

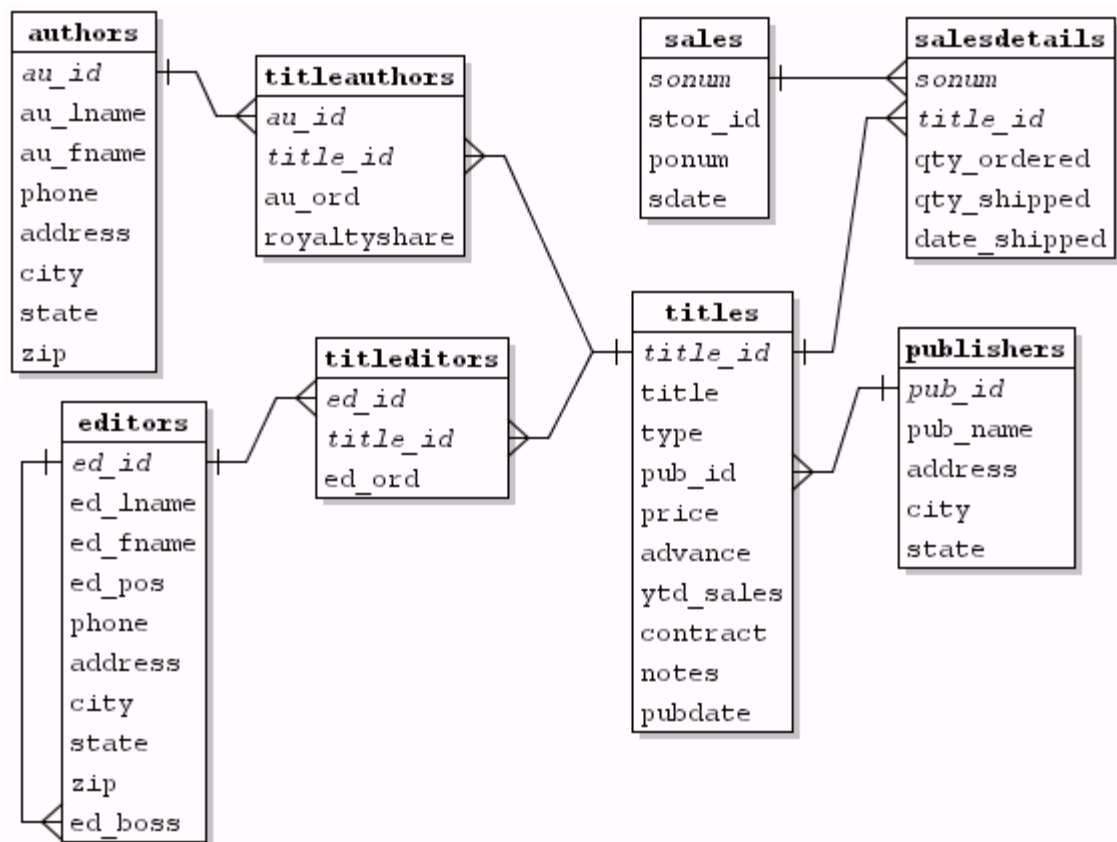
Datalog Exercise

(The latest Datalog software can be downloaded at <http://www.fdi.ucm.es/profesor/fernan/des/html/download.html>.)

1. Load Tables

Load the file bookbiz.dl to create the predicates for the following questions.

Browse through the predicates and try to understand the structures. Draw a ER diagram for this database.



2. Things to look out for:

- Do not put a space between the predicate name and the opening parenthesis. (i.e. link (a, b). is illegal).
- To test inequality, use the not predicate. (e.g. not(A = B))
- To test for null, use is_null. (e.g. is_null(A))
- Use >= and <= for testing inequalities.
- Use /abolish to clean out the buffer

- Use a new predicate name to “redefine” a rule since a new definition of a predicate will not erase an earlier rule.
- Use /assert to add new predicates and rules dynamically. (Alternatively, add to the bookbiz.dl file and load everything at the same time. A second /consult erases all previous loaded predicates and rules.)
- Calculation can be done in the body of a rule by assigning a new value to a variable. (e.g. ... A is B – C ..)
- The disjunction operator ; does not work with terms that begins with an upper case (even though it is quoted). E.g. given a('B', 'C'), a('D', 'E'), b(A) :- a(A, 'C') ; a(A, 'E'). will not return any result. But given a(b,c), a(d,e), b(A) :- a(A,c) ; a(A,e) will return b(b) and b(d).

3. Create Datalog Queries for each of the following:

- a. Show all columns and rows in the authors predicate.

Datalog:

SQL:

Relational Algebra:

- b. Show the first and last names of all authors.

Datalog:

SQL:

Relational Algebra:

- c. Which authors live in Walnut Creek? Show all columns. You should get 1 row (Akiko Yakomoto).

Datalog:

SQL:

Relational Algebra:

- d. Which orders are incomplete? List the title_ids and the number of titles that still have to be shipped to complete orders. Show only rows where not as many titles have been shipped as ordered. Use the salesdetails table for this. There should be 5 rows in your result.

Datalog:

SQL:

Relational Algebra:

- e. Which editors don't have a boss? List all editors (first and last names) without a boss (NULL in ed_boss field). There will be two rows in the result.

Datalog:

SQL:

Relational Algebra:

- f. Which editors do NOT have 993-86-0420 as a boss? List all editors (last names) and their bosses. HINT: This may not be quite as easy as it looks because of NULLs.

Datalog:

SQL:

Relational Algebra:

- g. Which non-business books cost between \$20 and \$30? Give the title, type, and price. There are 5 rows in the result.

Datalog:

SQL:

Relational Algebra:

- h. List the last names of all authors who have a letter 'k' in their last name? There should be three matching rows.

Datalog:

SQL:

Relational Algebra:

- i. List all titles (title) followed by their publishers (pub_name).

Datalog:

SQL:

Relational Algebra:

- j. Show all publishers (pub_name) who are in either California (CA), Indiana (IN), or Maryland (MD).

Datalog:

SQL:

Relational Algebra:

- k. Find all business books, as well as any books with a price higher than \$10, and include all books with an advance of less than \$20,000.

Datalog:

SQL:

Relational Algebra:

- l. How many authors are there in each state? Show two columns of output: the state code and the number of authors in that state. Note that there is a separate `authors.state` field; it isn't combined with the city as the diagram on page 47 shows.

Datalog:

SQL:

Relational Algebra: