**DEPARTMENT OF COMPUTER APPLICATIONS**

**18MX24 - DBMS – EXTRENDED ERD – EXERCISES**

1. We wish to create a database for a company that runs training courses. For this, we must store data about the trainees and the instructors. For each course participant (about 5,000), identified by a code, we want to store her social security number, surname, age, sex, place of birth, employer’s name, address and telephone number, previous employers (and periods employed), the courses attended (there are about 200 courses) and the final assessment for each course. We need also to represent the seminars that each participant is attending at present and, for each day, the places and times the classes are held. Each course has a code and a title and any course can be given any number of times. Each time a particular course is given, we will call it an ‘edition’ of the course. For each edition, we represent the start date, the end date, and the number of participants. If a trainee is selfemployed, we need to know her area of expertise, and, if appropriate, her title. For somebody who works for a company, we store the level and position held. For each instructor (about 300), we will show the surname, age, place of birth, the edition of the course taught, those taught in the past and the courses that the tutor is qualified to teach. All the instructors’ telephone numbers are also stored. An instructor can be permanently employed by the training company or freelance. Develop an Extended ERD for the case study given above. Also, brief explain the model.

**Some rules for requirements analysis**

* Choose the appropriate level of abstraction.
* Standardize sentence structure.
* Avoid complex phrases.
* Identify synonyms and homonyms, and standardize terms.
* Make cross-references explicit.
* Construct a glossary of terms.

**An example of a glossary of terms**

|  |  |  |  |
| --- | --- | --- | --- |
| **Term** | **Description** | **Synonym** | **Links** |
| Trainee | Participant in a course. Can be an employee or self employed. | Participant | Course, Employer |
| Instructor | Course tutor. Can be freelance. | Tutor | Course |
| Course | Course offered. Can have various editions. | Seminar | Instructor, Trainee |
| Course | Company by which a trainee is employed or has been employed |  | Trainee |

**Rewriting and structuring of requirements (I)**

|  |
| --- |
| **Phrases of a general nature** |
| We wish to create a database for a company that runs training courses. We wish to hold the data for the trainees and the instructors. |
| **Phrases relating to the trainees** |
| For each trainee (about 5000), identified by a code, we will hold the social security number, surname, age, sex, town of birth, current employer, previous employers (along with the start date and the end date of the period employed), the editions of the courses the trainee is attending at present and those he or she has attended in the past, with the final marks out of ten. |
| **Phrases relating to the employers of the trainees** |
| For each employer of a trainee we will hold the name, address and telephone number. |

**Rewriting and structuring of requirements (II)**

|  |
| --- |
| **Phrases relating to the courses** |
| For each course (about 200), we will hold the name and code. Each time a particular course is given, we will call it an ‘edition’ of the course. For each edition, we will hold the start date, the end date, and the number of participants. For the editions currently in progress, we will hold the dates, the classrooms and the times in which the classes are held. |
| **Phrases relating to specific types of trainee** |
| For a trainee who is a self-employed professional, we will hold the area of expertise and, if appropriate, the professional title. For a trainee who is an employee, we will hold the level and position held. |
| **Phrases relating to the instructors** |
| For each instructor (about 300), we will hold surname, age, town of birth, all telephone numbers, the edition of courses taught, those taught in the past and the courses the instructor is qualified to teach. The instructors can be permanently employed by the training company or can be freelance. |

**Example of operational requirements**

* **operation 1:** insert a new trainee including all his or her data (to be carried out approximately 40 times a day);
* **operation 2**: assign a trainee to an edition of a course (50 times a day);
* **operation 3:** insert a new instructor, including all his or her data and the courses he or she is qualified to teach (twice a day);
* **operation 4:** assign a qualified instructor to an edition of a course (15 times a day);
* **operation 5:** display all the information on the past editions of a course with title, class timetables and number of trainees (10 times a day);
* **operation 6:** display all the courses offered, with information on the instructors who are qualified to teach them (20 times a day);
* **operation 7:** for each instructor, find the trainees all the courses he or she is teaching or has taught (5 times a week);
* **operation 8:** carry out a statistical analysis of all the trainees with all the information about them, about the editions of courses they have attended and the marks obtained (10 times a month).

**General criteria for data representation**

* If a concept has significant properties and/or describes classes of objects with an autonomous existence, it is appropriate to represent it by an entity.
* If a concept has a simple structure, and has no relevant properties associated with it, it is convenient to represent it by an attribute of another concept to which it refers.
* If the requirements contain a concept that provides a logical link between two (or more) entities, it is convenient to represent this concept by a relationship.

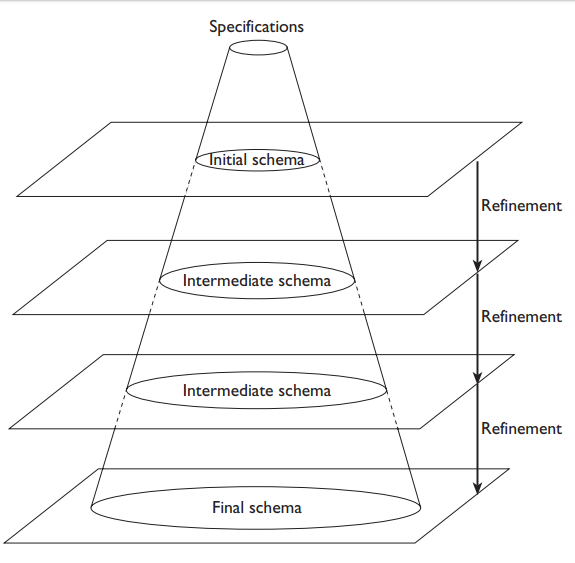
**Design strategies for conceptual design**

• The development of a conceptual schema based on its specification must be considered to all intents and purposes an engineering process, and, as such, design strategies used in other disciplines can be applied to it.

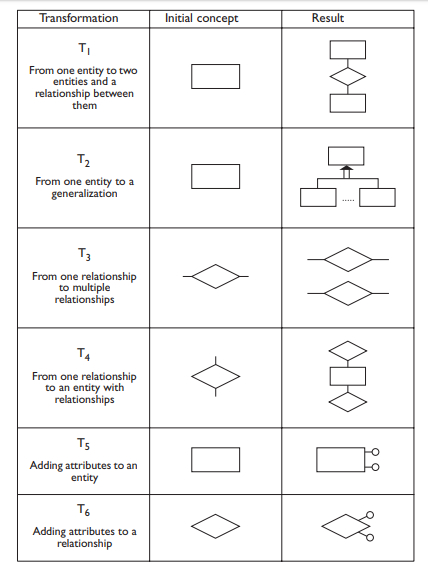
* Top-down
* Bottom-up
* Inside-out
* Mixed

**Top-down strategy**

* The conceptual schema is produced by means of a series of successive refinements, starting from an initial schema that describes all the requirements by means of a few highly abstract concepts.
* The schema is then gradually expanded by using appropriate modifications that increase the detail of the various concepts.
* Moving from one level to another, the schema is usually modified using some basic transformations called top-down transformation primitives.

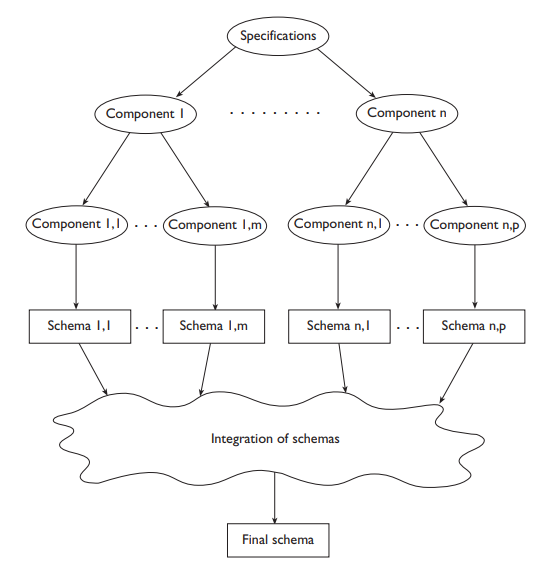
****

**Top-down transformation primitives**

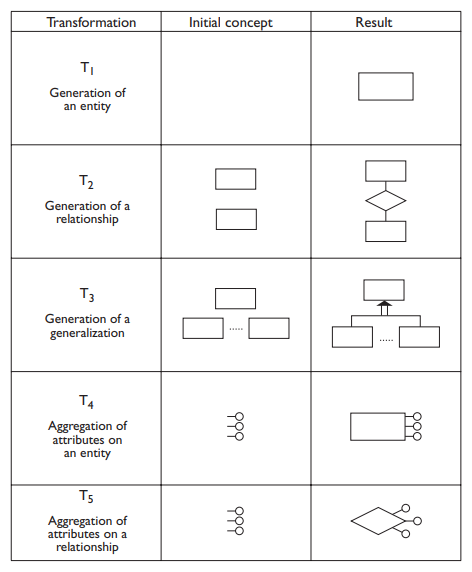
****

**Bottom-up strategy**

* The initial specifications are decomposed into smaller and smaller components, until each component describes an elementary fragment of the specifications.
* The various components are then represented by simple conceptual schemas that can also consist of single concepts.
* The various schemas thus obtained are then integrated until a final conceptual schema is reached.
* The final schema is usually obtained by means of some elementary transformations, called bottom-up transformation primitives.

****

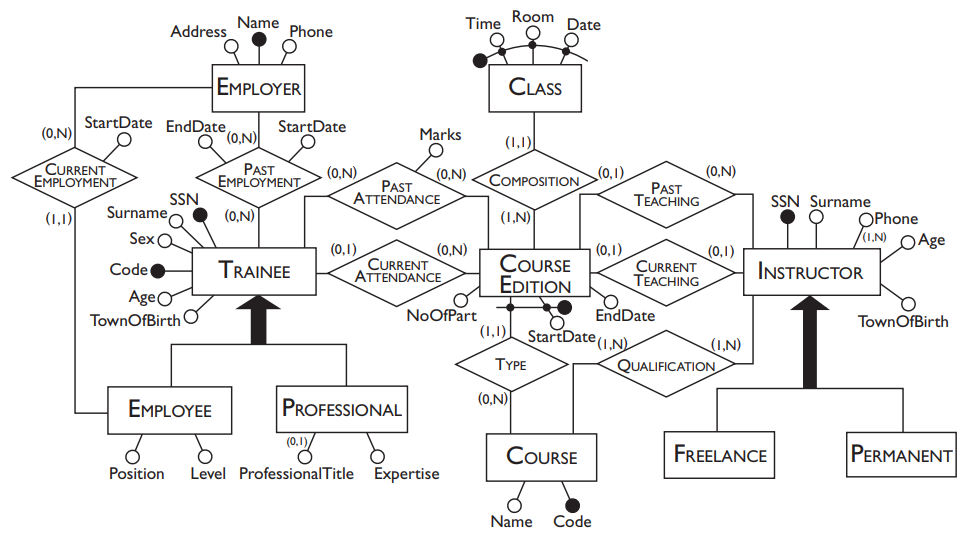
**Bottom-up transformation primitives**

****

**Inside-out strategy**

* This strategy can be regarded as a particular type of bottom-up strategy.
* It begins with the identification of only a few important concepts and, based on these, the design proceeds, spreading outward ‘radially’.
* First the concepts nearest to the initial concepts are represented, and we then move towards those further away by means of ‘navigation’ through the specification.

