Course: COSC457.102 – Database Management Systems

Nathan Cox

Instructor: Leon Bernard

Room: YR- 304

Assignment #5

Due Date: April 16, 2020 11:59pm

*Exercise 13.7*

Create part of an XML instance document to correspond to the data stored in the relational database shown in Figure 5.6 such that the XML document conforms to the XML schema document in Figure 13.5.

For EMPLOYEE:

< ?xml version=”1.0” encoding=”UTF-8”? >

<xsi= http://www.w3.org/2001/XMLSchema-instance>

<Employee>

<Ssn>123456789</Ssn>

<Fname>John</Fname>

<Minit>B</Minit>

<Lname>Smith</Lname>

<Bdate>1965-01-09</Bdate>

<Address>731 Fondren, Housten, TX</Address>

<Sex>M</Sex>

<Salary>30000</Salary>

<Super\_Ssn>333445555</Super\_Ssn>

<Dno>5</Dno>

</Employee>

For Department:

< ?xml version=”1.0” encoding=”UTF-8”? >

<xsi= http://www.w3.org/2001/XMLSchema-instance>

<Department>

<Dnumber>5</Dnumber>

<Dname>Research</Dname>

<Mgr\_ssn>333445555</Mgr\_ssn>

<Mgr\_start\_date>1988-05-22</Mgr\_start\_date>

</Department>

For DEPT\_LOCATIONS

< ?xml version=”1.0” encoding=”UTF-8”? >

<xsi= http://www.w3.org/2001/XMLSchema-instance>

<Dept\_Locations>

<Dnumber>1</Dnumber>

<Dlocation>Houston</Dlocation>

</Dept\_Locations>

For WORKS\_ON

< ?xml version=”1.0” encoding=”UTF-8”? >

<xsi= http://www.w3.org/2001/XMLSchema-instance>

<Works\_On>

<Essn>1</Essn>

<Pno>1</Pno>

<Hours>32.5</Hours>

</Works\_On>

For PROJECT

< ?xml version=”1.0” encoding=”UTF-8”? >

<xsi= http://www.w3.org/2001/XMLSchema-instance>

<Project>

<Pnumber>1</Pnumber>

<Pname>ProductX</Pname>

<Plocaiton>Bellaire</Plocation>

<Dnum>5</Dnum>

</Project>

For DEPENDENT

< ?xml version=”1.0” encoding=”UTF-8”? >

<xsi= http://www.w3.org/2001/XMLSchema-instance>

<Dependent>

<Essn>333445555</Essn>

<Dependent\_name>Alice</Dependent\_name>

<Sex>F</Sex>

<Bdate>1986-04-05</Bdate>

<Relationship>Daughter</Relationship>

</Dependent>

*Exercise 13.10*

Specify the following views as queries in XQuery on the company XML schema shown in Figure 13.5.

1. A view that has the department name, manager name, and manager salary for every department

LET $d := doc([www.company.com/info.xml](http://www.company.com/info.xml))

FOR $x IN $d/company/department

$y IN $d/company/employee

WHERE $x.departmentManagerSSN = $y.employeeSSN

RETURN $x/departmentName,

$y/employeeName, $y/employeeSalary

1. A view that has the employee name, supervisor name, and employee salary for each employee who works in the Research department

LET $d := doc([www.company.com/info.xml](http://www.company.com/info.xml))

FOR $x IN $d/company/department[departmentName=”Research”],

$y IN $d/company/employee

$z IN $d/company/employee

WHERE $y/departmentNumber = $x/departmentNumber AND $y.employeeSupervisorSSN = $z.employeeSSN

RETURN $y/employeeName, $y/employeeSalary, $z/employeeName

1. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project

LET $d := doc([www.company.com/info.xml](http://www.company.com/info.xml))

FOR $w IN $d/company/project,

$x IN $d/company/department,

$y IN $d/company/employee,

$z IN $d/company/employeeWorksOn

WHERE $w.projectDepartmentNumber = $x.departmentNumber AND $x/departmentNumber = $y/departmentNumber AND $y.employeeSupervisorSSN = $z.employeeSSN  
RETURN $x/projectName, $x/departmentName, {count($y)}, {sum($z/hours)}

1. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the -Duplicate Question?

*Exercise 14.28*

Consider the relation R, which has attributes that hold schedules of courses and sections at a university; R = {CourseNo, SecNo, OfferingDept, CreditHours,

CourseLevel, InstructorSSN, Semester, Year, Days\_Hours, RoomNo,

NoOfStudents}. Suppose that the following functional dependencies hold on R:

{CourseNo} -> {OfferingDept, CreditHours, CourseLevel}

{CourseNo, SecNo, Semester, Year} ->

{Days\_Hours, RoomNo, NoOfStudents, InstructorSSN}

{RoomNo, Days\_Hours, Semester, Year} -> {InstructorSSN, CourseNo, SecNo}

Try to determine which sets of attributes form keys of R. How would you normalize this relation?

There is no partial dependency so R is in 2nd NF. Converting to 3 NF; transitive dependencies: {Course\_no} -> {Offering\_dept, Credit\_hours, Course\_level} is removed.

R1 : {Course\_No, Offering\_dept, Credit\_hours, Course\_level}

-FD in R1 : {Course\_no} -> {Offering\_dept, Credit\_hours, Course\_level} With key = {Course\_No}

-R2 : {Course\_No, Sec\_no, Instructor\_ssn, Semester, Year, Days\_hours, Room\_no, No\_of\_students}

-FD in R2 : {Room\_no, Days\_hours, Semester, Year} ->{Instructor\_ssn, Course\_No, Sec\_no}

and {Course\_no, Sec\_no, Semester, Year} -> {Days\_hours, Room\_no, No\_of\_students, Instructor\_ssn}

-Relation is in 3NF does not have transitive dependencies.

--> Converting to BCNF :

All attributes are functionally dependent on Super key of respective relation.

Therfore, R1 and R2 both are in BCNF.

*Exercise 14.31*

Consider the following relation for published books:

BOOK (Book\_title, Authorname, Book\_type, Listprice, Author\_affil, Publisher)

Author\_affil referes to the affiliation of the author. Suppose the following dependencies exist:

Book\_title -> Publisher, Book\_type

Book\_type -> Listprice

Author\_name -> Author-affil

1. What normal form is the relation in? Explain your answer.

It is in 1st NF because the normal form relation at most times contains only atomic values.

1. Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition.

2NF:

Book1(Book\_title, Authorname)

Book2(Book\_title, Publisher, Book\_type, Listprice)

Book3(Authorname, Author\_affil)

This decomposition eliminates the partial dependencies.

3NF:

Book1(Book\_title, Authorname)

Book2-2(Book\_title, Publisher, Book\_type)

Book2-3(Book\_type, Listprice)

Book3(Authorname, Author\_affil)

This decomposition eliminates the transitive dependency of Listprice

*Exercise 14.35*

Consider the relation:

BOOK (Book\_Name, Author, Edition, Year)

with the data:

|  |  |  |  |
| --- | --- | --- | --- |
| Book\_Name | Author | Edition | Year |
| DB\_fundamentals | Navathe | 4 | 2004 |
| DB\_fundamentals | Elmasri | 4 | 2004 |
| DB\_fundamentals | Elmasri | 5 | 2007 |
| DB\_fundamentals | Navathe | 5 | 2007 |

1. Based on a common-sense understanding of the above data, what are the possible candidate keys of this relation?

{Author, Edition},{Author, Copyright\_year}, {Book\_name, Author, Edition}, {book\_name, author, copyright\_year}, {author, edition, copyright\_year}, {book\_name, Author, Edition, Copyright\_year}

1. Does the above have one or more functional dependency (do not list FDs by applying derivation rules)? If so, what is it? Show how you will remove it by decomposition.

This relation has Multi valued Dependency.

1. Does the resulting relation have an MVD? If so, what is it?

Yes. It is {Book} -> {Author}|{Edition, Year}

1. What will the final decomposition look like?

Book1(Book\_Name, Author, Edition)

Book2(Edition, Copyright\_Year)

Highest Normal Form:

Book1\_1(Book\_Name,Author)

Book1\_2(Book\_Name, Edition)

Book2(Edition, Copyright\_Year)