National University of Computer and Emerging Sciences, Lahore Campus

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| final design | **Course:** | **Database Systems** | **Course Code:** | **CS2005** |
| **Program:** | **BS (Computer Science)** | **Semester:** | **Spring 2023** |
| **Out Date:** | **22-Mar-2023** | **Total Marks:** | **60** |
| **Due Date:** | **Mon 3-Apr-2023 (start of class)** | **Weight:** |  |
| **Section** |  | **Page(s):** | **3** |
| **Assignment:** | **3 (FDs & NFs)** |  | **SOLUTION** |

**Instructions:**  This assignment is an individual assignment.

* Clearly mention any assumption you have made.
* You are required to submit the hard copy of your assignment at the start of your cla ss.
* For any query, please contact your TA.

**Q1)** Consider the relation Art (ID, CreatorName, ReleaseDate, SaleDate, QuantitySold), which is abbreviated as A (I, C, R, S, Q) and has the following FDs:

# IC → R, I → C, C → I, R → I, R → CS, S → Q

Which of the following FDs may or may not hold over schema A?

**NOTE:** Give a valid reason for your answer.

1. ICR → Q, ii. SQ → R, iii. R → Q, iv. I → S, v. SQ → S

[ICR]+=[I,C,R,C,S,Q] hence (i) holds

[SQ+]=[S,Q] hence (ii) does not hold

[R+]=[R,Q,I] hence (iii) holds

[I+]=[I,C,R,S,Q] hence (iv) holds

[SQ+]=[S,Q] hence (v) holds

**Q2**) Consider the relation Course (CourseID, Name, Grade, Instructor), which is abbreviated as Course (C, N, G, I) and has a set of FDs:

# F = {NG→C, CN→I, I→G}.

Find all possible keys (i.e., minimal of super key) of the relation Course.

[NI]+=[N,I,G,C]

[NG]+=[N,G,C,I]

[NC]+=[N,C,I,G]

Possible Keys Are = NI, NG, & NC

**Q3)** Consider the relation S (B, C, D, E, F, G, H, I, J) and a set of FDs.

# F = {D→H, E→C, BC→DE, B→F, H→D, I→J, C→GI}.

Find all possible keys of S.

[BC]+=[B,C,D,E,H,F,G,I,J]

[BE]+=[B,E,C,D,H,G,I,J]

Possible keys are BC & BE

**Q4)** Consider the following two sets of FDs of Relation R (A, B, C, D, E, H).

# F= {B → CD, A → B, EA →H, C → E} K= {A →BD, AC →H, BC → DE}

Check whether they are equivalent. **NOTE:** Show all steps.

For equivalence F Should cover K & K should cover F

Check if F Covers K:

[A]+=[A,B,C,D] yes

[A,C]+=[A,C,B,E,H] yes

[B,C]+=[B,C,D,E] yes

Hence F Covers K

Now, Check if K Covers F:

[B+]=[B] NO

NOT EQUIVALENT

**Q5)** Consider the relation S (G, H, I, J, K, L) and a set of FDs.

# F = {GH→I, J→KL, I→G, L→IJ, G→JL}.

Compute the minimal cover for F (i.e., Fc). Also find all possible keys of S.

Fc={~~GH→I,~~ J→KL, I→G, L→I~~J~~, G→J~~L~~} or Fc={J→K, J→L,I→G, L→I, G→J}.

[HG]+=[H,G,I,J,L,K]

[HI]+=[H,I,G,J,L,K]

[HJ]+=[H,J,K,L,I,G]

[HL]+=[H,L,I,J,K,G]

Possible keys are = HG,HI, HJ & HL

**Q6)** Consider the relation R (A, B, C, D, E, H, Z, F, G) and a set of FDs F = {AB → C, BC → D, CD → E, DE → H, EH

→ Z, HZ → A, FG → E, DE → F, AF → G, GH → B}. Find anomalies exist in it, and find all key & non key attributes, and minimal cover.

Fc=F= {AB → C, BC → D, CD → E, DE → H, EH→ Z, HZ → A, FG → E, DE → F, AF → G, GH → B}

Anomalies:

Insertion, Deletion, Update (Any)

Alternatively, accepting answers explaining what conditions violated to not be in a particular NF.

Key Attributes: A, B, C,D,E,F,G,H,Z (ALL)

Non-Key Attributes: NONE

**Q7) Consider a relation schema R (A, B, C, D, E, H, Z, F, G) with FDs F= {ABC → DE, CD → H, BE → Z, H → FG, C → G}**

* 1. Identify the best normal form that R satisfies (1NF, 2NF, 3NF, or BCNF). Justify your answer.
  2. Decompose the relation R into a 2NF schema if it is not in 2NF. (Remove 2NF violations only, in this part)
  3. Check whether your answer to part (b) is in 3NF. If not, decompose it into a 3NF schema.
  4. Check whether your answer to part (c) is in BCNF. If not, decompose it into a BCNF schema.

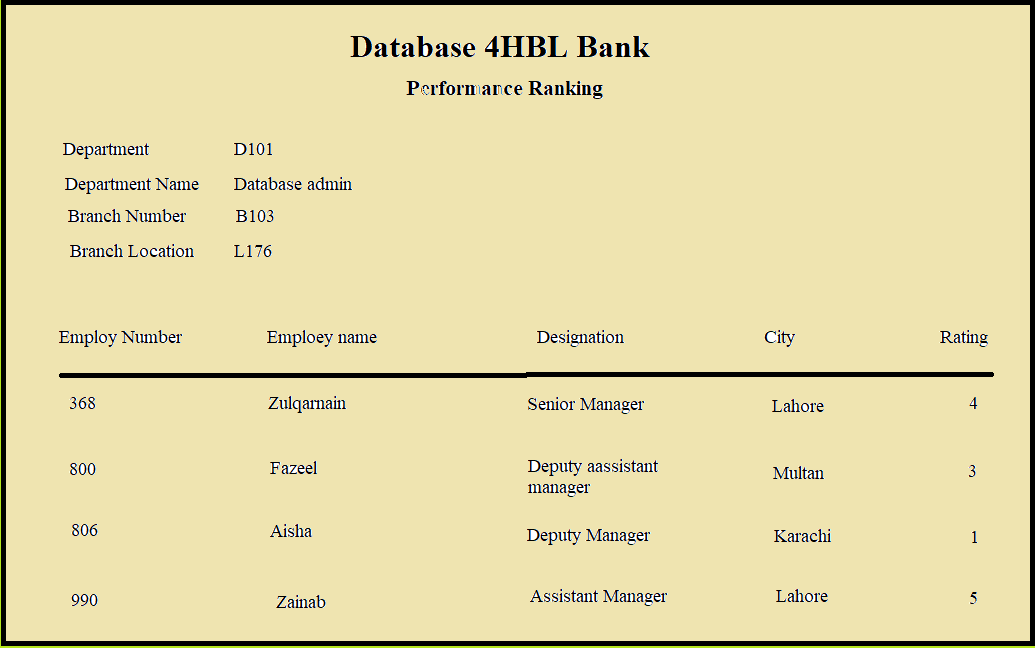
**Answer:**

1. **Is it in 1nf and it is not in 2nf as part of key which is c identifying g.**
2. **Key is ABC, so decompose it into R1 (C, G), R2 (A, B, C, D, E, H, Z, F) to remove partial dependency.**
3. **It is not in 3nf, H is not prime and right side attributes are also not part of prime attributes so ,**

**R1(C, G), R2 (A, B, C, D, E,F), R3(C,D,H),R4(B,E,Z)**

1. **It is also in BCNF now.**

**Q8)** Figure given below shows the performance ranking for Database 4HBL Bank. Assume the following:

* A branch has a unique location.
* An employee has a unique designation.
* A department has a unique name.

Your assignment is as follows:

1. Draw a relational schema for above and show the functional dependencies in the relation.
2. Identify the key and highest normal form of your relation?
3. Decompose your relation into a set of 3NF relations. Show your steps.
4. Decompose your relation in part (a) into set of BCNF relations. Show your steps.

**Answer:**

1. **DB4HBLBank (Branch No, Branch location, Employ No, Employ Name, Designation, Rating, Department No, Department name) FDS= {Branch No🡪 Branch location, Employ No 🡪 (Employ Name, Designation, Rating), Department No🡪(Department name)}**
2. **Key = Branch No + Employ No + Department No and highest normal form is 1NF.**
3. **1st decomposition is R1(Branch No, Branch Location} , R2(Branch No , Employ No , Employ Name, Designation, Rating , Department No , Department Name)**

**2nd decomposition is on R2, R21(Employ No, Employ Name , Designation, Rating), R22(Branch No, Employ No , Department No, Department Name)**

**3rd decomposition is on R22, R221(Department No, Department Name), R222(Branch No, Employ No, Department No)**

1. **It is also in BCNF now.**

**Q9)** Consider a university library system that maintains information about books, students, departments, and courses. The system includes the following attributes:

Book: BookID, Title, Author, Publisher, Edition, ISBN

Student: StudentID, Name, Email, Phone, DepartmentID, CourseID, AdvisorID Department: DepartmentID, Name, Building, ChairpersonID

Course: CourseID, Name, Description, DepartmentID, InstructorID

In addition, the system keeps track of the following information about each book issued by a student: Issue: IssueID, BookID, StudentID, IssueDate, DueDate, ReturnDate

Fine: FineID, IssueID, Amount, PaidDate

The following functional dependencies exist:

BookID → Title, Author, Publisher, Edition, ISBN

**StudentID** → Name, Email, Phone, DepartmentID, **CourseID**, AdvisorID DepartmentID → Name, Building, ChairpersonID

CourseID → Name, Description, DepartmentID, InstructorID IssueID → BookID, StudentID, IssueDate, DueDate, ReturnDate FineID → IssueID, Amount, PaidDate

Check either it is in 1NF,2NF or 3NF, if not, give reason for each and convert them into 3NF.

**Answer:**

**It is in 1nf as all tables have atomic value and no repeating group.**

**Relation is in 2nf.**

**To convert it into 3nf tables will change and few add, (blue highlighted transitive dependency)**

**Bookissue (BookID, StudentID, IssueDate, DueDate, ReturnDate)**

**BookingFine (IssueID, Amount, PaidDate),**

**Studentdepartment (DepartmentID, StudentId, CourseID, AdvisorID)**

**Student (StudentID, Name, Email, Phone)**

**Q10)** Consider the following collection of relations and dependencies. For each relation, state the highest normal form that the relation is in. Find minimal cover and if it is not in BCNF, decompose it into a collection of BCNF relations.

1. R1 (A, C, B, D, E, F, G, H) F = {A → BCEFGH, AC → BD, BC → DEFGH, B → EFG, D → FH, E → AGH}

**Answer:**

**Key =A**

**No partial dependency as key is single attribute.**

**Following violating the BCNF, AC → BD, BC → DEFGH, B → EFG, D → FH, E → AGH**

**1st decomposition R1(ACBD), R2(ABCEFGH)**

**2nd decomposition is of R2, R3(BEFG), R4(AECH)**

**Minimal Cover= [ A → BA → CA → EA → FA → GA → HAC → BBC → DBC → EBC → FBC → GBC → HB → ED → FE → AE → G**

**]**

1. R2 (I, J, K, L, M, N, O, P, Q) F= {IJ → K, IJ → L, K → M, L → N, O → P, P → Q, NO → K, NO → P, IP → L, KP → Q, JO → N, JO → Q}

**Answer:**

**Key=IJO**

**Functional Dependencies violating 2nf as part of key is defining attributes**

**So, R1(I, J, K, L, M, N), R2(O, P, Q)**

**Now to remove the transitive dependency,**

**R11(I,J,K,L) , R12(K,M), R13(L,N) , R21(O,P) R22(P,Q)**

**It is also in BCNF now.**

1. R3 (R, S, T, U, V, W, X, Y, Z) F={RS→ T,TU → V,VW→X, X → Y, Y → Z, T→ W, UY→ S, WZ→ R, RY→ U, X→ Z}

**Answer:**

**Key = {RX,WX,RY,TV,TU,TX,TY,WV,WX,WY}**

**IT’s BCNF form will be,**

**R1 (R, S, T, W), R2 (T, U, V), R3 (U, Y, S), R4 (W, Z, R), R5 (R, Y, U)**

**Q11)** Suppose you are designing a database for a large e-commerce company that sells products online. The database will store information about customers, orders, products, and reviews. You have gathered the following requirements:

Customers can place multiple orders, but each order can only belong to one customer. Each order can contain multiple products, and each product can be in multiple orders. Each product can have multiple reviews, but each review can only belong to one product.

Customers can write multiple reviews, but each review can only be written by one customer. Products can have multiple categories, and each category can contain multiple products.

The database must be able to handle a high volume of reads and writes.

Based on these requirements, you have come up with the following initial set of relations and functional dependencies:

**Customer** **(CustID, CustName, CustEmail, CustAddress)**

CustID → CustName, CustEmail, CustAddress

**Order** **(OrderID, OrderDate, CustID)**

OrderID → OrderDate, CustID

**OrderProduct (OrderID, ProductID, Qty, Price)**

(OrderID, ProductID) → Qty, Price

**Product** **(ProductID, ProdName, ProdDesc, ProdPrice)** ProductID → ProdName, ProdDesc, ProdPrice **ProductReview (ReviewID, ProductID, CustID, ReviewText)** ReviewID → ProductID, CustID, ReviewText **ProductCategory (CategoryID, CategoryName)**

CategoryID → CategoryName

**ProductCategoryLink (ProductID, CategoryID)**

(ProductID, CategoryID) → (nothing)

Based on this initial design, answer the following questions:

Are all these relations in BCNF? If not, decompose any relations that violate BCNF into a collection of BCNF relations.

