A “Relational Green Card”
Supporting Data Modeling in IS 2002
an ontology of the relational paradigm

Les Waguespack, Ph.D.
Computer Information Systems
Bentley University
Waltham, MA  01254-4705
## Database Content Required in IS2002

<table>
<thead>
<tr>
<th>C#</th>
<th>Course Title</th>
<th>PR</th>
<th>LU</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>Personal Productivity With IS Technology</td>
<td></td>
<td>3/21</td>
</tr>
<tr>
<td>1</td>
<td>Fundamentals of Information Systems</td>
<td></td>
<td>P0</td>
</tr>
<tr>
<td>2</td>
<td>Electronic Business Strategy, Architecture and Design</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Information Systems Theory and Practice</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Information Technology Hardware and System Software</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Programming, Data, File and Object Structures</td>
<td>1</td>
<td>2/20</td>
</tr>
<tr>
<td>6</td>
<td>Networks and Telecommunications</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Analysis &amp; Logical Design</td>
<td>1</td>
<td>1/14</td>
</tr>
<tr>
<td>8</td>
<td>Physical Design and Implementation With DBMS</td>
<td>5, 7</td>
<td>7/16</td>
</tr>
<tr>
<td>9</td>
<td>Physical Design and Implementation in Emerging Environments</td>
<td>2, 8</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Project Management and Practice</td>
<td>7</td>
<td>1/11</td>
</tr>
</tbody>
</table>

14/82 LU's (> 1/6) include DB concepts
The Relational Paradigm

The Relational Ontology

This ontology is consistent with the practice in computer science and information science categorizing a domain of concepts (i.e. individuals, attributes, classes and relationships). This ontology of the relational paradigm of data modeling minimizes the vestiges of implementation languages and methodologies to expose the core nature of relational concepts.

1. Individuals

The most concrete concept in the relational paradigm is the tuple.

1.1. Tuple

A tuple corresponds 1-1 with a single concept of reality that it represents. A tuple collects the facts that identify it as a single concept and the facts most closely identified with it.

2. Attributes

Attributes are those characteristics (facts) that describe a tuple. In the relational paradigm attributes define data characteristics - each of which has a static and dynamic form. A prescribed set of attributes defines what is called the structure of a tuple. From inception to extinction the structure of a tuple is immutable. The number of attributes in a tuple is called its degree.

2.1. Data Attribute

Data attributes store information (data) in the tuple and implement the property of remembrance. Remembrance is a mathematical antecedent where attribute values within each tuple reflect a correspondence with the coincidence of facts in the "real world," a correspondence (attribute relationship) that is shared by every tuple in that relation.

3. Relation

The relation concept combines both a definition of structure and the collection of tuple(s) based on that structure. A relation is defined as a fixed set of data attributes. Every tuple is an instance of a specific relation and shares the same static structure defined by that relation with every other tuple of that relation. The relation concept purely exists the existence of the tuples to that of their relation; tuples cannot exist independent of their defining relation. Tuples are said to be members of their relation. Tuples are added to or deleted from their relation. The order of attributes in a relation is insignificant except that the order is consistent for all tuples. A relation is also commonly called a table and each of its instances, a row. The collection of every data attribute value(s) for a particular data attribute in a table is called a column.

4. Relationships

Relationships in the relational paradigm are based on the property of remembrance and the juxtaposition of data attribute values in one or more tuples in the same or across relations.

4.1. Behavioral Relationships

The behavioral relationships are all based upon the data attribute value(s) and which values are permitted to coexist in and across tuples and relations.

4.1.1. Functional Dependency

In a relation a data attribute is functionally dependent upon another attribute when its data value is always the same in any tuple for a given value in a second data attribute. In other words, the value of the first data attribute is determined by the value of the second (called the determinant). Functional dependency expresses the informational integrity of relations.

4.1.1.1. Entity Integrity

Entity integrity defines the two-fold quality of tuple uniqueness in a relation: a) every tuple in a relation is distinct in some data attribute

value(s) from every other tuple in that relation or symmetrically, b) there is a designated subset of data attributes (column(s)) called the primary key such that the data attribute value(s) in that relation is distinct for all tuples and no values may be null (a value which is unknown and incomparable to any other value). There may be more than one subset of data attributes with the value characteristics of the primary key (each called a candidate key) but only one is designated as the primary key.

4.1.2. Association

An association is a relationship between tuples in the same or different relations. Tuples are intrinsically separable by way of entity integrity. At the same time, humans are compelled to categorize their experience of things in the physical world by superimposing groupings that collect tuples into sets. Tuples become members in a group based upon data attribute value(s). This property is called membership IV. This property also permits humans to identify a tuple that is not in a set (i.e. discrimination). (Membership IV an association is distinct from membership OF a relation which is intrinsic by way of instance relationship.)

