A SIMPLE GUIDE TO FIVE NORMAL FORMS

(See the next slide for required reading)

A SIMPLE GUIDE TO FIVE NORMAL FORMS

• Given an application, how to structure the tables that support this application?
  Solution: Use the five normal forms as guidelines. Why is this important? If
  one is not careful then:
  1. Information might be duplicated, resulting in: update anomalies, and data
     inconsistencies.
  2. Loss of information

• attribute = field

• First normal form: all occurrences of a record type must contain the same
  number of fields. Example: Emp(SS#, name, age, salary, dno) is a relation
  with five attributes. All tuples/records of this table have five attributes.

• Second and third normal forms: A non-key attribute is a fact about the key,
  the whole key and nothing but the key. Key is defined as a set of one or more
  attributes which, taken collectively, allow us to uniquely identify one record
  from the others in a table.

A SIMPLE GUIDE TO FIVE NORMAL FORMS (Cont…)

• The second normal form is violated when a non-key field is a fact about a
  subset of a key. This circumstance arises when the key is composite.

WAREHOUSE-ADDRESS QUANTITY WAREHOUSE PART
……

• Limitations:
  1. Warehouse address is repeated in every record (every part)
  2. If the address of a warehouse changes, many records must be updated
  3. The data may become inconsistent if the update is not done correctly
  4. If a warehouse becomes empty then the database management system might lose
     track of it. This is because no records may reflect the existence of the warehouse in
     the database.

• Solution: Satisfy the second normal form by decomposing the above relation
  into two different relations

<table>
<thead>
<tr>
<th>PART</th>
<th>WAREHOUSE</th>
<th>QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>......</td>
<td>key ........</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WAREHOUSE</th>
<th>WAREHOUSE-ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>......</td>
<td>key ........</td>
</tr>
</tbody>
</table>

A SIMPLE GUIDE TO FIVE NORMAL FORMS (Cont…)

• The process of replacing un-normalized records with normalized records is
  termed normalization.
• Advantage of normalization: it enhances the integrity of data by minimizing
  redundancy and inconsistency.
• Disadvantage of normalization: performance might be lost. For example, the
  execution of the following query now requires the execution of a join operator:
  Retrieve the address of all warehouses that contain part-id=5.

<table>
<thead>
<tr>
<th>PART</th>
<th>WAREHOUSE</th>
<th>QUANTITY</th>
<th>WAREHOUSE-ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>......</td>
<td>key ........</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A SIMPLE GUIDE TO FIVE NORMAL FORMS (Cont…)

• The third normal form is violated when a non-key attribute is a fact about
  another non-key attribute

<table>
<thead>
<tr>
<th>EMPLOYEE</th>
<th>DEPARTMENT</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>......</td>
<td>key ........</td>
<td></td>
</tr>
</tbody>
</table>
A SIMPLE GUIDE TO FIVE NORMAL FORMS (Cont...)  

• Address is a fact about department and not the faculty. In addition to suffering from the limitations of the second normal form, violating the third normal form suffers from update ambiguities: Logically speaking, an update that changes address of a faculty (say Shahram) from SAL to HNB has two alternative meaning:  
  1. Shahram is moving to a different department and his department attribute value will be updated later or  
  2. the computer science department is changing address and all the faculty in this department will observe an update in their address attribute later.  
• It is difficult to design algorithms to maintain consistency with such ambiguities.  
• Solution: decompose the above record into two records:  
  \[
  \begin{array}{l|l|l}
  \text{DEPARTMENT} & \text{EMPLOYEE} & \text{LOCATION} \\
  \text{key} & \text{key} & \text{key} \\
  \end{array}
  \]

A SIMPLE GUIDE TO FIVE NORMAL FORMS (Cont...)  

• Functional dependency: An attribute \( Y \) is functionally dependent on attribute (or set of attributes) \( X \) if it is invalid to have two records with the same \( X \) value but different \( Y \) values. A given \( X \) value must always occur with the same \( Y \) value.  
• If \( X \) is the key of a relation then all other attributes of a relation are by definition functionally dependent on \( X \).  
• The fourth and fifth normal forms minimize the number of attributes involved in a composite key. They deal with multi-valued facts. A multi-valued fact represents either a many-to-many relationship (e.g., employees having skills) or a many-to-one relationship (e.g., children of an employee assuming only one parent is employed).  
• With the fourth normal form, a record type should not contain two or more independent multi-valued facts about an entity. In addition, the record must satisfy third normal form. Example: bilingual employees with multiple skills. If language is independent of skills when we have two many-to-many relationships: (1) between employees and skills; (2) between employees and languages. Fourth normal form is violated when these two relationships are represented in a single record:  
  \[
  \begin{array}{l|l|l|l}
  \text{EMPLOYEE} & \text{SKILL} & \text{LANGUAGE} & \text{key} \\
  \text{key} & \text{key} & \text{key} \\
  \end{array}
  \]

A SIMPLE GUIDE TO FIVE NORMAL FORMS (Cont...)  

• To satisfy fourth normal form, they should be represented as two records:  
  \[
  \begin{array}{l|l|l}
  \text{EMPLOYEE} & \text{SKILL} & \text{LANGUAGE} \\
  \text{key} & \text{key} & \text{key} \\
  \end{array}
  \]

A SIMPLE GUIDE TO FIVE NORMAL FORMS (Cont...)  

• Fifth normal form deals with those cases where information can be reconstructed from smaller pieces of information which can be maintained with less redundancy. It deals with semantics and constraints! Consider the example represented in the first column of Page 70. Although the normalized form involves more record types, there may be fewer total record occurrences. This is because the normalized relations increase in an additive fashion while the unnormalized relations increase in a multiplicative fashion. For example, if we add a new agent who sells \( x \) product for \( y \) companies, where each of these companies makes each of these products, we have to add \( x \times y \) new records to the normalized set of new relations, but \( x \times y \) new records to the unnormalized relations.  
• Limitations:  
  1. With repetitions, the update must be done in multiple records and the records could become inconsistent.  
  2. Insertions of a new skill may involve looking for a record with a blank skill, inserting a new record with possibly blank language, or inserting multiple records pairing the new skill with some or all of the languages.  
  3. Deletion of a skill may involve blanking out the skill field in one or many records (perhaps with a check that this does not leave two records with the same language and a blank skill) or deleting one or more records, coupled with a check that the last mention of some language has not been deleted also.  
• Note that the fourth normal form is not violated if there is dependence between these multi-valued fields: e.g., an employee can exercise certain skills in certain languages. If Smith can cook French cuisine only, but type French, German, and Greek then the pairing of skills and languages is meaningful, and there is not longer an ambiguity of maintenance policies.