**ER-Diagram**

****

2. A library service wants to create a database to store details of its libraries, books and borrowers. Details include the following:’ A book has a unique ISBN number, a title and one or ore authors. The library service may own several copies of a given book, each of which is located in one of the service’s libraries. A given library contains many books, and in order to distinguish different copies of the same book a library assigns a different copy-number to each of its copies of a given book; the price that was paid for each copy is also recorded. Every library has a unique name and is either a main library or a branch library. A main library may have zero or more branch libraries and every branch library is a branch of exactly one main library. A borrower has a name and a unique ID code. A borrower can have many books on loan, but each copy of a book can only be on loan to one borrower. A borrower could borrow the same book on several occasions, but it is assumed that each such loan will take place on a different date.

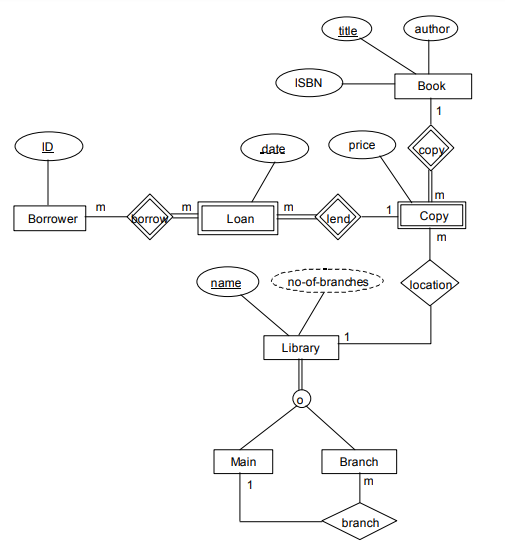
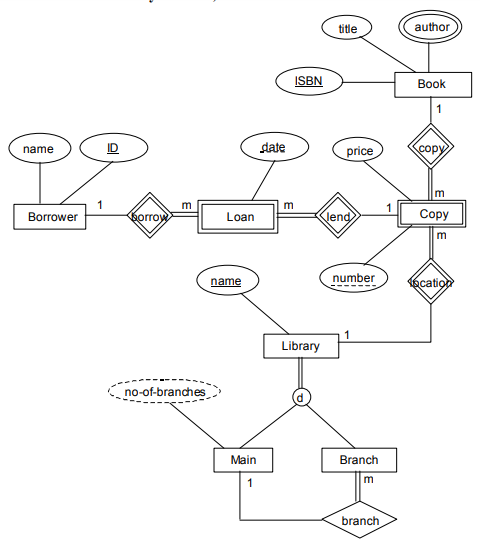


Figure 2 illustrates a preliminary design of an Extended-Entity-Relationship model intended to meet the above specification. The design contains at least 10 errors. Describe each error, clearly stating both the nature of the problem and its solution, and draw a corrected EER model.

1. ***author should be a multivalued attribute.***
2. ***location should be an identifying relationship.***
3. ***Copy should have total participation in location.***
4. ***a Loan can have only 1 Borrower.***
5. ***number should be a discriminating attribute for Copy.***
6. ***Branch should have total participation in the branch relationship.***
7. ***no-of-branches should be an attribute on Main not Library.***
8. ***specialisation of Library should be disjoint.***
9. ***Borrower should have a name attribute.***
10. ***ISBN should be the key of book, not name.***

******

3. A record company wishes to use a computer database to help with its operations regarding its performers, recordings and song catalogue. A requirements analysis has elicited the following information: Songs have a unique song number, a non-unique title and a composition date. A song can be written by a number of composers; the composer’s full name is required. Songs are recorded by recording artists (bands or solo performers). A song is recorded as a track of a CD. A CD has many songs on it, called tracks. CDs have a unique

record catalogue number, a title and must have a producer (the full name of the producer is required). Each track must have the recording date and the track number of the CD. A song can appear on many (or no) CDs, and be recorded by many different recording artists. The same recording artist might re-record the same song on different CDs. A CD must have only 1 recording artist appearing on it. CDs can be released a number of times, and each time the release date and associated number of sales is required.

Use this information to design an appropriate EER model.