CS252:HACD Fundamentals of Relational Databases

Notes for Section 3: Predicates and Propositions

1. Cover slide

Relational database theory is based very directly, as we shall see, on logic—specifically, in fact, on first-order predicate calculus.

Fortunately, we need only to understand a few basic principles on which logic is founded. You may well already have a good grasp of the principles in question, but we do need to fix the terminology to be used on this course and in any case it is a good idea to go over it again and pick out the points that are particularly relevant to relational theory.

2. Recommended Book

What I especially like about this book is the way Hodges relates logic to human language. I think this is very appropriate for databases, for reasons that I hope will soon become apparent.

3. What Is A Predicate?

To elaborate very briefly (for now) on that last point, consider what we get if we merely remove those designators, marking the holes that are left:

Student … is enrolled on course …

When we assume that the first hole is to be occupied by a student identifier, the second by a course identifier, we know the *general* meaning of every sentence that can be thus formed.

4. What Kind of Sentence?

So a declarative sentence has the form of a statement but is not necessarily a statement, as we shall see in Slide 6. A declarative sentence *denotes* a predicate—different sentences (in different languages, perhaps) can denote the same predicate.

5. Some Counterexamples

No notes.

6. Some Examples

Some are written in English, others in mathematical notation.

The last four illustrate the use of *parameters* (*x, y, a, b, c, s, c* again, and *x* again), also called *free variables*. In the present context we prefer the term *parameter* because we use *variable* for its usual meaning in computer languages.

The first four, containing no parameters, denote what are called *propositions*. A proposition is a declarative sentence (strictly the *meaning* of such a sentence) of which we can say "that is true" or "that is false". The last four are not declarative sentences but they do have the form of such sentences.

Note very carefully the use of nouns or noun phrases as *designators* in these sentences. These are "S1", "C1", "The king of France", "I", and "you". We can agree on what the sentences mean only if we have the same understanding of what their designators designate. In the case of "the king of France", it depends on the time at which the sentence is uttered—and at the present time designates nothing at all (a problem that has intrigued logicians). In the case of the pronouns "I" and "you", it depends on who is speaking to whom.

Some designators have the special property of always referring to the same thing. For example, numbers have this property, and so does "water". Such designators are the ones we use in relational databases.

"$x < y$" is a predicate with two parameters (a *dyadic* predicate). "$a + b = c$" is a triadic predicate. A proposition can be accurately thought of as a *niladic* (or *0-adic*) predicate (i.e., one having no parameters).

Actually, most textbooks on logic reserve the term "predicate" for those that contain at least one parameter. The generalisation to include propositions as the degenerate case is more appropriate in the context of relational theory, as we shall eventually see.

Notice that in the general form, $P(x)$, the P is italicised as well as the x. We use this form when we wish to talk about predicates in general. P is itself a variable, standing for some arbitrary predicate. The form appears to assume that every predicate is monadic, but in fact we do not lose generality because x can stand for a collection of parameters (possibly an empty collection).

7. Deriving Predicates from Predicates (1)

No notes.

8. Intension and Extension

Note the spelling of "intension". It does not mean "intention"! Of course, if the term "predicate" refers to the meaning of a declarative sentence, then we don't need the word "intension" as well. But "intension" is not only used for predicates; it is also used for designators. For example, at the time of writing "the president of the USA" and "Barack Obama" designate the same thing; but do they *mean* the same thing?

The reason why the concept of *extension* is so important is that a relational database consists of representations of extensions of predicates. As we shall see, we have a single method of representing an instantiation of a predicate and a single way of representing its extension by enumerating all of its instantiations that are (believed to be) true.

9. Deriving Predicates from Predicates (2)

Have a close look at the first example, "Student *s* is enrolled on course *c* **and** *s* is called *name*." The two predicates being connected both include an appearance of the parameter *s*. The resulting predicate therefore contains *two* appearances of this parameter and is a triadic predicate (having three parameters, *s, c,* and *name*), not a tetradic one (having four).

It is of utmost importance to note that when we substitute a designator for *s* we necessarily substitute it for every appearance of *s*. As a consequence, the following proposition is not an instantiation of "Student *s* is enrolled on course *c* and *s* is called *name*.": "Student S1 is enrolled on course C1 and S2 is called Boris."

