Middle East Technical University

Department of Computer Engineering

CENG 352

Database Management Systems

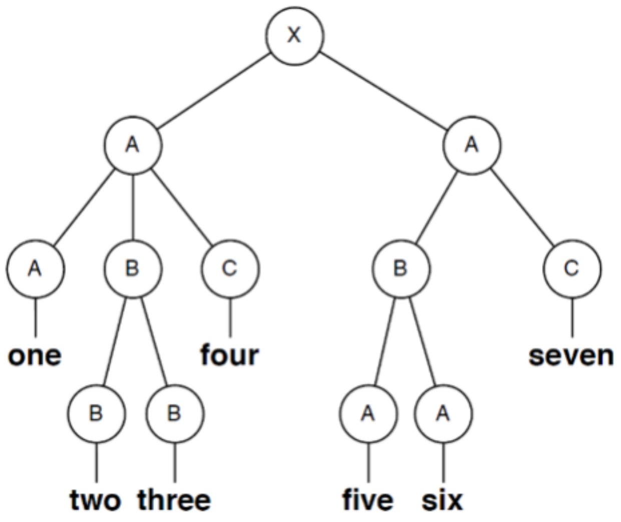
Spring 2020–2021

Written Assignment 1

1 XML and JSON

1.1 XML

Consider the following data in the form of a tree.



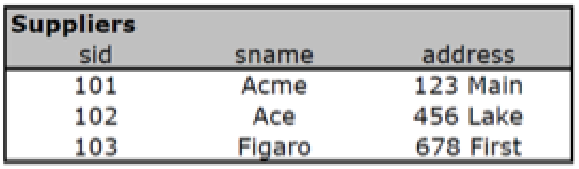
a. Write down the XML document that describes this tree.

b. For each of the XPath expressions below indicate what they return when evaluated on the data represented above. For example, if the expression is: /X/A/B/B/text() then you would answer: two, three.

  i. /X/A/C/text()
  ii. /X/A/*/text()
  iii. //A[B/A]/C/text()
  iv. /X/A[A]/B/*/text()
  v. //A/B/*/text()
  vi. /X/A[A][C]/B/*/text()

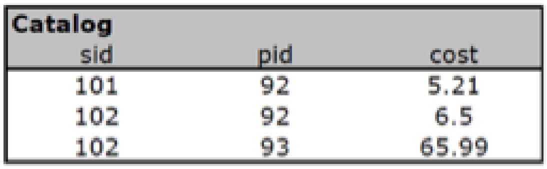
1.2 JSON

Represent the three relations shown below as a single JSON document. (Try http://jsonlint.com/ for an easy way to check the validity of your JSON document.)
There is more than one way to represent these relations in JSON. Please begin your document with a collection of Suppliers, each containing a collection of parts they sell and the price at which those parts are sold. Then find a reasonable way to include parts not sold by any supplier. In addition, explain whether this representation avoids redundancy.



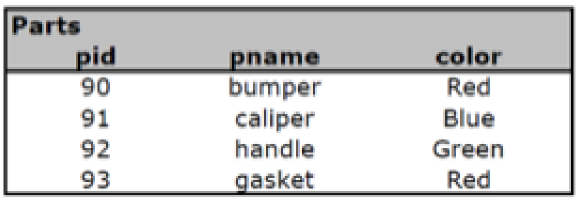
Suppliers

sid	sname	address
101	Acme	123 Main
102	Ace	456 Lake
103	Figaro	678 First



Catalog

sid	pid	cost
101	92	5.21
102	92	6.5
102	93	65.99



Parts

pid	pname	color
90	bumper	Red
91	caliper	Blue
92	handle	Green
93	gasket	Red

2 Database Design

2.1 BCNF Decomposition

Consider the following schema for table Conference:

Conference(PaperNo, FirstAuthorNo, AuthorNo, AuthorName, AuthorEmail, AuthorAddress, PaperTitle, PaperAbstract, PaperStatus, ReviewerNo, ReviewerName,ReviewerEmail, Comm-ments, ProgramComm,ReviewDate, Rating, ReviewerAddress)

a) Using the BCNF decomposition algorithm with the following list of FDs, design a database schema with minimal number of tables and no redundancies. You may need to merge some FDs in applying the algorithm or merge the resulting tables to have a good database design. In the final list of tables, you should show the primary keys, foreign keys, and unique constraints. You do not need to write CREATE TABLE statements.

- AuthorNo $\rightarrow$ AuthorName
- AuthorEmail $\rightarrow$ AuthorNo
- PaperNo $\rightarrow$ FirstAuthorNo
- AuthorNo $\rightarrow$ AuthorAddress
- AuthorNo $\rightarrow$ AuthorEmail
- PaperNo $\rightarrow$ PaperTitle
- PaperNo $\rightarrow$ PaperAbstract

- PaperNo → PaperStatus
- ReviewerNo → ReviewerName
- ReviewerNo → ReviewerEmail
- ReviewerEmail → ReviewerNo
- ReviewerNo, PaperNo → Comments
- ReviewerNo, PaperNo → ProgramComm
- ReviewerNo, PaperNo → ReviewDate
- ReviewerNo, PaperNo → Rating
- ReviewerNo → ReviewerAddres

b) Is your decomposition lossless join decomposition? Is it dependency preserving? Explain.