4.1.2.1. Relational Operations

Membership IV is realized through relational operations keying on relation structure and values. Each relational operation produces a real or virtual relation as its result. The selection operation retrieves tuple(s) based upon a selection predicate testing data attribute value(s) to determine whether each tuple is or is not in the set. Selection predicates are based on any boolean comparison including constant values or values referenced in data attribute value(s). The projection operation copies all the data attribute value(s) for a particular column(s).

Association between relations (or a relation and itself) is based upon relating (matching) data attribute values in tuples of one relation with those of another. The join operation pairs every combination of tuples from one relation with those of another relation and copies the data attribute values from the pairs where the pairing satisfies a selection predicate. This relational operation is called join because facts from two sources are joined in the result.

4.1.2.2. Join Compatibility

Join compatibility requires that the values involved in comparisons (i.e. selection predicates) whether constants or data attribute values derive from the same data attribute domain.

4.1.2.3. Referential Integrity

When relations are devised such that a tuple in one relation predetermines the existence of (own) tuple(s) in another, the data attribute(s) of the second required to join the relations is called a foreign key. Referential integrity asserts that any value found in the data value attribute(s) of a foreign key must appear in a tuple of the first relation as the value of a candidate key or itself be null.

4.1.3. Normalization

Relational model consistency depends on the semantic concurrence of the behavioral relationships and the objectives of the database modeler, the extension, (rather than the accident of a relation's contents at any particular instant, its extension). The integrity properties defined above enable the database modeler to devise a structure and behavior of relations that avoids semantic discord called anomalies, the unintended loss or modification of information by relational operations. Relations designed to avoid certain kinds of anomalies are said to be normalized or in normal form. Normalization is the arrangement of data attributes and their relationships among relation structures to prevent particular anomalies.

4.1.3.1. First Normal Form

First Normal Form asserts that every data attribute value is atomic, indivisible in value and form and may not be operated upon except as a whole and single value.

4.1.3.2. Second Normal Form

Second Normal Form is first normal form and asserts that every data attribute value not in the primary key is fully functionally dependent upon the primary key. ("Fully" means applying to every data attribute of the primary key.)

4.1.3.3. Third Normal Form

Third Normal Form presupposes first and second normal forms and asserts that no data attribute outside the primary key is transitively dependent upon the primary key. ("Transitively" means an attribute(s) functionally dependent upon an attribute functionally dependent upon an attribute ( . . . ) functionally dependent upon the primary key.)
What is an ontology?

* ontology: the branch of metaphysics dealing with the nature of being [New Oxford Dictionary]

* metaphysics: the branch of philosophy that deals with the first principles of things, including abstract concepts such as being, knowing, substance, cause, identity, time, and space.

* “what exists, how do we understand it, what explains it, what does it explain?”
What does an ontology do for us?

- It helps us describe the “world!”
  - a common terminology shared by the community
  - shared rationale explaining properties
- What questions does an ontology answer?
  - what are the things? - “individuals”
  - how are they described? - “attributes”
  - what things go together? - “classes”
  - how do things relate to one another? - “relationships”
Relational Ontology - Graphically

- relation
- instance relationship
- membership OF property
- tuple
  - value data attribute
  - atomicity property
  - remembrance property
  - association relationship
- domain data attribute
  - data attribute
- membership IN property
- functional dependency property
- entity integrity property
- referential integrity property
- membership IN relationship
- remembrance property
- atomicity property
- value
  - domain
  - membership IN
  - functional dependency
  - referential integrity
  - entity integrity
  - atomicity
  - value
  - remembrance
  - association relationship
  - instance relationship
  - membership OF property
Once over quickly!

- tuple
- relation
- domain
- data attribute
- membership IN
- property
- functional dependency
- property
- instance
- relationship
- membership OF
- property
- referential integrity
- property
- value
- data attribute
- entity integrity
- property
- atomicity
- property
- remembrance
- property
- membership IN
- property
- association
- relationship
- atomicity
- property
What things are Relational?

* Relational Paradigm: Individual - tuple
  - derived from the living physical experience of humans seeing and touching things - projected onto non-concrete abstractions as well

* tuples are distinct
  - they are separable by nature of how they are described,
  - they are distinguishable because of their attribute values,

* tuples are sometimes called rows

* the set of all data attribute values in a single attribute for all tuples in a relation is sometimes called a column
What describes relational things?

- Relational Paradigm: Attributes - Attributes
- Relational "things" are described by their attributes
- Data attribute values are atomic - indivisible and whole

<table>
<thead>
<tr>
<th>attributes</th>
<th>data</th>
<th>behavioral</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>data attributes domains define what &quot;can&quot; be stored and recalled: the property of remembrance</td>
<td>the functional dependency of one attribute value on another expresses intension</td>
</tr>
<tr>
<td>dynamic</td>
<td>data attributes values possess what &quot;is&quot; stored and recalled: the property of remembrance</td>
<td>adherence to intension in the actual values in tuples (extension) realizes integrity</td>
</tr>
</tbody>
</table>
What things go together?

