In many applications, our information about the world is incomplete. For example, we may know about a group of persons some information about their ages, such as that one person’s age is greater or less than forty or one person is older than another, without knowing their precise ages. Section 9.1 describes how our incomplete information about the whole world or universe can be represented as a single universal constraint object relation, or UCOR.

The UCOR requires different assumptions about the database than the assumptions made in earlier chapters. Section 9.2 describes the closed, open, and possible world assumptions. The UCOR satisfies the possible world assumption.

Refinement queries add (conjoin) new constraints to the UCOR. Each constraint added eliminates some of the possibilities or ambiguities from the UCOR. For example, as we add more constraints about the ages of the group of persons, we may get more precise knowledge about their ages. In the limit, this will approach the state where there is only one solution of the UCOR, and then we obtain the exact age of each person. Sections 9.3 and 9.4 describe the syntax and the semantics of refinement queries. Section 9.5 describes projection queries from the UCOR. Section 9.6 describes the evaluation of refinement queries. Finally, Section 9.7 considers the related notion of constraint XML objects.

9.1 The Constraint Objects Data Model

Our incomplete information can be described using constraint objects and a special relation called the universal constraint object relation, or UCOR.
Each *constraint object* is composed of a unique object identifier (oid) and a set of attributes. For example, each person in a set of persons with different names can be represented by an object, with oid equal to the name of the person and an age attribute variable. The UCOR contains all known constraints about all objects and their attribute variables.

For example, suppose we represent the persons Alice, Brian, and Tom as constraint objects with oids equal to their names and the constraint attribute $y$ representing their ages. If we know that Alice is three years older than Brian, the total age of the three persons is less than 60, and Tom is the youngest, then this information can be represented by the following UCOR:

```
UCOR

<table>
<thead>
<tr>
<th>Alice_Age</th>
<th>Brian_Age</th>
<th>Tom_Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice.y</td>
<td>Brian.y</td>
<td>Tom.y</td>
</tr>
</tbody>
</table>

Alice.y = Brian.y + 3
Alice.y + Brian.y + Tom.y ≤ 60
Tom.y ≤ Alice.y
Tom.y ≤ Brian.y
```

The UCOR is a constraint relation like the ones we saw earlier, except that instead of regular variables it contains attribute variables of constraint objects. All substitutions of nonnegative integer constants for the three attribute variables that satisfy the constraint are possible age combinations for the three persons.

### 9.2 Closed, Open, and Possible Worlds

There are three types of assumptions that are commonly made about the validity of the data stored in a database relation:

- **Closed world assumption:** This means that every tuple in the relation is true, and every tuple outside the relation is false. For example, the *Crops* relation satisfies the closed world assumption because every possible combination of planting is within the relation, and what is not a possible combination is left out.

- **Open world assumption:** This means that every tuple in the relation is true, and every tuple outside the relation is either true or false. For example, the *Go* relation satisfies the open world assumption. Every tuple within this relation is a possible travel between two cities, but there are travel possibilities that are not in the relation. Another example, comprises the following relations *Firesource* and *Flammable*: