CS157A: Introduction to Database Management Systems

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Chapter 3. Design Theory for Relational Databases

Database Design

- Designing a relational database schema for an application.
- If a database schema has flaws, it will cause problems referred to as "design anomalies".
- Often, the design anomalies are caused by too much information is crammed into a single relation.
- "Dependencies" can imply what makes a good relational database schema and what we can do about a schema if it has flaws.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Design Anomalies

- Anomalies: problems caused by too much information is crammed into a single relation
  - Redundancy - capturing info multiple times
  - Update Anomalies – forget to update in a tuple
  - Deletion Anomalies – deleting a tuple causes a loss of other information as a side effect

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Solution: Normalization

The goal of normalization is to decompose a relation into several in a way that the decomposition will have

Assignment Project Exam Help

- Elimination of Anomalies

https://powcoder.com

- Recoverability of Information

Add WeChat powcoder

- Preservation of Dependencies

Normalization

- 1NF → 2NF → 3NF → BCNF → 4NF (optional)
- The higher NF implies the lower NF.
- These days, there is little use for 1NF and 2NF
- BCNF will be covered in detail.
- 3NF is more relaxed condition than BCNF to allow a relational schema that cannot be decomposed into BCNF relations without losing both the lossless-join and dependency –preservation properties
- The most commonly used and easiest normal form to get to that will give you a good database is known to be third normal form (3NF).

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

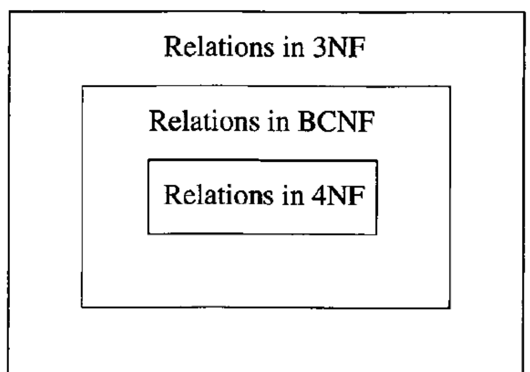


Figure 3.12: 4NF implies BCNF implies 3NF

BCNF Normalization

Relation + its functional dependencies → normalization → Relations in BCNF

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Idea of Normalization

- Define BCNF in terms of FD and key.
- From a given mega relation, discover all true FD's: closure algorithm
- Identify BCNF violations and decompose relations until no BCNF violation exists

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Functional Dependencies

- Definition
  - In a relation *R*, a set of attributes A is said to **functionally determine** another set of attributes B (A→B), if two tuples of R agree on A then they also agree on B.
  - R satisfies a FD if the FD is true for every instance of R
- Implication
  - Each A value is associated with precisely one *B* value.
  - If the A value is known, then the B value corresponding to *A* can be determined by looking up in any tuple of R containing the A value.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Functional Dependencies

Suppose a relational schema is (A, B, C) and A → B

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

A	**B**	**C**
a	b	c1
a	b	c2

Example: Functional Dependencies

Based on the knowledge of the real world:

`Movies1(title, year, length, genre, studioName, starName)`

title	year	length	genre	studioName	starName
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne’s World	1992	95	comedy	Paramount	Dana Carvey
Wayne’s World	1992	95	comedy	Paramount	Mike Meyers

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

`title year → length genre studioName (o)`
`title year → starName (x)`

Keys

- In a relation R with no duplicates, if <u>A</u> $\rightarrow$ all other attributes, then <u>A</u> is a key of R.
- Minimal key: no proper subset of <u>A</u> functionally determines all other attributes
- Super key: a set of attributes that contains a key.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Example: Keys

[Q] Is {title, year, starName} a key for Movie1 ?

[A] Yes. {title, year, starName} functionally determines all other attributes of Movie1.

Assignment Project Exam Help

https://powcoder.com

[Q] Is the key minimal?

[A] Yes. No proper subset of the key can functionally determine all other attributes.

Add WeChat powcoder

{title} {year} {starName} {title, year}

{title, starName}{year, starName}

Functional Dependency Rules

With a given set of FDs, we can deduce other functional dependencies using following rules.

Assignment Project Exam Help

- Transitive rule

https://powcoder.com

- Splitting rule

Add WeChat powcoder

- Combining rule
- Trivial dependency rules (two of them)

Transitive Rule

Example: If we are told that a relation R (A, B, C) satisfies the FD's $A \rightarrow B$ and $B \rightarrow C$, we can deduce that R also satisfies the FD $A \rightarrow C$.

Assignment Project Exam Help

- Let (a, b_1, c_1) and (a, b_2, c_2) be two tuples that agree on attribute A.

https://powcoder.com

Add WeChat powcoder

- Since R satisfies $A \rightarrow B$ it follows that $b_1=b_2$ so the tuples are: (a, b, c_1) and (a, b, c_2)
- Similarly, since R satisfies $B \rightarrow C$ and the tuples agree on B they will agree also on C. So, $c_1=c_2$.

The Splitting/Combining Rule involving the right side of FDs

$A_1A_2 \ldots A_n \rightarrow B_1$
$A_1A_2 \ldots A_n \rightarrow B_2$
...
$A_1A_2 \ldots A_n \rightarrow B_m$

Combining Rule $\longrightarrow$

$\longleftarrow$ Splitting Rule

$A_1A_2 \ldots A_n \rightarrow B_1B_2 \ldots B_m$.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Example: The Splitting/Combining Rule involving the right side of FDs

title year → length
title year → genre
title year → studioName

Combining Rule →

← Splitting Rule

title year → length genre studioName

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Can we split the left side ?

[Q]

From, title year $\rightarrow$ length, can we deduce

title$\rightarrow$ length (false FD)

year $\rightarrow$ length (false FD) ?

[A] No

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Trivial Functional Dependencies

- A functional dependency $A_1A_2\ldots A_n \rightarrow B$ is **trivial** if B is a subset of A

  Example: title year $\rightarrow$title

Assignment Project Exam Help

- A functional dependency $A_1A_2\ldots A_n \rightarrow B_1B_2\ldots B_m$ is:

https://powcoder.com

  – **Nontrivial** if at least one of the B's is not among the A's.

Add WeChat powcoder

  – **Completely nontrivial** if A and B do not have any overlap.

  Example: title year $\rightarrow$ year length is nontrivial but not completely nontrivial.

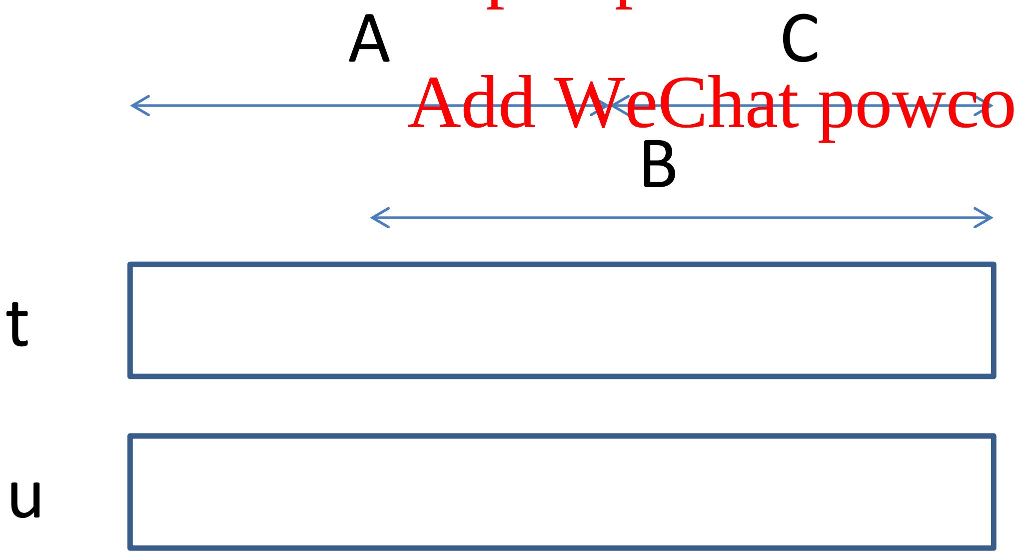
Trivial Dependency Rule

- We can always remove from the right side of a FD those attributes that appear on the left.
- Suppose A $\rightarrow$ B. Then, A$\rightarrow$ C

Assignment Project Exam Help

e.g) title, year $\rightarrow$ year, length. Then, title, year$\rightarrow$ length

https://powcoder.com



Add WeChat powcoder

Computing the Closure of Attributes

- The closure of a set of attributes $\{A_1, A_2, \ldots, A_n\}$: $\{A_1, A_2, \ldots, A_n\}^+$
- Suppose $\{A_1, A_2, \ldots, A_n\}$ is a set of attributes and S is a set of FD's. $\{A_1, A_2, \ldots, A_n\}^+$ under the dependencies in S is the set of attributes B, which are functionally determined by $A_1, A_2, \ldots, A_n$
- That is, it finds $A_1A_2\ldots A_n \rightarrow B$ that follows from S
- Since we allow trivial dependencies, $A_1, A_2, \ldots, A_n$ are in $\{A_1, A_2, \ldots, A_n\}^+$.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Algorithm 3.7: Closure of a Set of Attributes

Input: A set of attributes $\{A_1, A_2, \ldots, A_n\}$ and a set of FD's S

Output: The closure $\{A_1, A_2, \ldots, A_n\}^+$

1. Let X be a set of attributes that eventually will become the closure. Initialize X to be $\{A_1, A_2, \ldots, A_n\}$.
2. Now, repeatedly search for some FD in S:

   $$B_1B_2\ldots B_m \rightarrow C$$

   such that all of B's are in the set X, but C is not. We then add C to X.
3. Repeat step 2 as many times as necessary until no more attributes can be added to X. (Since X can only grow, and the number of attributes is finite, eventually nothing more can be added to X.)
4. The set X after no more attributes can be added to it is the: $\{A_1, A_2, \ldots, A_n\}^+$.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Example

- Let's consider a relation with attributes A, B, C, D, E and F. Suppose that this relation satisfies the FD's:

  $AB \to C$, $BC \to AD$, $D \to E$, $CF \to B$.

  What is $\{A,B\}^+$?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

$X = \{A,B\}$ Use: $AB \to C$

$X = \{A,B,C\}$ Use: $BC \to AD$

$X = \{A,B,C,D\}$ Use: $D \to E$

$X = \{A,B,C,D,E\}$ No more changes to X are possible so $X = \{A,B\}^+$.

- The FD: $CF \to B$ cannot be used because its left side is never contained in X.

Use of the closure of attributes

We can test whether any given functional dependency $A_1A_2\ldots A_n \rightarrow B$ **follows** from a set of dependencies **S**.

- Compute $\{A_1, A_2, \ldots, A_n\}^+$ using the set of dependencies S.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- If $B \in \{A_1, A_2, \ldots, A_n\}^+$ then the FD: $A_1A_2\ldots A_n \rightarrow B$ **does** follow from **S**.

- If $B \notin \{A_1, A_2, \ldots, A_n\}^+$ then the FD: $A_1A_2\ldots A_n \rightarrow B$ **doesn't** follow from **S**.

Example: Use of the closure of attributes

[Q] Consider the previous example. Test whether $AB \rightarrow D$ follows from the set of the dependencies.

[A] Yes since $D \in \{A,B\}^+ = \{A,B,C,D,E\}$.

Assignment Project Exam Help

[Q] Consider the FD: $D \rightarrow A$.

https://powcoder.com

[A] No since $A \notin \{D\}^+ = \{D,E\}$. We say, $D \rightarrow A$ does not follow from the given set of dependencies.

Add WeChat powcoder

X= {D} Use D→ E

X= {D, E} we have reached the closure.

The Transitive Rule

- If $A_1A_2 \ldots A_n \rightarrow B_1B_2 \ldots B_m$ and $B_1B_2 \ldots B_m \rightarrow C_1C_2 \ldots C_m$, then $A_1A_2 \ldots A_n \rightarrow C_1C_2 \ldots C_k$
- Prove this rule using the closure of attributes

Assignment Project Exam Help

1. With $\{A_1A_2 \ldots A_n\}$ and two FDs $A_1A_2 \ldots A_n \rightarrow B_1B_2 \ldots B_m$ and $B_1B_2 \ldots B_m \rightarrow C_1C_2 \ldots C_m$

https://powcoder.com

Add WeChat powcoder

2. $\{A_1, A_2, \ldots, A_n\}^+ = \{A_1A_2 \ldots A_n, B_1B_2 \ldots B_m, C_1C_2 \ldots C_m\}$
3. Therefore, $A_1A_2 \ldots A_n \rightarrow C_1C_2 \ldots C_k$ follows from the given FDs.