* Relational Paradigm: Classification - relation

* The relationship instance of defines a collection of tuples individually expressing specialized instances of the same concept (A relations is sometimes called a table)

* Every tuple is an instance of its relation and shares the same static structure defined by that relation with every other tuple of that relation

* Tuples are said to be “members of their relation.”

* Relation structure - data

* Static data attributes are defined in the relation’s attribute domains
How do things relate?

Relational Paradigm: Relationships - (structural and behavioral)

- every tuple in a relation shares the identical structure of attributes and domains
- although the ordering of attributes is immaterial in a relation their ordering among all tuples is consistent
- integrity properties defined in the relation apply to all tuples of the relation
How do things relate? (cont.)

- Relational Paradigm: Relationships (continued)
  - behavioral- (functional dependency, entity integrity, association, referential integrity)
    - functional dependency relates attributes - prescribes that the value of one attribute(s) will determine the value of another(s) in a tuple for all tuples in a relation.
    - entity integrity stipulates that exists a subset of the attribute(s) in a relation whose value(s) uniquely discriminates each tuple within a relation and that no attribute(s) in this subset may be null (of indeterminate value). A subset of attributes of this type is called a candidate key one of which is designated as the primary key!
How do things relate? (cont..)

* Relational Paradigm: Relationships (continued)

* behavioral- (functional dependency, entity integrity, association, referential integrity)

* association relates tuples across two relations (or a relation with itself) - data attribute values in attributes of the two relations from the same domain (join compatibility) define a set of tuples in one relation as related to one tuple in the other - the property of membership in the association (membership in a group is distinct from member of a relation)
association (continued)

Relational Operations - Membership IN is realized through relational operations keying on relation structure and values. Each relational operation produces a real or virtual relation as its result. The selection operation retrieves tuple(s) (also called row(s)) based upon a selection predicate testing data attribute value(s) to determine whether each tuple pair is or is not in the set. Selection predicates are based on any boolean comparison including constant values or values referenced in data attribute value(s). The projection operation copies all the data attribute value(s) for a particular attribute (also called a column).
How do things relate? (cont...)

+ **association (continued)**

+ Association between relations (or a relation and itself) is based upon relating (matching) data attribute values in tuples of one relation with those of another. The join operation pairs every combination of tuples from one relation with those of another relation and copies the data attribute values from the pairs into the result where the pairing satisfies a selection predicate. This relational operation is called join because facts from two sources are joined in the result.
How do things relate? (cont...)

* association (continued)

* Join compatibility requires that the values involved in comparisons (i.e. selection predicates) whether constants or data attribute values derive from the same data attribute domain.

* Referential Integrity: when relations are devised such that a tuple in one relation predisposes the existence of (owns) tuple(s) in another, the data attribute(s) of the second required to join the relations is called a foreign key. Referential integrity asserts that any value found in the data value attribute(s) of a foreign key must appear in a tuple of the first relation as the value of a candidate key or itself be null (of indeterminate value).
Normalization: relational model consistency depends on the semantic concurrence of the behavioral relationships and the objectives of the database modeler, the intension, (rather than the accident of a relation’s contents at any particular instant, its extension). The integrity properties defined above enable the database modeler to devise a structure and behavior of relations that avoid semantic discord called anomalies, the unintended loss or modification of information by relational operations. Relations designed to avoid certain kinds of anomalies are said to be normalized or in normal form. Normalization is the arrangement of data attributes and their relationships among relation structures to prevent particular anomalies.
Normalization (continued)

- **First Normal Form** asserts that every data attribute value is *atomic*, indivisible in value or form and may not be operated upon except as a whole and single value.

- **Second Normal Form** is first normal form and asserts that every data attribute value not in the primary key is *fully functionally dependent* upon the primary key. ("Fully" means applying to every data attribute of the primary key.)

- **Third Normal Form** presupposes first and second normal forms and asserts that no data attribute outside the primary key is *transitively* dependent upon the primary key. ("Transitively" means an attribute(s) functionally dependent upon an attribute functionally dependent upon an attribute (…) functionally dependent on the primary key.)
One More Time!
You Need to be able to Explain:

- The Relational Ontology
  - Individuals
  - Tuple
  - Attributes
    - Data Attribute
  - Classes
    - Relation
  - Relationships
    - Behavioral Relationships
      - Functional Dependency
      - Entity Integrity
      - Association
    - Relational Operations
      - Selection
      - Projection
      - Join
    - Join Compatibility
    - Referential Integrity
- Normalization
  - First Normal Form
  - Second Normal Form
  - Third Normal Form
The arrangement, presentation, original illustrations and organization of the materials are copyrighted by Leslie J. Waguespack, Ph.D. with all rights reserved (©2008). Derivations and excerpts of additional materials are referenced as follows:

