Abhinav Yalamaddi - YXA210040

Database Design - CS 6360.002 - Fall 2022 - HW3A

4.12. Specify the following queries in SQL on the database schema of Figure 1.2.
a. Retrieve the names of all senior students majoring in ‘CS’ (computer science).
SELECT Name FROM STUDENT WHERE Class = 4 AND Major = 'CS';

b. Retrieve the names of all courses taught by Professor King in 2007 and 2008.
SELECT Course.Course_name FROM section s JOIN course c ON s.Course_number = c.Course_number WHERE s.Instructor = "King" AND (s.Year = "08" OR s.Year = "07");

c. For each section taught by Professor King, retrieve the course number, semester, year, and a number of students who took the section.
SELECT s.Course_number, s.Semester, s.Year, COUNT(g.Student_number) AS Num_Students FROM section s JOIN grade_report g ON s.Section_identifier = g.Section_identifier WHERE s.Instructor = "King"

d. Retrieve the name and transcript of each senior student (Class = 4) majoring in CS. A transcript includes the course name, course number, credit hours, semester, year, and grade for each course completed by the student.
SELECT st.name, c.Course_name, c.Course_number, c.Credit_hours, s.Semester, s.Year, g.Grade FROM student st, course c, grade_report g, section s WHERE st.Student_number = g.Student_number AND g.Section_identifier = s.Section_identifier AND s.Course_number = c.Course_number AND st.Class = "4";

4.13. Write SQL update statements to do the following on the database schema shown in Figure 1.2.
a. Insert a new student, <‘Johnson’, 25, 1, ‘Math’>, in the database.
INSERT INTO STUDENT (NAME, STUDENT_NO, CLASS, MAJOR) VALUES (‘Johnson’, 25, 1, ‘Math’)

b. Change the class of student ‘Smith’ to 2.
UPDATE Student
SET class = 2
WHERE Name = ‘Smith’;

c. Insert a new course, <‘Knowledge Engineering’, ‘CS4390’, 3, ‘CS’>.
INSERT INTO COURSE (Course_Name, Course_Number, Credit_hours, Department) VALUES (‘Knowledge Engineering’, ‘CS4390’, 3, ‘CS’)

Abhinav Yalamaddi
YXA210040

d. Delete the record for the student whose name is ‘Smith’ and whose student number is 17.
DELETE FROM STUDENT WHERE Name = ‘Smith’ AND Student_Number = ‘17’

6.16. Specify the following queries on the COMPANY relational database schema shown in Figure 5.5, using the relational operators discussed in this chapter. Also, show the result of each query as it would apply to the database state in Figure 3.6.
a. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project.
SELECT Fame, Lname,
FROM EMPLOYEE AS E, PROJECT AS P, WORKS_ON AS W
WHERE E.Ssn=W.Essn AND W.Pno=P.Pnumber AND P.Pname=’Product X’ AND W.HOURS>10

b. List the names of all employees who have a dependent with the same first name as themselves.
SELECT Fname, Lname
FROM EMPLOYEE, DEPENDENT
WHERE Ssn=Essn AND Dependent_name=Fname

d. For each project, list the project name and the total number of hours per week (by all employees) spent on that project.
SELECT P.Pname, PS.H
FROM PROJECT AS P, (SELECT Pno SUM(hours) AS H
FROM WORKS_ON
GROUP BY Pno) AS PS
WHERE P.Pnumber=PS.Pno

e. Retrieve the names of all employees who work on every project.
SELECT Fname, Lname FROM EMPLOYEE WHERE NOT EXIST (SELECT * FROM PROJECT WHERE NOT EXIST (SELECT * FROM WORKS_ON WHERE Pnumber=Pno AND Essn=Ssn))

f. Retrieve the names of all employees who do not work on any project.
SELECT Fname, Lname FROM EMPLOYEE AS E WHERE E.Ssn NOT IN (SELECT Essn FROM WORKS_ON)

g. For each department, retrieve the department name and the average salary of all employees working in that department.
SELECT Dname, AVG(Salary) FROM EMPLOYEE JOIN DEPEARTMENT ON Dno=Dnumber GROUP BY (Dnumber,Dname)

