

1. Some database models do not have a way to enforce referential integrity constraints. Explain this statement with one such example data model.

Relational databases express relationship between relations by introducing a Foreign Key. Using foreign key we can enforce integrity constraints by picking up appropriate option like cascading, default etc. There are no dangling references in relational model.

HTML is a data model which doesn't ensure integrity constraint. Sometimes we do get to see Error 404 if document at the other end of the HTML link is not available.

- 2 Explain the following statements with the help of *appropriate concrete example/s constructed by you*.
The Owner Entity Set and the Weak Entity Set must participate in a *one-to many* identifying relationship set where the Weak Entity Set must have *total participation* in the identifying relationship set.

(No credit will be given if you are using the example/s discussed in the class)

Example should not be : Course (course_ID, title, credits) and Section (sec_ID, semester, year) are related through Sec-Course relationship set. **(from University database)**

Example:

One-to-many: Entity set Section is weak, therefore it does not have its own key as there are multiple sections corresponding to the same course. The identifier / key for this weak entity set is derived by including primary key of the owner entity set. Identifier for Section is (course_ID, sec_ID, semester, year)

Total participation: Section must have a total participation otherwise there will be some section entities which are not been identified with any course (using Course_ID).

- 3 List the options we have in SQL (applicable to the other entity set) when we delete tuples from the strong entity set in a relationship when the other entity set is a strong or weak entity set.
Note: (Explain using an example constructed by you)
(No credit will be given if you are using the example/s discussed in the class)

Strong-Strong

Course(courseID, title, dept_name, credits)

Department(dept_name, building, budget)

if a particular department is deleted from Department

- **Cascade** (delete all **course** tuples which refer to the deleted **department** tuple)
- **Set Default** (sets foreign key **dept_name** value of a **Course** tuple to default)
- **Reject/ No Action** (delete action on **department** tuple is rejected on **Department**)

Weak-Strong

Course(course_ID, title, dept_name, credits)

Section(Sec_ID, semester, year)

if a particular course is deleted from Course

- **Cascade** (delete all **section** tuples which refer to the deleted **course** tuple)
- **Reject/ No Action** (delete action on **course** tuple is rejected on **Course**)

- 4 Clearly explain the DBMS versus File System Approach in the context of the Online Registration Database Management System of DAIICT.

Structure: files have data, data tables with constraints for data integrity

Concurrency control, crash recovery

Special query language SQL

How data is accessed, combined

Problems with file system/advantages with DBMS

- 5 Answer the following in the context of the *Specialization* in the E-R Model of *Digital Library Database Management System*.

This Specialization contains Resource (resource ID, title, price) as a super entity set. It has 3 sub entity sets :

- *Books*(author, publisher, no of Pages)
- *Journals* (publication year, volume, editor, list of papers)
- *Videos* (format, length)

What options you have when you *map this Specialization* to the corresponding *Relational Model* representation? How *covering constraints* influence the choice?

If Books, Journals, Videos don't cover Resource

- 4 Separate Relations: Resource, Books, Journals, Videos
- In future, any new kind of resource type (like Magazine) can be accommodated by creating a new relation (Magazine) by extending Resources relation.

If Books, Journals, Videos cover Resource

- 3 Separate Relations: Books, Journals, Videos
- In future, accommodating any new resource type (like magazine) will require us to create a new relation corresponding to that type (Magazine) from the scratch.