Example: The Transitive Rule

title	year	length	genre	studioName	studioAddr
Star Wars	1977	124	sciFi	Fox	Hollywood
Eight Below	2005	120	drama	Disney	Buena Vista
Wayne's World	1992	95	comedy	Paramount	Hollywood

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

title year → studioName
studioName → studioAddr

Then, we can deduce a new FD based on the transitive rule.

title year → studioAddr

Closures and Keys

- $\{A_1, A_2, \ldots, A_n\}^+$ is the set of **all** attributes of a relation if and only if $\{A_1, A_2, \ldots, A_n\}$ is a **superkey** for the relation.
- We can test if $A_1, A_2, \ldots, A_n$ is a minimal key for a relation by checking:
  - **first** that $\{A_1, A_2, \ldots, A_n\}^+$ contains all attributes,
  - **and** if any of attribute is removed from $\{A_1, A_2, \ldots, A_n\}$, then {reduced set of attributes}$^+$ will not contain all the attributes.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Design of Relational Database Schemas

- The principal problem that we encounter is redundancy, where a fact is repeated in more than one tuple.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Example: relation with redundancy

Movie1 Relation

Assignment Project Exam Help

title	year	length	genre	studioName	starName
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

https://powcoder.com

Add WeChat powcoder

Anomalies

- **Redundancy**.
  - capturing info multiple times unnecessarily. E.g. length and genre.
- **Update anomalies**.
  - forget to update in a tuple
  - E.g. we could change the length of *Star Wars* to 125, in the first tuple, and forget to do the same in the second and third tuple.
- **Deletion anomalies**.
  - deleting a tuple causes a loss of other information as a side effect
  - E.g. if we delete Vivien Leigh. we will lose all the information about Gone With the Wind.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Decomposing Relations - Example

Movie2 Relation

title	year	length	genre	studioName
Star Wars	1977	124	sciFi	Fox
Gone With the Wind	1939	231	drama	Disney
Wayne's World	1992	95	comedy	Paramount

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

title	year	starName
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Gone With the Wind	1939	Vivien Leigh
Wayne's World	1992	Dana Carvey
Wayne's World	1992	Mike Meyers

Movie3 Relation

Decomposing Relations - Example

- No true redundancy!
- The update anomaly disappeared. If we change the length of a movie, it is done only once.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- The deletion anomaly disappeared. If we delete all the stars from $Movie_2$ we still will have the other info for a movie.

Boyce-Codd Normal Form (BCNF)

- The goal of decomposition is to replace a relation by several that do not exhibit anomalies.
- There is a simple condition called Boyce-Codd Normal Form under which the anomalies can be guaranteed not to exist.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- A relation is in BCNF if and only if: whenever there is a nontrivial dependency $A_1 A_2 \ldots A_n \rightarrow B_1 B_2 \ldots B_m$ for R, it must be the case that $\{A_1 , A_2 , \ldots , A_n\}$ is a superkey for R.
- That is, the left side of every nontrivial functional dependency must contain a key.

Example: BCNF

Relation Movie1 is not in BCNF.

- {title, year, starName} is a key of the relation.
- Consider the FD: title year $\rightarrow$ length genre studioName
- The left side of the above dependency is not a super key. In particular, we know that the title and the year does not functionally determine starName.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Movie2 is in BCNF.

- The only key is {title, year} and
- title year $\rightarrow$ length genre studioName holds in the relation

Example: BCNF

A relation with two attributes is always in BCNF.

(a) If there is no non Trivial FDs, then it is in BCNF

Assignment Project Exam Help

(b) A→ B, but not B→A: A is the only key and the left side of non-trivial FD A→ B contains A.

https://powcoder.com

Add WeChat powcoder

(c) B→ A, but not A→ B: Symmetric to (b)

(d) A→ B and B→ A: both A and B are keys. A→ B contains a key (A) and B→ A contains a key(B) in their left sides, respectively.

Decomposition into BCNF

- Decomposition Strategy
  - Find a non-trivial FD $A_1A_2 \ldots A_n \rightarrow B_1B_2 \ldots B_m$ that violates BCNF, i.e. $A_1A_2 \ldots A_n$ is not a super key.
  - Decompose the relation schema into two overlapping relation schemas:
  - One is all the attributes involved in the violating FD and
  - The other is the left side of the FD and all the other attributes not involved in the FD.
  - By repeatedly, choosing suitable decompositions, we can break any relation schema into a collection of smaller schemas in BCNF.
- The original relation should be able to be reconstructed from the decomposed relations.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Projecting Functional Dependencies

- It will be used to find FDs for the decomposed relations so that we can eventually check that the decomposed relations are in BCNF.
Assignment Project Exam Help
- Suppose a relation R with a set of FD's S and R1 is a projection of R.
https://powcoder.com
Add WeChat powcoder
  - What FDs hold for R1 ?
  - The algorithm will find all FDs that follow from S and involve only attributes of R1

Algorithm 3.12: Projecting Functional Dependencies

Input: R1 and a set of FD's on R

Output: a set of FD's T that hold in R1

Method:

- Consider each subset X of attributes of R1.
- Compute X^+ using the FD on R.
- At the end, throw out the attributes of R, which aren't in R1.
- Then, add to T all nontrivial FD's X→ A such that A is in X^+ and A is an attribute of R1
- Construct a minimal basis of T.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Example: Projecting Functional Dependencies

- Consider R(A, B, C, D, E) decomposed into R1(A, C, D) and another relation. Let FDs of R be: $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$.
- $\{A\}^+ = \{A,B,C,D\}$ $T = \{A \rightarrow C, A \rightarrow D\}$
- $\{C\}^+ = \{C,D\}$ $T = \{A \rightarrow C, A \rightarrow D, C \rightarrow D\}$
- $\{D\}^+ = \{D\}$ $T = \{A \rightarrow C, A \rightarrow D, C \rightarrow D\}$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Since $\{A\}^+$ includes all attributes, we don't need to consider any superset of {A}.

- $\{C,D\}^+ = \{C,D\}$ $CD \rightarrow C$ or $CD \rightarrow D$ are trivial. Therefore, T won't be changed.
- Based on the transitive rule, $A \rightarrow D$ follows from $A \rightarrow C$ and $C \rightarrow D$.
- $T = \{A \rightarrow C, C \rightarrow D\}$ which is a minimal basis.

Some simplifications

- Don’t need to compute the closure of the empty set or of the set of all attributes because they never yield a non-trivial FD.
- If we find X^+ = all attributes, don’t bother computing the closure of any supersets of X.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Algorithm 3.20: BCNF Decomposition Algorithm

Input: A Relation R with a set of functional dependencies S
Output: Decomposed relations in BCNF

The following steps can be applied recursively to any relation R and a set of FD's S.

Assignment Project Exam Help

- Check if R is in BCNF. If so, return R as it is.
- If there are BCNF violations, let one be X→Y.
- Use Algorithm 3.7 to compute X^+. The relation will be decomposed into R1 = X^+ and R2 = X and the attributes that are not in X^+.
- Use Algorithm 3.12 to project FD's for R1 and R2. Let these be S1 and S2, respectively.
- Recursively decompose R1 and R2 using this algorithm.

https://powcoder.com
Add WeChat powcoder

Return the union of the results of these decompositions.

Example: BCNF (continued)

- Consider a schema:
  `(title, year, studioName, president, presAddr)`
  and functional dependencies:
  `title year → studioName`
  `studioName → president`
  `president → presAddr`

Assignment Project Exam Help

- To find BCNF violating FDs, you need to find keys of this relation. Compute {title, year}+, {studioName}+, {president}+ and see if you get all the attributes of the relation. If not, you got a BCNF violation, and need to break relation. You will find that {title, year} is the only key.

https://powcoder.com
Add WeChat powcoder

- Last two violate BCNF.

Example: BCNF (continued)

- Decomposition can start with any of these violating FDs. Let’s starting with studioName → president
- Add to the right-hand side any other attributes in the closure of studioName (optional but, often reduces the amount of work)

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

1. X={studioName} studioName→president
2. X={studioName, president} president→presAddr
3. X={studioName}$^+$={studioName, president, presAddr}

Example: BCNF (continued)

The choice of FD is now studioName → president presAddr
Therefore, the original relation is decomposed into

Movies1(title, year, studioName)
Movies2(studioName, president, presAddr):

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

[Q1] Is Movies1 in BCNF ?
[A] Yes

1. Using Algorithm 3.12, find a minimal basis of FDs that hold in Movies1.
2. You will find that Movies1 has a basis title year → studioName.
3. See if {title, year} is a key by finding its closure and see if the closure includes all attributes of Movie1.
   $\{title, year\}^+ = \{title, year, studioName\}$

Example: BCNF (continued)

Movies2(studioName, president, presAddr)

[Q2] Is Movies2 in BCNF ?

[A] No

1. Using Algorithm 3.12, find a minimal basis of FDs that hold in Movies2.

Assignment Project Exam Help

2. You will find that Movies2 has bases

https://powcoder.com

studioName→ president & president → presAddr

Add WeChat powcoder

3. See if {studioName} and {president} are keys.

{studioName}$^+$ = {studioName, present, presAddr}

{president}$^+$ = {president, presAddr} and thus it is not a key.
We conclude president→ presAddr is a BCNF voilation.

