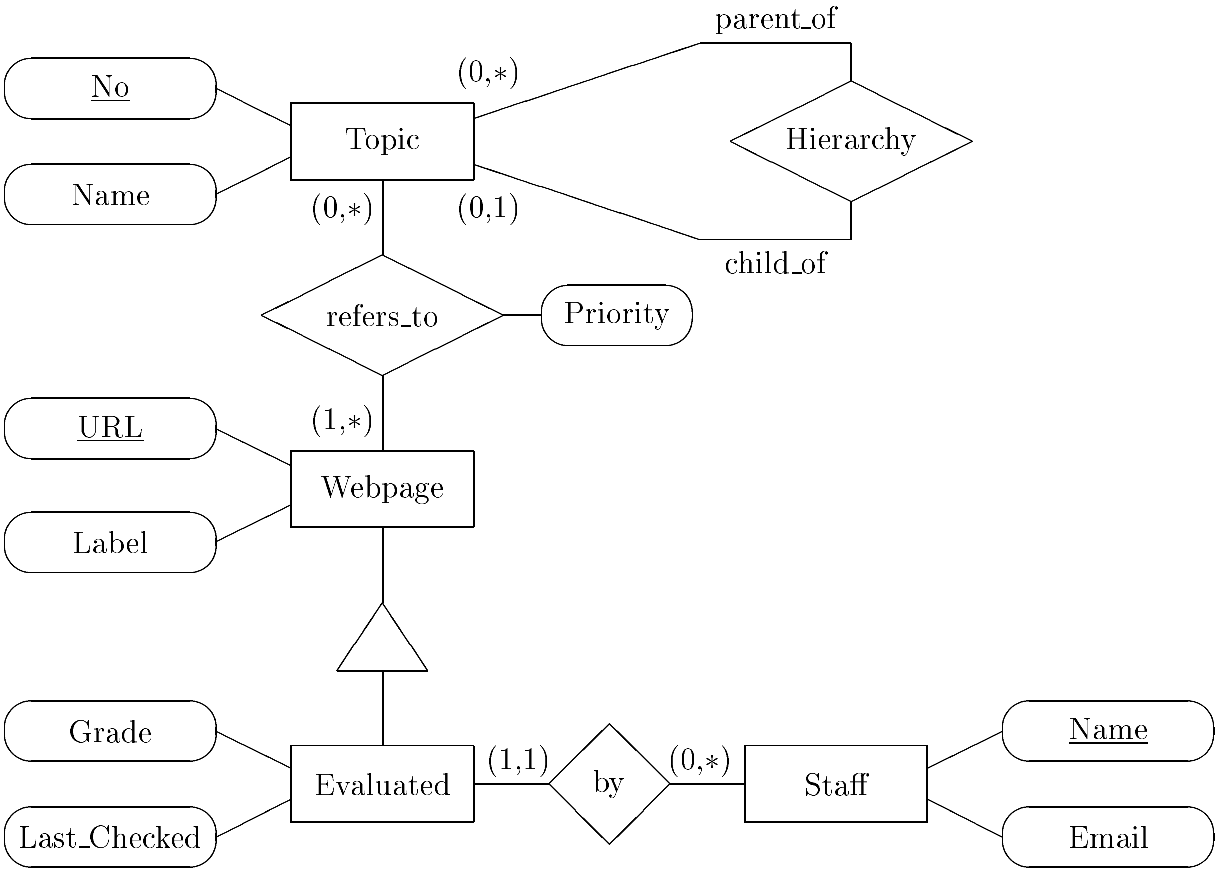
Dr. Stefan Brass
School of Information Sciences
University of Pittsburgh

April 19, 2000

# INFSCI 2710 "Database Management" — Example II for Final Exam —

## Exercise 1 (Translation ER to Relational Model) 10 Points

Consider a database for a web catalog like "Yahoo!". It contains a hierarchical list of topics, and for each topic a set of links to recommended web pages:




- One topic could e.g. be "Oracle". Topics are ordered in a hierarchy, so "Oracle" could be a sub-topic (child) of "Database Systems". Vice versa, "Database Systems" is more general than (parent of) "Oracle". Each topic can have only one parent topic, but many child topics. E.g. "DB2" could be another child topic of "Database Systems". The top entries in the hierarchy, e.g. "Software" have no parent.

- Each topic refers to a number of web pages which are relevant to this topic. E.g. the topic "Oracle" would refer to "`http://www.oracle.com`", but also to the Oracle user group and maybe to the homepage of this course.
- The same web page may be referenced in different topics. E.g. a page comparing Oracle and DB2 would appear in the lists of both topics.
- For each reference to a web page, we also store a priority. This is used for ordering the references to web pages. E.g. `http://www.oracle.com` must appear first in the listing for the topic Oracle (we cannot simply order them alphabetically). Note that the priority depends on both, the topic and the web page.
- For web pages we store their URL and the label which is used to refer to this page (usually the title of the page).
- Some of the pages were evaluated by staff members. They got a grade ("***", "**", "*" or no star). We also store the date when the grade was given or last re-checked (to see whether the web page has changed significantly).

Please translate this schema into the relational model. You can use an abbreviated syntax for the relations/tables, e.g. $R(A_1, \ldots, A_n)$. Please specify keys and foreign keys. If other constraints should be needed to make the relational schema equivalent to the given ER-diagram, it suffices to sketch them in natural language.