i. Find the names and addresses of all employees who work on at least one project located in Houston but whose department has no location in Houston.
SELECT E.Fname,E.Lname,E.Address
FROM EMPLOYEE AS E WHERE E.Ssn IN (SELECT Essn FROM WORKS_ON WHERE Pno IN (SELECT Pnumber FROM PROJECT WHERE Plocation='Houston')) AND NOT EXISTS(SELECT * FROM DEPT_LOCATIONS WHERE Dnumber=Dno AND Dlocation='Houston')

j. List the last names of all department managers who have no dependents.
SELECT E.Lname FROM EMPLOYEE AS E, DEPARTMENT AS D WHERE E.Ssn=D.Mgr_ssn AND D.Mgr_ssn NOT IN (SELECT Essn FROM DEPENDENT)

15.2. Discuss insertion, deletion, and modification anomalies. Why are they considered bad? Illustrate with examples.
Insertion Anomalies - Insertion anomalies are issues that arise when you initially enter data into a database. We must enter the necessary information into the table so that it matches the values in the other rows. Missing entries and entries with incorrect formatting are two of the most common insertion issues.

Deletion Anomalies - Deletion anomalies are issues with data deletion, either while attempting to delete and being stopped by an error or by undetected data loss. If the table item representing the final bit of data is destroyed, the information about it is likewise gone from the database. The least likely of these to catch you or prevent you from moving forward. The most costly deletion errors in terms of recovery may be those that get unnoticed for an extended period of time.

Update Anomalies - Modification or update anomalies are data irregularities caused by redundant data or incomplete updates. Update anomalies are problems caused by duplicate data in database tables. Inconsistency in the database will ensue if a modification is not implemented for all relevant entries. As a result, every database insertion, deletion, or alteration that results in an inconsistent state in the database is classified as an update anomaly.

15.24. ConsidertheuniversalrelationR={A,B,C,D,E,F,G,H,I,J}andthesetof functional dependencies F = { {A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J} }. What is the key for R? Decompose R into 2NF and then 3NF relations.

Closure of A = {A,D,E,I,J}
Closure of B = {B, F, G, H}
Closure of F = {F,G,H}
Closure of D = {D,I,J}
Closure of AB = {A,B, C, D, E,F, G,H,I,J}

So primary Key is AB because its closure includes all the attributes

2NF Decomposition -> R1 = {A,D,E,I,J} , R2 = {B,F,G,H}, R3 = {A,B,C}

Abhinav Yalamaddi
YXA210040

F1 = {A}→{D, E}, {D}→{I, J} ,
F2 = {B}→{F}, {F}→{G, H}, F3 = {A, B}→{C}
3NF Decomposition -> R1 = {A,D,E}, R1' = {D,I,J} [F1 = {A}→{D, E}, F1' = {D}→{I, J}]
R2 = {B,F} , R2' = {F,G,H} [F2 = {B}→{F}, F2' = {F}→{G, H}]
R3 = {A,B,C} [F3 = {A, B}→{C}]

15.28. Consider the relation R, which has attributes that hold schedules of courses and sections at a university; R = {Course_no, Sec_no, Offering_dept, Credit_hours, Course_level, Instructor_ssn, Semester, Year, Days_hours, Room_no, No_of_students}. Suppose that the following functional dependencies hold on R:
{Course_no} → {Offering_dept, Credit_hours, Course_level} {Course_no, Sec_no, Semester, Year} → {Days_hours, Room_no, No_of_students, Instructor_ssn}
{Room_no, Days_hours, Semester, Year} → {Instructor_ssn, Course_no, Sec_no}
Try to determine which sets of attributes form keys of R. How would you normalize this relation?

Closure of Course_no = {Course_no, Offering_dept, Credit_hours, Course_level }

Closure of Course_no, Sec_no, Semester, Year = {Course_no, Sec_no, Semester, Year, Days_hours, Room_no, No_of_students, Instructor_ssn, Offering_dept, Credit_hours, Course_level}

Closure of Room_no, Days_hours, Semester, Year = {Instructor_ssn, Course_no, Sec_no, Room_no, Days_hours, Semester, Year, Offering_dept, Credit_hours, Course_level, No_of_students }

2 set of Keys **-** {Room_no, Days_hours, Semester, Year} and {Course_no, Sec_no, Semester, Year}

