

Questions on Normalization

Q.1.

Consider the following relation:

CAR_SALE(Car#, Date_sold, Salesman#, Commission%, Discount_amt)

Assume that a car may be sold by multiple salesmen, and hence {CAR#, SALESMAN#} is the primary key. Additional dependencies are

Date_sold → Discount_amt

and

Salesman# → Commission%

Based on the given primary key, is this relation in 1NF, 2NF, or 3NF? Why or why not? How would you successively normalize it completely?

Answer:a. Its in 1 NF because no multi valued/composite attribute and no nested relations.

b. not in 2NF because partial dependency due to the FD, salesman#--> commission%, so commission% is partially dependent on the primary key {car#, salesman#}

Normalizing to 2NF:

Car_Sale

<u>car#</u>	<u>salesman#</u>	date_sold	discount_amt
-------------	------------------	-----------	--------------

Salesman

<u>salesman#</u>	commission%
------------------	-------------

c. Still not in 3NF as there is transitive dependency, {car#, salesman#}--> date_solddiscount_amount.

Normalizing to 3NF:

Salesman

<u>salesman#</u>	commission%
------------------	-------------

Car_Sale

<u>car#</u>	<u>salesman#</u>	date_sold
-------------	------------------	-----------

Discount_Info

<u>date_sold</u>	discount_amt
------------------	--------------

Q.2. (Book_Title is primary key)

Consider the following relation for published books:

BOOK (Book_title, Authorname, Book_type, Listprice, Author_affil, Publisher)

Author_affil refers to the affiliation of author. Suppose the following dependencies exist:

Book_title \rightarrow Publisher, Book_type

Book_type \rightarrow Listprice

Authorname \rightarrow Author-affil

- What normal form is the relation in? Explain your answer.
- Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition.

Answer:

a. Its in 2NF. It is already in 1 NF because no multi valued/composite attribute and no nested relations. Also it satisfies the conditions for 2NF as there are no partial dependency on primary key.

b. Applying 3NF because there is transitive dependency due to 2nd and 3rd FD.

Book(Book_title, Authorname, Book_type, Publisher)

Book_price_info(Book_type, List_price)

Authos (Author_name, Author_affil)

**** Note:** Table names are up to you, you can give any name you want, but it is preferable to give meaningful names as I have done above.

Q. 3.

Consider the following relational schema for a library database:

Book (Title, Author, Catalog_no, Publisher, Year, Price, bookCoverType, contractDate)

Collection (Title, Author, Catalog_no)

Assume {Author, Title} is the key for both relations

Additional functional dependencies are

I. Title, Author \rightarrow Catalog_no

II. Catalog_no \rightarrow Publisher, Year, bookCoverType

III. Publisher, bookCoverType \rightarrow Price

IV. Author \rightarrow contractDate

- Explain what normal form the relation is in.

b. Apply normalization until the 3rd NF. State reasons behind each normalization

Answer:

a. It's in 1 NF, because no multi valued/composite attribute and no nested relations.

b. It is not in 2NF as there is partial dependency due to FD IV. Therefore normalizing to 2NF:

Collection(Title, Author, Catalog_no)

Book(Title, Author, Catalog_no, Publisher, Year, Price, bookCoverType)

Author_Info(Author, ContractDate)

Now normalizing to 3NF as there is still transitive dependency in "Book" table due to Fd II and III.

Collection(Title, Author, Catalog_no)

Author_Info(Author, ContractDate)

Book(Title, Author, Catalog_no)

Catalog(Catalog_no, Publisher, year, bookcovertype)

Price_info(Publisher, bookcovertype, price)

****Important Note:** If any attribute is not mentioned in any FD, then by default they will be dependent on the whole primary key and therefore they should be placed in the original table. Example given below:

Project_Grade (Project_id, Student_id, student_name, project_title, grade, submission_date, bonus)

FD1: student_id--> student_name

FD2: project_id-->project_title

FD3: submission_date--> bonus marks

So, it is not mentioned what submission_date or grade depends on, so by default they should be considered to be fully dependent on entire primary key {project_id, student_id}

Solution:

a. In 1NF, not multivalued/composite attribute or nested relations.

b. FD1 and FD2 cause partial dependencies so not in 2NF. Normalizing to 2NF:

Project_Grade(Project_id, Student_id, grade, submission_date, bonus)

Student(Student_id, student_name)

Project(Project_id, project_title)

c. FD3 causes transitive dependency, so normalizing to 3NF.

Student(Student_id, student_name)

Project(Project_id, project_title)

Project_Grade(Project_id, Student_id, grade, submission_date)

Bonus_Info(submission_date, bonus)