2.2 3NF Decomposition

Let R = A, B, C, D, E, F, G, H, K and assume the following functional dependencies:

- AC → BGH
- D → E
- G → B
- E → FK
- FD → K
- ADF → C
- H → BGH

a. Find a minimal cover.

b. Decompose this relation into 3NF.

2.3 Finding Dependencies

You are given a ".csv" file which contains a table for a simple database. The table has some data anomalies due to redundancy. Your task is to load this table into PostgreSQL and identify the functional dependencies that cause anomalies by writing SQL statements. Once you identify the "bad" functional dependencies, your task is to normalize the table. Do the following:

1. Download and install PostgreSQL if you haven't done so far and download "sample.csv" file that is provided to you.
2. Create a table in the database and load the table with the data given in the given ".csv" file. You can use the "COPY" property of PostgreSQL. You can get detailed information from here.
3. Find all functional dependencies in the table by writing appropriate SQL queries. Remember that a functional dependency is a constraint on a database instance. First, try to identify simple FDs like A $\rightarrow$ B, then try AB $\rightarrow$ C, etc. You should write an SQL query for each candidate FD. You can see if the FD holds or not by checking the answer of the query.
4. Decompose the table into BCNF tables using the FDs that you discovered. Create tables for normalized relations. Don't forget to create keys and foreign keys for the BCNF schema.
5. Load the new tables with the data from the original table. For this step, you should write SQL statements to load data into the new tables
6. Dump created database into 'eXXXXXXX.sql' file. You can dump the database using "pg_dump". Detailed information can be found here

What to turn in:

a) List of all FDs you identified and the corresponding SQL queries to discover them at the end of step 3 above.

b) List of all SQL statements to create normalized tables.

c) List of all SQL statements that load the contents of the tables.

d) Send "eXXXXXXX.sql" file along with the pdf. **If there is no ".sql" file your will only get half the grade from this question. Also the dump version sometimes becomes corrupted, please check before sending.**

3 SQL DDL

Assume the following database schema:

Customer(<u>CustNo</u>, CustFirstName, CustLastName, CustCity, CustState, CustZip, CustBal)
Employee(<u>EmpNo</u>, EmpFirstName, EmpLastName, EmpPhone, EmpEmail, EmpDeptName, EmpStatus, EmpSalary, supervisor) ##Note that supervisor is another Employee's no
FOREIGN KEY(supervisor) REFERENCES Employee

Product(<u>ProdNo</u>, ProdName, ProdPrice, ProdShipDate)

Order(<u>OrdNo</u>, CustNo, EmpNo, OrdDate, OrdName, OrdCity, OrdZip)
FOREIGN KEY(custno) REFERENCES customer
FOREIGN KEY(empno) REFERENCES employee
CustNo NOT NULL

Contains(<u>OrdNo, ProdNo</u>, Qty)
FOREIGN KEY(ordno) REFERENCES order
FOREIGN KEY(prodno) REFERENCES product

Using the schema above, answer the following questions:

Create Table Write create table statements for this database schema. Don't forget to specify the primary key and foreign keys. For foreign key constraints, assume the following:

- If a customer is deleted, all the records on order relation should also be deleted.
- If an order is deleted, all related contains record should also be deleted. Same thing holds if a product is deleted.
- If an employee is deleted, all orders taken by that employee should remain but the employee on that order should be set to NULL.
- If a supervisor is deleted, the employee should remain but its supervisor should be the system's default supervisor "007".

SQL Check Define appropriate SQL CHECK clauses to implement the following constraints:

- Each order should contain at least 3 of the same products.
- Each order name should include the order city. For example if the order city is "Paris" order name must include "Paris" somewhere, "ParisOrder111" or "OrderParis111" are possible order names.
- Our company decided to make employee emails a little bit complex. Which means an employee's email should not contain employee's name or lastname.

Assertion Define an assertion to ensure that each order have at least 30 products in total (Not distinct products, you should check total quantity of the products).

Trigger Define an AFTER trigger to implement the following constraint on the Employee table: "Whenever an employee's salary has been changed, if the salary is increased more than 15 percent, set the Status of that employee to 'Successful'."

4 Submission

You should send a pdf file, named ‘eXXXXXXX.pdf’ (your seven-digit ID number) contains your answers. You can prepare the pdf using both ‘DOCS’ or ‘Latex’, doesn’t matter. Moreover, you can send a scanned version of your handwritten answers, but please be sure that it is **readable**.

For Question 2.3, if you completed all of the steps you should have created an ‘eXXXXXXX.sql’ file (your seven-digit ID). Compress both ‘.pdf’ file and ‘.sql’file to ‘eXXXXXXX.zip’ file (‘.rar’ or ‘.tar.gz’ are also accepted) and upload it.