Let's assume the second set as the primary key {Course_no, Sec_no, Semester, Year}
Because Offering_dept, Credit_hours, and Course_level depend upon Course_no which is a prime key attribute, it violates 2NF. So decompose it into {Course_no, Offering_dept, Credit_hours, Course_level} and {Course_no, Sec_no, Semester, Year, Days_hours, Room_no, No_of_students, Instructor_ssn}

FD1 = {Course_no} → {Offering_dept, Credit_hours, Course_level}
FD2 = {Course_no, Sec_no, Semester, Year} → {Days_hours, Room_no, No_of_students, Instructor_ssn}, {Room_no, Days_hours, Semester, Year} → {Instructor_ssn, Course_no, Sec_no}

Abhinav Yalamaddi
YXA210040

15.30. Considerthefollowingrelation:
CAR_SALE(Car#, Date_sold, Salesperson#, Commission%, Discount_amt)
Assume that a car may be sold by multiple salespeople, and hence {Car#, Salesperson#} is the primary key. Additional dependencies are
Date_sold → Discount_amt and Salesperson# → 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?
It violates all three normal forms.
1NF -> SalesA person is not atomic. it's a multivalued
2NF -> Commission depends on the Salesperson which is part of the primary key
3NF -> Discount amount depends on date_sold, which is a non prime attribute; it's also not in 2NF. To satisfy 1st normal form decompose R to R1 and R2
R1 = (Car#, Date_sold, Discount_amt)
FD1 = Date_sold → Discount_amt
R2 = (Car#, Salesperson#, Commission%)
FD2 = Salesperson# → Commission%
To Satisfy 2NF Decompose R2 to R2’
R2 = (Car#, Salesperson#)
R2’ = (Salesperson#, Commission%)
FD2’ = Salesperson# → Commission%
To Satisfy 2NF Decompose R1 to R1 and R1’
R1 = (Car#, Date_sold)
R1’ = (Date_sold, Discount_amt)
FD1’ = Date_sold → Discount_amt

15.31. Considerthefollowingrelationforpublishedbooks:
BOOK (Book_title, Author_name, Book_type, List_price, Author_affil, Publisher)
Author_affil refers to the affiliation of the author. Suppose the following dependencies exist:
Book_title → Publisher, Book_type Book_type → List_price Author_name → Author_affil
a. What normal form is the relation in? Explain your answer.
The key is {Book_title, Author_name}. The relation is in 1NF because no multivalued attribute exists, and it’s not in 2NF because Author_affil is partially dependent on Author_name and Publisher, Book_type are partially dependent on Book_title.
b. Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition.
The relation BOOK must be decomposed into 2NF, Author_affil, Author_name and Publisher need to be separated to new relations, so the result is {Book_title, Author_name, Book_type, List_price },{Book_title, Publisher, Book_type},{Author_name, Author_affil}
Then the relation need to be decomposed into 3NF, Book_type need to be separated to a new relation, so the result is {Book_title, Author_name, Book_type},{Book_title, Publisher, Book_type},{Author_name, Author_affil}, {Book_type, List_price }.

Abhinav Yalamaddi
YXA210040

16.32. Consider the relationship REFRIG (Model#, Year, Price,Manuf_plant,Color),which is abbreviated as REFRIG(M, Y, P, MP, C), and the following set F of functional dependencies: F = {M → MP, {M, Y} → P, MP → C}
a. Evaluate each of the following as a candidate key for REFRIG, giving reasons why it can or cannot be a key: {M}, {M, Y}, {M, C}.
{M} is not a key because there isn't a dependency from M to P or M to Y.
{M, C} is not a key because there isn't a dependency from {M,C} to P or {M,C} to Y.
{M,Y} is a key because M→MP -------{M,Y}→MP
{M, Y}→P
MP→C and {M,Y}→MP -------{M,Y}→C

b. Based on the above key determination, state whether the relation REFRIG is in 3NF and in BCNF, giving proper reasons.
The relation REFRIG is not in 3NF because there is a transitive dependence from MP to C.
REFRIG is also not in BCNF because it is not in 3NF.

c. Consider the decomposition of REFRIG into D = {R1(M, Y, P), R2(M, MP, C)}. Is this decomposition lossless? Show why. (You may consult the test under Property NJB in Section 16.2.4.)
R1∩R2 is {M}, and {M}→{M, MP, C} is in F+, so this decomposition is lossless.