Now look at the second example, "$a < b$ **or** $c < d$". Under the strict meaning of "predicate", the two sentences being connected are in fact the same predicate, because their meanings are identical (they have the same *intension*). But they differ in the symbols used for their parameters, and that makes them different sentences. And the resulting sentence is indeed a tetradic predicate, with instantiations such as "2 < 1 or 5 < 4" (recall that an instantiation isn't necessarily true), and "2 < 1 or 2 < 3" (we are of course permitted to substitute the same designator for two different parameters if we want to).

Relational DBMSs provide operators, closely allied to logical operators, that allow the extensions of derived predicates to be computed, given the extensions of the predicates from which they are derived. A collection of such operators is called a *relational algebra*.

10. Deriving Predicates from Predicates (3)

I have chosen propositions for these examples because we might need to be reminded about their truth values.

The first is false precisely when the first operand ("you ask me nicely") is true and the second ("I will marry you") is false; in all other cases, it is true, including in particular whenever "you ask me nicely" is false! It can alternatively be pronounced, "You asking me nicely implies that I will marry you." Notice that it is equivalent to "**Either** it is not the case that you ask me nicely, **or** I will marry you."

The second example is merely the inverse of the first. It can be read, "**If** I marry you, **then** you will have asked me nicely."

The third example is called logical equivalence because it is true whenever the two operands have the same truth value and false otherwise. So it could perhaps be written like this: "I will marry you" = "You ask me nicely".

11. Deriving Predicates from Predicates (4)

To quantify something is to say how many of it there are, being derived from the Latin *quantis*, meaning "how many".

Exact quantifiers such as "there are exactly 2 of" can be defined in terms of either the existential quantifier or the universal one, and either of these two can be derived from the other (with the help of negation):

"There is exactly one x such that p(x)" ≡ "There exists x such that p(x) and there does not exist y such that p(y) and $x \neq y$"

"For all x, p(x)" ≡ "There does not exist x such that it is not the case that p(x)"

"There exists x such that p(x)" ≡ "It is not the case that for all x it is not the case that p(x)"

12. Sets

I follow the convention by using Z to denote the set of all integers.

Sometimes a vertical bar is used instead of a semicolon:

{ x | $x \in$ Z **and** $1 < x$ **and** $x < 4$ }

And sometimes you see this:

{ $x \in$ Z | $1 < x$ **and** $x < 4$ }

which is perhaps more appropriate to the pronunciation shown in the slide:

"x is an integer such that $1 < x < 4$"

13. The Language of Sets (1)

If B is a subset of A, then every member of B is a member of A (and A is a superset of B).

If B is a proper subset of A, then B is a subset of A and some member of A is not a member of B.

Remember *the empty set.* It has no elements. It is a subset of every set, including itself. It is a superset of itself and no other set.

Note that an element might be a set. That allows us to talk about sets of sets, such as a certain set of subsets of a given set. In fact, there is a term for set of all the subsets of a given set: **power set**.

Exercises:

1. Write down the power set of {1,2,3}.
2. If *A* is a set of cardinality *n* (meaning that *A* contains *n* elements), what is the cardinality of the power set of *A*?

14. The Language of Sets (2)

The colon in "{*x* : ... }" is pronounced "such that".

15. EXERCISES

For each of the following propositions, state whether it is true or false, basing your conclusions on the relation depicted in Slide 15:

1. There exists a course *CourseId* such that some student named Anne is enrolled on *CourseId*.
2. Every student with StudentId S1 who is enrolled on some course is named Anne.
3. Every student who is enrolled on course C4 is named Anne.
4. Some student who is enrolled on course C4 is named Anne.
5. There are exactly 5 students who are enrolled on some course.
6. It is not the case that there is no course on which no student who is enrolled on some course but is not named Boris is not enrolled.
7. There are exactly 10 pairs of StudentIds (*SID1, SID2*) such that there is some course on which student *SID1* is enrolled and student *SID2* is enrolled.
8. There are exactly 3 pairs of StudentIds (*SID1, SID2*) such that there is some course on which student *SID1* is enrolled and student *SID2* is enrolled.
9. If a student named Eve is enrolled on course C1, then student S1 is named Adam.
10. If student S1 is named Anne, then S1 is enrolled on course C2.

End of Notes