Example: BCNF (continued)

- We have to decompose Movie2 into

Movie2-1 (president, presidentAddr)

Movie2-2 (president, studioName)

Assignment Project Exam Help

Both of them are relations with 2 attributes and thus in BCNF.

https://powcoder.com

Add WeChat powcoder

- Final result

Movies1(title, year, studioName)

Movie2-1 (president, presidentAddr)

Movie2-2 (president, studioName)

3NF

- In some cases, it is not possible to decompose a relation into BCNF relations that have both the lossless-join and dependency-preservation properties.
- 3NF allows us to make a tradeoff between preserving dependencies and BNCF.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Example

- Bookings (title, theater, city)
- FD's: theater→ city, title city → theater
- Keys: {title, city} (theater, title}
- BCNF violation: theater→ city
- If Bookings is decomposed into (theater, city) and (theater, title) to satisfy the BNCF condition, the functional dependency "title city → theater" cannot be preserved as shown below

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

theater	city
Guild	Menlo Park
Park	Menlo Park

theater	title
Guild	Antz
Park	Antz

theater	city	title
Guild	Menlo Park	Antz
Park	Menlo Park	Antz

Definition of 3NF

- A relation R is in 3NF if and only if for every non-trivial FD A1A2…An $\rightarrow$ B, either

  {A1, A2, …, An} is a super key, or

  B is an attribute in some key (prime).

- 3NF violation

A non-trivial FD X$\rightarrow$A violates 3NF if and only if X is not a super key and A is not prime.

- If there exists a 3NF violation, decompose R using the synthesis algorithm presented in the next slide.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

The Synthesis Algorithm for 3NF Schemas

- Input: A relation R and a set F of functional dependencies that hold for R
- Output: A decomposition of R into a collection of relations, each of which is in 3NF. The composition has the lossless-join and dependency-preservation properties
- Method:

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

1. Find a minimal basis for F, say G
2. For each functional dependency X→ A in G, use XA as the schema of one of the relations in the decomposition.
3. If none of the relation schemas from Step 2 is a super key for R, add another relation whose schema is a key for R.

Example

R(A, B, C, D) with FD's $AB \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$

- There are 14 nontrivial dependencies. They are: $C \rightarrow A$, $C \rightarrow D$, $D \rightarrow A$, $AB \rightarrow D$, $AB \rightarrow C$, $AC \rightarrow D$, $BC \rightarrow A$, $BC \rightarrow D$, $BD \rightarrow A$, $BD \rightarrow C$, $CD \rightarrow A$, $ABC \rightarrow D$, $ABD \rightarrow C$, and $BCD \rightarrow A$.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- There are three keys. They AB, BC, and BD. Since all the attributes on the right sides of the FDs are prime, there are no 3NF violations.

- Since there are no 3NF violations, it is not necessary to decompose the relation.

Example

R(A, B, C, D) with FD's B→ C and B→ D

- There are 8 nontrivial dependencies. They are: B→C, B→D, AB→C, AB→D, BC→D, BD→C, ABC→D and ABD→C.
- AB is the only key. FDs where the left side is not a superkey and the attributes on the right are not part of some key (prime) are 3NF violations. The 3NF violations are B→C, B→D, BC→D and BD→C.
- Using the synthesis algorithm 3.26, we can decompose into relations using the minimal basis B→C and B→D. The resulting decomposed relations would be BC and BD.
- However, none of these two sets of attributes is a superkey. Thus we add relation AB to the result. The final set of decomposed relations is BC, BD and AB.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

2NF

- Every non-key attribute is dependent on the key – the whole key for composite keys
- Not 2NF because name, role, and dID are dependent on the key (eID, skill) but they are also dependent on just eID.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

eID	name	role	dID	skill
1	John	Developer	1	Java
1	John	Developer	1	Python
2	Dan	DBA	2	MySQL

- Solution – decompose skills table (like before)

1NF

- Each attribute is atomic.
- Not 1NF because skills has multiple values.

Assignment Project Exam Help

eID2	name	role	dID	skills
1	John	Developer	1	Java, Python
2	Dan	DBA	2	MySQL, Oracle

https://powcoder.com

Add WeChat powcoder

- Solution

eID	name	role	dID
1	John	IT	1
2	Dan	DBA	2

eID	Skill
1	Java
1	Python
2	MySQL
2	Oracle