ShelfCode, ItemCode_Located )
    CONSTRAINT ItemEvent_Is_ItemLocation_fk
        FOREIGN KEY           ( ItemCode_Located, DateTime )
        REFERENCES ItemEvent ( ItemCode_Located, DateTime )
    CONSTRAINT Location_Contains_ItemLocation_fk
        FOREIGN KEY          ( AisleNo, RackNo, ShelfCode )
        REFERENCES Location ( AisleNo, RackNo, ShelfCode )
    )
```

Beware

- The "theoreticians" promote pre-relational Record Filing Systems, characterised by Record IDs (physical, not row identifiers) as "keys", and fraudulently claim such to be "relational".
- Such physicalised filing systems (stored in a database container, accessed via SQL for convenience), have **none of the Integrity; Power; or Speed** of Relational databases.
- Terms such as "candidate key" are anti-relation, used to misrepresent non-relational RFS as "relational", and to hide the failure to implement a logical Primary Key.
- Likewsie, a "surrogate key" is a physical record (not row) identfier, always an additional column and index.

Relational Db vs Record Filing System

The *identifying* (sorry) mark of a pre-relational Record Filing System is this:

- All files have a Record ID as "key"
- All files are Independent
- All relations are Non-Identifying