Is this database design normalized to 3NF? If not, explain which relations violate 3NF and how they can be normalized to 3NF do convert it if not in it.

**Answer: Already in 3NF.**

**Q12)** Why in industry we do not go beyond 3NF/BCNF why not 4NF or 5NF? Is well normalized structure of 3NF, BCNF,4NF, 5NF always be good as no anomalies left after schema is in them.

**Why in industry we do not go beyond 3NF/BCNF why not 4NF or 5NF?**

**Answer:**

**The main reason that the industry typically does not go beyond the 3NF/BCNF is that it may not be necessary and can result in decreased performance and increased complexity. While higher normalization levels such as 4NF or 5NF can further reduce data redundancy and improve data integrity, they can also require additional tables and more complex joins, which can negatively impact database performance. Additionally, implementing higher normalization levels may not provide significant benefits in terms of data quality or reduce data anomalies beyond what is already achieved with 3NF/BCNF. Therefore, in most cases, 3NF/BCNF are considered sufficient for database design, and going beyond that is only considered when there is a specific need to address a data integrity issue that cannot be resolved with lower normalization levels.**

**Is a well normalized structure of 3NF, BCNF,4NF, 5NF always be good as no anomalies left after schema is in them?**

**Answer:**

**No, a well-normalized structure of 3NF, BCNF, 4NF, or 5NF does not guarantee the absence of all types of data anomalies, such as insertion, deletion, and update anomalies. While normalization can reduce data redundancy and improve data integrity, it does not guarantee data accuracy or completeness, and it may not address all types of data anomalies that can arise in a database. Other measures such as data validation, error handling, and business rules should also be implemented to ensure data quality. Too much normalized structure not good as need too many joins for basic query. Now a world where storage cost is decreasing a trade of denormalized will be better.**

Also answer below parts:

1. **You are designing a database for a hospital system that needs to store information about patients, doctors, appointments, and medications. How would you ensure that the database is in 3NF? What trade- offs might you have to make in terms of performance or usability?**

**Answer:**

**By using 3nf rules learn in this course. Second,**

**Increased complexity: Creating a database that is flexible and scalable can result in a more complex database design, which may make it harder to maintain or modify the system.**

**Slower performance: Having a normalized database with many tables can lead to more complex queries and slower performance.Increased storage requirements: Normalization can increase the number of tables in a database, which may result in increased storage requirements.Increased development time: Designing and building a normalized database can take longer than a non-normalized database, which may delay project delivery.User complexity: A highly normalized database may require more complex queries and joins, which may be more difficult for end-users to understand and using a fully normalized may not be good as it can slower performance and in the system where life involves each second wait matters.**

1. **Consider a database that stores information about music artists, albums, and tracks. You want to allow users to search for tracks by artist, album, or genre. How would you design the database to ensure that it is in BCNF? What challenges might you face in ensuring that the database is also optimized for search performance?**

**Answer:**

**By using BCNF rules learn in this course. Other one,**

**I would make sure that every non-key column is dependent exclusively on the primary key of its associated table while designing the database for music artists, albums, and tracks in BCNF. To reduce redundancy, I would also find and remove any functional dependencies that overlapped.**

**I would establish indexes on the columns that are most used in searches, such as the artist, album, and genre columns, to improve the database's search performance. To lower the volume of database queries and lessen server stress, I would also think about adopting query optimization strategies such caching frequently used data.**

**Finding a balance between the demand for efficient search queries and the requirement for normalized tables can be difficult when creating the database for search performance optimization. Additionally, can store the join form of tables in view which are accessed at high frequency.**

1. **You are designing a database for a financial institution that needs to store information about transactions, accounts, and customers. How would you ensure that the database is in 4NF, 5NF? What implications might this have for data retrieval and reporting?**

**Answer:**

**We must examine the functional dependencies in the relations and confirm that they are in Boyce-Codd Normal Form (BCNF) or Fifth Normal Form to confirm that the database is in 4NF and 5NF. (5NF). In order to describe complicated interactions, this may include breaking down relations that deviate from these forms and building new relations.**

**Higher normal forms can decrease redundancy and improve data integrity, but they can also complicate data retrieval and reporting because they may call for connecting several tables or sophisticated relationship queries. It's crucial to carefully weigh the trade-offs and create the database with the financial institution's unique requirements in mind.**