## Example Database for Exercise 2, 3, and 4

Consider a shop for customized computers on the world wide web. It offers a selection of different base computer models described in the table "Models":

| Models | | | |
|---|---|---|---|
| MNo | CPU | MHz | Price |
| 140 | Celeron | 400 | 600 |
| 150 | Celeron | 500 | 650 |
| 240 | Pentium II | 400 | 700 |
| 340 | Pentium III | 450 | 800 |
| 350 | Pentium III | 500 | 950 |
| ⋮ | ⋮ | ⋮ | ⋮ |

The table "Components" lists additional components which may be put into the computer systems:

| Components | | | |
|---|---|---|---|
| CNo | Name | Category | Price |
| 1001 | 64KB SDRAM | RAM | 60 |
| 1002 | 128KB SDRAM | RAM | 120 |
| 2001 | 6.4GB Ultra ATA | Disk | 100 |
| 2002 | 13.0GB Ultra ATA (7200 RPM) | Disk | 170 |
| 3001 | 8MB ATI XPERT | Video | 90 |
| 3002 | 16MB Diamond Viper | Video | 120 |
| 4001 | CD-ROM 40x | CD-ROM | 50 |
| 4002 | DVD 6x | DVD | 140 |
| ⋮ | ⋮ | ⋮ | ⋮ |

Next, there is a table with information about orders and customers. Each order is about one base computer with several of the optional components:

| Orders | | | | | | | |
|---|---|---|---|---|---|---|---|
| OrdNo | ODate | MNo | Name | Address | City | Zip | State |
| 14001 | 05-AUG-99 | 350 | J. Ullman | Broad Str | Stanford | 94305 | CA |
| 23050 | 09-AUG-99 | 240 | S. Brass | Centre Ave | Pittsburgh | 15213 | PA |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

The column "MNo" is a foreign key which references "Models".

Finally, "`Ordered_Comp`" stores which of the optional components belong to an order:

| Ordered_Comp | |
|---|---|
| OrdNo | CNo |
| 14001 | 1002 |
| 14001 | 2002 |
| 14001 | 3002 |
| 14001 | 4002 |
| 23050 | 1001 |
| ⋮ | ⋮ |

The column "OrdNo" is a foreign key which references "Order", and "CNo" is a foreign key referencing "Components". You can assume that all attributes in all tables are not null.

## Exercise 2 (SQL CREATE TABLE) 4 Points

Write a `CREATE TABLE` command for "Ordered_Comp". You can assume that the other tables are already declared. For which of the two foreign keys would "`DELETE CASCADES`" make more sense?

## Exercise 3 (SQL) 18 Points

Formulate the following queries in SQL. Your queries must work with any database state, not only with the example. You get three points for every correct query. You may lose points also for unnecessary complications or duplicate output rows. You may define views and use them in the queries.

a) How many orders are in the database for a model with Celeron or Pentium II processor? Print only a single total number (*not* a number for each model). Note that the database might contain further processor models besides the three ones shown.

b) Which orders contain a CD-ROM and a DVD drive (in one computer system)? Print only the order number of each such order. Make sure that no order number appears twice.

c) Print for each base model how often it is sold. The output should contain the product number, the CPU, the MHz, and the number of orders for this computer model. Computer models which were never sold do not have to appear in the output.

d) Which customers have not ordered any disk for their computer? Print the order number and the name of the customer.

e) Print the names of all customers who have ordered a computer system which five or more optional components. Make sure that each name appears only once in the output.

f) Print the total price for each order i.e. the sum of the prices of the optional components plus the price of the base model. There are orders without optional components, in which case the price of the base model must be printed. The output should contain one row for each order listing the order number and the total price.

## Exercise 4 (Views, Security) 3+2+2=7 Points

Our computer company has a contract with a shop called "Pittsburgh Computers". This shop also inputs orders into the database and gets 25% of the total price for each order this shop has acquired. Such orders are distinguished in the table "Orders" by an order number greater than 90000.

a) Please define a view `Pgh_Orders` with the same columns as `Orders` which only shows orders with a number greater than (or equal to) `90000`.

b) Please give the database user `pghshop` read access to the view and the right to insert tuples. Make sure that only tuples can be inserted which have an order number which marks them as orders acquired by the shop.

c) Suppose we would give the shop also the right to delete tuples. How would

```
DELETE FROM Pgh_Orders WHERE Name = 'S. Brass'
```

be translated to the base table? Or is this impossible? Then please explain why this view is not updatable.

## Exercise 5 (BCNF) 3+2=5 Points

Suppose a hotel uses the following table to store reservations for its rooms:

```
Reservation(Room_No, From_Date, To_Date, Guest_Name, Num_of_Nights)
```

They have discovered that the following functional dependencies hold:

- Room_No, From_Date ⟶ To_Date
- Room_No, To_Date ⟶ From_Date
- Room_No, From_Date ⟶ Guest_Name
- From_Date, To_Date ⟶ Num_of_Nights
- From_Date, Num_of_Nights ⟶ To_Date
- To_Date, Num_of_Nights ⟶ From_Date

a) Determine all possible keys for this relation. Only mention minimal keys, i.e. sets of attributes which together uniquely identify tuples, but of which no strict subset has the unique identification property.

b) Is this relation in BCNF? Explain your answer.